

OWA for Bipartite Assignments

Jabari Hastings 

Stanford University, Stanford, CA, USA

Sigal Oren 

Ben-Gurion University of the Negev, Beer-Sheva, Israel

Omer Reingold 

Stanford University, Stanford, CA, USA

Abstract

In resource allocation problems, a central planner often strives to have a fair assignment. A challenge they might face, however, is that there are several objectives that could be argued to be fair, such as the max-min and maximum social welfare. In this work, we study bipartite assignment problems involving the optimization of a class of functions that is sensitive to the relative utilities derived by individuals in allocation and captures these traditional objectives. We introduce and study a subclass of evaluation functions that targets the average welfare attained within some interval of the economic ladder (e.g., the bottom 10%, middle 50%, or top 80%). We provide an efficient algorithm that can be used to optimize the welfare for an arbitrary interval and also show how the approach can be used to approximate more general evaluation functions. We also study a subclass of evaluation functions consisting of the “fair” ordered weighted averages (OWA) introduced by Lesca et al. (Algorithmica 2019), which are most sensitive to the utilities received by the worst-off individuals. We provide a simple proof that optimizing this objective belongs to the class XP.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory and mechanism design

Keywords and phrases fairness, matchings, approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.FORC.2025.21

Funding *Jabari Hastings*: Supported by the Simons Foundation Collaboration on the Theory of Algorithmic Fairness and the Simons Foundation investigators award 689988.

Sigal Oren: Supported in part by BSF grant 2018206.

Omer Reingold: Supported by the Simons Foundation Collaboration on the Theory of Algorithmic Fairness, the Sloan Foundation Grant 2020-13941 and the Simons Foundation investigators award 689988.

Acknowledgements We thank Shahar Dobzinski and Gergely Csáji for helpful discussions.

1 Introduction

Assignment problems are abundant in real life, arising in contexts such as assigning students to schools (e.g., [2]), matching doctors with fellowship positions (e.g., [18]), and allocating housing to individuals in need (e.g., [4]). In many of these scenarios, agents have numerical valuations (cardinal utilities) for each available object. Typically, a central planner is responsible for determining how to allocate these resources. A common approach involves defining a real-valued objective function over possible assignments and optimizing it.

One widely used objective is to maximize economic efficiency or social welfare – i.e., to maximize the sum of the values that agents derive from their assigned object. However, a potential worry with this approach is that it inherently favors agents with higher valuations, as improving overall welfare is more easily achieved by prioritizing those with the highest values. This can lead to *unfair* outcomes where the most advantaged agents benefit disproportionately.



© Jabari Hastings, Sigal Oren, and Omer Reingold;
licensed under Creative Commons License CC-BY 4.0
6th Symposium on Foundations of Responsible Computing (FORC 2025).
Editor: Mark Bun; Article No. 21; pp. 21:1–21:19



Leibniz International Proceedings in Informatics
LIPIcs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

As an alternative, the Rawlsian perspective advocates for a more equitable allocation by maximizing the value of the least well-off agent in the assignment (max-min objective). This ensures that the agent with the lowest value in the chosen assignment is as well-off as possible.

A central planner that has to choose between max-min and maximizing social welfare faces a strict tradeoff between efficiency and fairness. To allow a smoother tradeoff and capture more nuanced preferences, we can consider a more general objective that encompasses the two objectives. Such objectives can be stated in terms of a *generalized Gini social-evaluation* (GSE) function [20], which aims to capture the inequality of wealth levels in a population. This function, denoted by $\psi_w : \mathbb{R}^n \rightarrow \mathbb{R}$, is defined as $\psi_w(v) := \sum_i w_i v_i^\uparrow$, where $w \in \mathbb{R}^n$ is a non-increasing weight vector and $v^\uparrow$ represents the value vector $v \in \mathbb{R}^n$ sorted in ascending order. By setting all weights to 1, we recover the standard objective of maximizing social welfare. Conversely, if $w_1 = 1$ and the rest of the weights are 0, we obtain the max-min objective. More generally, this framework allows us to express additional fairness-aware objectives, such as maximizing the total utility of the lowest 10 percent, optimizing social welfare while ignoring the top 10 percent, or lexicographic max-min which we can capture by assigning exponentially decreasing weights.

Optimizing GSE functions

Given their significance, GSE functions have been heavily studied within the combinatorial optimization literature. Starting with a seminal work by Yager [21], a long line of work examines GSE functions within the framework of *ordered weighted averages* (OWA) (cf. [22] for a survey). In particular, Lesca et al. [14] define “fair” OWA as the setting with non-increasing weights. A parallel line of work on norm optimization in combinatorial problems (e.g., [5, 1, 11]) has studied *ordered norms*, which are essentially the same as generalized Gini social-evaluation functions, but applied to settings involving cost minimization (e.g., job scheduling, facility location). Note that the difference between cost minimization and utility maximization is not merely terminological; maximization and minimization can lead to significant complexity differences. For example, finding the shortest path in a graph can be easily solved in polynomial time, whereas computing the longest path is NP-hard.

Despite the volume of work on GSE functions, their optimization in the context of the bipartite assignment problem has posed significant computational challenges.¹ Lesca et al. [14] showed that the decision variant of the problem, referred to as Fair OWA Multiagent Assignment (FOWAMA), is NP-hard. They also show that if the vector w is assumed to have at most k distinct values, the parametrized problem is in the class XP (i.e., problems that can be solved in $\mathcal{O}(n^{f(k)})$ time); the algorithm they present runs in polynomial time only for constant k .

Alternatives to GSE functions

Since the optimization problem for GSE functions is solved using a rather complex algorithm, whose running time depends on the number of distinct weights, this raises an important question: if a central planner using a GSE function were instead to optimize purely for social welfare, how much worse would the outcome be? It is easy to see that such a naive approach would fail, as the social welfare function cannot even approximate the max-min objective function. However, what if, instead of optimizing the social welfare of the entire population, we focus only on a specific *interval*? For instance, we could optimize over the middle 60

¹ The nature of the problem makes it also related to majorization theory [15].

percent of the population – those ranked between the bottom 20 percent and the top 20 percent. Such an optimization is also interesting in its own right. Consider, for example, a politician whose primary voter base consists of the middle class. When making assignment decisions, such a politician might prefer to optimize the utility of the middle class rather than that of the entire population. Observe that an arbitrary interval defines a function with binary weights that are not necessarily non-increasing. Therefore, existing paradigms like fair OWA and ordered norm optimization do not provide solutions for this new problem. Nonetheless, we show that optimization over an interval can be done in polynomial time.

Continuing with the perspective of the central planner as a politician trying to satisfy different segments of the population, motivates considering general weight functions. These can result from an aggregation rule that combines the preferences of different population segments into a single weight vector for the optimization problem. For example, if each segment prioritizes a specific interval, the central planner might assign equal weight to all positions that are considered important by at least one segment or weight each position according to the number of segments that care about it. An alternative interpretation of the aggregation process arises from considering the preferences of k stakeholders – agents who participate in the assignment process. In the spirit of the *veil of ignorance* [17], each stakeholder's preference for an assignment mechanism is determined by their estimate of where they might fall in the ranked distribution. Under this view, each agent effectively has a private OWA function, which the central planner aggregates into a single function. As part of this work, we provide approaches to approximate the aggregated functions and discuss their limitations.

1.1 Our Contributions

Our contributions can be distilled into the following vignettes:

- **Optimizing over a single interval.** We present a simple, efficient algorithm that optimizes the welfare for any specified interval within the population. In fact, we show that optimizing any interval is almost as easy as computing a maximum-weight perfect matching. This means that, with little computational overhead, we provide a decision-maker with additional flexibility regarding their choice of segment to optimize.
- **Optimizing over multiple intervals.** As a next natural step, we ask: what if we restrict our attention to binary weight functions? If the number of contiguous intervals of "1"s that we care about is k , then a trivial argument implies that, since there are k intervals, there must exist an interval such that optimizing over it guarantees a k -approximation. This follows simply by selecting the interval that contributes the maximal utility to $\psi_w(\cdot)$. We show that when the intervals are of equal size, we can do significantly better: there always exists a single interval that we can optimize over to achieve an $\mathcal{O}(\log k)$ -approximation of the optimal solution. We note that our result is in fact more nuanced and can also handle, to some degree, intervals of different sizes with some loss in the approximation ratio.
- **Optimizing GSE functions.** We provide a much simpler proof of the result of Lesca et al. [14] that optimizing GSE functions in the bipartite matching context is in the class XP. Our result is obtained by adapting a framework introduced by Chakrabarty and Swamy [5] for minimizing monotone symmetric vector norms. At a high level, their approach involves designing proxy functions to replace the vector norm with a nearly equivalent objective that is less sensitive to the relative order of entries in the input vector. The replacement objective is then (approximately) solved using linear programming techniques. For our problem, we have to construct new proxy functions and show that they provide a good approximation of our desired objectives.

- **Optimizing over General Weights.** Observe that since the restricted case of non-increasing weights [14] is NP-hard, the more general problem with arbitrary weights is also NP-hard.² Given this complexity, we again consider the question of optimizing over a single interval and show that such an optimization cannot guarantee an approximation ratio better than $\mathcal{O}(\log n / \log \log n)$. The lower bound holds even for restricted binary weights.

1.2 Additional Related Work

The theme of optimizing relaxations of max-min and maximum social welfare objectives has a long history in the Operations Research literature on facility location (cf. [3] for a comprehensive survey) and the work of Chakrabarty and Swamy [5] has sparked interest in applying this theme to other combinatorial problems (e.g. job scheduling, load balancing). In addition to the works mentioned in the introduction, there are several other related works worth discussing. Notably, the work of Deng and Zhang [6] analyzes cost minimization in facility location for a class of objectives that includes an interval. Regarding the subject of fairness, an exciting research direction aims to simultaneously (approximately) satisfy a class of objectives (e.g., [13, 7]). A more recent variant, proposed by Gupta et al. [10], studies the design of a small portfolio of solutions that simultaneously satisfies a class of objectives; this setting differs from ours in that we search for a single solution rather than a portfolio.

2 Preliminaries

We consider an assignment problem consisting of n agents and n indivisible items. Each agent $i \in [n]$ has a non-negative valuation for each item $j \in [n]$ denoted by $v_i(j)$. Throughout this work, we will use a square matrix $V \in \mathbb{R}^{n \times n}$ to concisely convey agents' valuations (so that $V_{ij} = v_i(j)$). We assume that each agent wants a single item (i.e., agents have unit demand valuations). We denote an allocation by a permutation $\pi : [n] \rightarrow [n]$, so that agent i receives item $\pi(i)$. Note that a valid assignment corresponds to a perfect bipartite matching. We let S_n denote the set of perfect matchings for our problem instance. Every allocation $\pi \in S_n$ induces a vector of values denoted by $v(\pi) := (v_1(\pi(1)), v_2(\pi(2)), \dots, v_n(\pi(n)))$. Our focus is on the *ranked value vector* $v(\pi)^\uparrow := (v(\pi)_1^\uparrow, v(\pi)_2^\uparrow, \dots, v(\pi)_n^\uparrow)$, the result of sorting the entries of the vector $v(\pi)$ in increasing order. In particular, $v(\pi)_1^\uparrow$ is the smallest value in $v(\pi)$ attained by an agent and $v(\pi)_n^\uparrow$ is the largest value attained in $v(\pi)$.

Ordered Weighted Averages

Given a vector $w \in \mathbb{R}_{\geq 0}^n$, we define a function $\psi_w : \mathbb{R}^n \rightarrow \mathbb{R}$ that maps an arbitrary assignment $\pi \in S_n$ to a weighted linear combination of the values derived by the individuals as follows:

$$\psi_w(v(\pi)) = \sum_{i=1}^n w_i v(\pi)_i^\uparrow \quad (1)$$

Note that the function applies weights that depend on the positions of entries in the ranked value vector, which is sorted in increasing order. This function is known as an *ordered weighted average* (OWA). In an *ordered optimization problem*, our goal is to find a one-to-one assignment $\pi^* \in S_n$ that maximizes the OWA operator $\psi_w(v(\cdot))$:

² We refer here to the decision variant of the problem, which asks whether, given a value Q , there exists an assignment whose resulting utility under ψ_w exceeds Q . It is easy to see that verifying the validity of a solution to this problem can be done in polynomial time for any weight vector.

$$\pi^* \in \arg \max_{\pi \in S_n} \psi_w(v(\pi)) \quad (2)$$

As we will see, the ordered optimization problem captures several well-studied objectives. For instance, when the vector w is uniform on the positions in the ranked value vector, the goal is to maximize total welfare. As another example, note that the max-min welfare problem arises when the vector w only applies weight to the first position of the ranked value vector.

3 Non-increasing Weights

In this section, we examine the class of ordered objectives ψ_w whose weight vector $w \in \mathbb{R}_{\geq 0}^n$ consists of non-increasing entries (i.e., $w_1 \geq w_2 \geq \dots \geq w_n \geq 0$). These objectives, studied by Lesca et al. [14], are often called “fair” Ordered Weighted Averages (OWA). One particularly compelling reason behind the fairness label is the following interpretation: since the largest weights are applied to the smallest utilities, the objective conveys a gradation of concerns for several of the worst-off individuals under a particular assignment. For example, if the planner is concerned about the average welfare attained within quartiles, then she could use a weight vector $w \in \mathbb{R}_{\geq 0}^n$ in which the same (highest) weight is placed on the bottom 25%, a smaller weight is placed on the next 25%, and so on.

Fair OWA Multiagent Assignment

In pursuit of algorithms for the fair OWA objective, Lesca et al. [14] introduced and studied the Fair OWA Multiagent Assignment decision problem defined as follows.

Problem FAIR OWA MULTIAGENT ASSIGNMENT (FOWAMA)

Input: A tuple (V, w, Q) , where $V \in \mathbb{Z}^{n \times n}$ is an integer-valued square matrix, $w \in \mathbb{R}^n$ is a vector of non-negative and non-increasing weights (i.e. $w_1 \geq w_2 \geq \dots \geq w_n \geq 0$), and Q is a positive rational number.

Output: Does there exist an assignment π such that $\psi_w(v(\pi)) \geq Q$?

Although many special instances of the FOWAMA problem are known to admit polynomial times solutions (e.g., the max-min [12], max-welfare [8], and leximin [19]; see also [16], [9] for additional special cases), Lesca et al. [14] showed that the decision variant of the general problem is NP-complete. To better understand which subclasses of the FOWAMA problem admit efficient algorithms, they introduced a parametrized version called the k -FOWAMA problem, where the vector w is assumed to have at most k distinct entries. They also showed that the k -FOWAMA problem lies in the complexity class XP (i.e., the set of problems that can be solved in time $\mathcal{O}(n^{f(k)})$, where f is a computable function) via a somewhat involved recursive argument. In this section, we give a simpler proof of that result. We remark that proving membership within the complexity class FPT (i.e., the set of problems that can be solved in time $f(k) \cdot n^{\mathcal{O}(1)}$, where f is a computable function) remains an open question.³

► **Theorem 1.** *There exists an algorithm that decides the k -FOWAMA problem in time $\mathcal{O}(n^{2k}(kn^2 + T(n)))$, where $T(n)$ is the running time of an algorithm that solves the maximum-weight bipartite matching problem.*

³ An earlier version of this paper had incorrectly claimed to resolve this problem.

At a high level, we prove Theorem 1 by adapting a framework initially outlined by Chakrabarty and Swamy [5] to tackle cost-minimization problems involving vector norms (e.g., minimizing the distance from a facility, or minimizing the makespan for parallel computation). Although the objective function for the FOWAMA problem does not involve a vector norm and our task is a maximization problem, we are still able to apply many of the key ideas, but with different techniques. First, we use a cleverly designed proxy function, which allows us to focus on optimizing a sum of *proxy* values. We then solve a linear programming relaxation of the proxy optimization problem. We direct the reader to Section 3.1 for the details.

3.1 Proof of Theorem 1

Consider the case in which $w_1 \geq w_2 \geq \dots \geq w_n$. For convenience, we also let $w_{n+1} = 0$. An important aspect of the vector w that we examine are the *breakpoints*, namely the positions in the string w for which the entry decreases in magnitude. Formally, we let $\text{POS} = \{\ell \in [n] : w_\ell - w_{\ell+1} > 0\}$ be the set of breakpoints. We also introduce some additional notation to ease our exposition. For any $\ell \in [n]$, we let $\text{next}(\ell) = \min\{i \in \text{POS} \cup \{n+1\} : i > \ell\}$ be the first breakpoint that appears after position ℓ .

As we have alluded to in our earlier discussion on the min-norm framework, our proof will rely on several key insights. We explain each of them step by step. We begin with the observation that our ordered objective admits a convenient decomposition.

Bottom- ℓ Decomposition

For ease of exposition, we abuse notation to let $\psi_\ell(u) := \sum_{i=1}^\ell u_i^\uparrow$ denote the value of the bottom- ℓ , that is the sum of the ℓ smallest entries of a vector u . We first note that the objective $\psi_w(u)$ can be viewed as the linear combination of bottom- ℓ objectives.

▷ **Claim 2.** For every vector $u \in \mathbb{R}_{\geq 0}^n$, we have $\psi_w(u) = \sum_{\ell \in \text{POS}} (w_\ell - w_{\text{next}(\ell)}) \psi_\ell(u)$.

Proof. We use a convenient telescoping sum and recall the definitions of POS and $\psi_\ell(u)$ to derive that

$$\begin{aligned} \psi_w(u) &= \sum_{i=1}^n w_i u_i^\uparrow \\ &= \sum_{i=1}^n \sum_{\ell=i}^n (w_\ell - w_{\ell+1}) u_i^\uparrow \\ &= \sum_{\ell=1}^n (w_\ell - w_{\ell+1}) \sum_{i=1}^\ell u_i^\uparrow \\ &= \sum_{\ell \in \text{POS}} (w_\ell - w_{\text{next}(\ell)}) \psi_\ell(u) \end{aligned}$$

The fact that our objective decomposes into bottom- ℓ objectives is good news, as we already have presented algorithms for the interval in Section 4. The drawback, however, is that we do not have a means of simultaneously optimizing those objectives. Our workaround is to consider replacing these objectives with proxies, which will in turn allow us to do the optimizations all at once. ◀

Proxy Values

As a warmup, we first design a proxy function for maximizing some bottom- ℓ interval. Inspired by our approach for an arbitrary interval, we consider a function that will potentially underestimate values that are too large to fall within the target interval for an optimal solution. For a parameter $\rho \geq 0$, we define the function $h_\rho(z) := \min\{z, \rho\}$. Observe that the function gives the following easy lower bound for $\psi_\ell(u)$.

▷ **Claim 3.** For every value $\rho \geq 0$, and vector $u \in \mathbb{R}_{\geq 0}^n$, we have $\psi_\ell(u) \geq \sum_{i=1}^n h_\rho(u_i) - (n - \ell)\rho$.

Proof. For any value $\rho \geq 0$ and any vector $u \in \mathbb{R}_{\geq 0}^n$, we have $0 \leq h_\rho(u_i) \leq \rho$ and hence

$$\begin{aligned} \sum_{i=1}^{\ell} u_i^\uparrow &\geq \sum_{i=1}^{\ell} h_\rho(u_i^\uparrow) \\ &\geq \sum_{i=1}^{\ell} h_\rho(u_i^\uparrow) + \sum_{i=\ell+1}^n (h_\rho(u_i^\uparrow) - \rho) \\ &\geq \sum_{i=1}^n h_\rho(u_i^\uparrow) - (n - \ell)\rho. \end{aligned} \quad \blacktriangleleft$$

Notice that for the lower bound in Claim 3, a single transformation is being applied to all the entries in the vector u , not just the smallest ℓ entries. Going forward, we want to exploit the oblivious nature of the bound. In the following claim, we also show that given a good guess of the value ρ , the lower bound will in fact be a good estimate for the quantity $\psi_\ell(u)$.

▷ **Claim 4.** Fix a vector $u \in \mathbb{R}_{\geq 0}^n$ and a real number $0 < \alpha \leq 1$. Suppose that there exists a value $\rho \geq 0$ such that $\alpha u_\ell^\uparrow \leq \rho \leq u_\ell^\uparrow$ for some integer $\ell \in [n]$. Then $\sum_{i=1}^n h_\rho(u_i) - (n - \ell)\rho \geq \alpha \cdot \psi_\ell(u)$.

Proof. Let $k = \max\{i \in [n] : u_i^\uparrow \leq \rho\}$. If $i \in \mathbb{N}$ is such that $i \leq k$ and $i \leq \ell$, then $h_\rho(u_i^\uparrow) = u_i^\uparrow \geq \alpha u_i^\uparrow$. Furthermore, if $i \in \mathbb{N}$ is such that $k < i \leq \ell$, then $h_\rho(u_i^\uparrow) = \rho \geq \alpha u_i^\uparrow$. Together, these facts imply that

$$\begin{aligned} \sum_{i=1}^n h_\rho(u_i) &= \sum_{i=1}^{\ell} u_i^\uparrow + (n - \ell)\rho \\ &\geq \alpha \sum_{i=1}^{\ell} u_i^\uparrow + (n - \ell)\rho. \end{aligned} \quad \blacktriangleleft$$

With the insights gleaned from our examination of bottom- ℓ optimization, we now return to the goal of designing a proxy for the weighted objective $\psi_w(u)$. The crucial step we take is replacing each objective $\psi_\ell(u)$ in our decomposition with the lower bounds we established. More precisely, for a vector $\rho \in \mathbb{R}^n$, we define the proxy value of a vector $u \in \mathbb{R}^n$ as follows:

$$\text{prox}_\rho(w; u) := \sum_{\ell \in \text{POS}} (w_\ell - w_{\text{next}(\ell)}) \left[\sum_{i=1}^n h_{\rho_\ell}(u_i) - (n - \ell)\rho_\ell \right]$$

As with our proxy for the bottom- ℓ , we now have an expression that does not depend on the relative order of entries in the vector u . On the other hand, it depends on the advice for each of the breakpoints in the set POS.

The following two claims, which immediately follow from Claim 3 and Claim 4, show that with the appropriate advice, our proxy is a good estimator of the objective $\psi_w(u)$.

21:8 OWA for Bipartite Assignments

▷ **Claim 5.** For every value $\rho \in \mathbb{R}_{\geq 0}^n$, we have $\psi_w(u) \geq \text{prox}_\rho(w; u)$.

▷ **Claim 6.** Fix a vector $u \in \mathbb{R}_{\geq 0}^n$ and a real number $\alpha \leq 1$. Suppose that there exists $\rho \in \mathbb{R}_{\geq 0}^n$ such that $\alpha u_\ell^\uparrow \leq \rho_\ell \leq u_\ell^\uparrow$ for every $\ell \in \text{POS}$. Then $\text{prox}_\rho(w; u) \geq \alpha \cdot \psi_w(u)$

The only step that remains is to treat the proxy as the original objective and attempt to maximize it.

Optimizing the Proxy

Our new goal is to find a perfect matching that maximizes our proxy value, given value some advice ρ . We formulate this as the following integer program.

$$\begin{aligned}
 \max \quad & \sum_{\ell \in \text{POS}} c_\ell \left(\sum_i \sum_j h_{\rho_\ell}(u_{ij}) x_{ij} - (n - \ell) \rho_\ell \right) \\
 \text{s.t.}, \quad & \sum_i x_{ij} = 1 \quad \forall j \in [n] \\
 & \sum_j x_{ij} = 1 \quad \forall i \in [n] \\
 & c_\ell = w_\ell - w_{\text{next}(\ell)} \quad \forall \ell \in \text{POS} \\
 & u_{ij} = v_i(j) \quad \forall i, j \in [n] \\
 & x_{ij} \in \{0, 1\} \quad \forall i, j \in [n]
 \end{aligned}$$

Observe that this is precisely the maximum weight perfect matching problem, in which the weight on each edge in the graph is modified by a function of the advice ρ and distribution w . Hence, we can use any of the appropriate maximum weight matching algorithms to perform our optimization (e.g. the Hungarian algorithm [12]).

Finally, we have all the ingredients to solve our optimization problem.

Proof of Theorem 1. We propose the following procedure. We first guess the $|\text{POS}|$ relevant entries of the vector ρ . Given this guess, we replace all values with appropriate weights given by the proxy function. Finally, we compute the maximum weight matching in the modified problem instance. We ultimately return the matching with the largest objective value. The main subroutine is described in Algorithm 1. We discuss the correctness and running time separately.

■ **Algorithm 1** Optimization for decreasing weights, with guess.

Input: A tuple (V, a, b) of the problem instance; guess $\{\rho_\ell\}_{\ell \in \text{POS}}$

- 1: Replace each weight $V_{i,j}$ with $\sum_{\ell \in \text{POS}} c_\ell \cdot h_{\rho_\ell}(V_{ij})$, yielding a matrix V' .
 - 2: Compute the maximum weight perfect matching τ for the valuation matrix V'
 - 3: Compute the objective $\psi_w(v(\sigma))$.
 - 4: **Return** τ .
-

We first show that for a suitable choice of ρ , Algorithm 1 returns an optimal matching. Let σ^* be an optimal solution to the ordered optimization problem with non-increasing weights and let $\rho_\ell = v(\sigma^*)_\ell^\uparrow$ for every $\ell \in \text{POS}$. Suppose that τ is returned by our algorithm above. Applying Claim 5 and Claim 6, we have

$$\begin{aligned}
\psi_w(v(\sigma^*)) &\geq \psi_w(v(\tau)) \\
&\geq \text{prox}_\rho(w; v(\tau)) \\
&\geq \text{prox}_\rho(w; v(\sigma^*)) \\
&\geq \psi_w(v(\sigma^*))
\end{aligned}$$

Hence, we have $\psi_\ell(v(\tau)) = \psi_\ell(v(\sigma^*))$, as desired. Since our procedure encounters an optimal matching on some iteration and returns the matching with largest objective over all the iterations, it is clear that it returns an optimal solution.

Next, we compute the runtime of our procedure. We first compute the runtime for a single iteration of step 2 for a given sequence $(\rho_{\ell_1}, \rho_{\ell_2}, \dots, \rho_{\ell_{|\text{POS}|}})$. Replacing a single edge weight can be done in time $\mathcal{O}(|\text{POS}|n^2)$. Computing the maximum weight perfect matching σ can be done in time $\mathcal{O}(T(n))$, where $T(n)$ is the running time of a maximum weight perfect matching algorithm. Computing the objective value for the resulting matching can be done in time $\mathcal{O}(n \log n)$; we sort the edges based on weights and multiply each by the appropriate entry in w . Overall the cost for a single iteration is $\mathcal{O}(|\text{POS}|n^2 + T(n))$. It remains to determine the amount of advice needed to guarantee an optimal solution.

▷ **Claim 7.** Our procedure considers at most $\mathcal{O}(n^{2|\text{POS}|})$ possible values of ρ .

Proof. We consider a simple counting argument. We have some non-decreasing sequence $a_1, a_2, \dots, a_{n^2}$ of values and we want to insert $|\text{POS}|$ delimiters between consecutive sequence terms or after the final term a_{n^2} . It is clear that there is a surjection between the possible sequences of ρ and the ways of inserting the delimiters. Indeed, we can associate each delimiter with the largest sequence term that precedes it. The number of ways of inserting the delimiters is

$$\binom{n^2 + |\text{POS}|}{|\text{POS}|} \leq \left(\frac{e(n^2 + |\text{POS}|)}{|\text{POS}|} \right)^{|\text{POS}|} \leq \left(\frac{2en^2}{|\text{POS}|} \right)^{|\text{POS}|},$$

which implies our desired result. ◀

Granting the claim above, and setting that $|\text{POS}| = k$, we conclude that the overall runtime of the algorithm is $\mathcal{O}(n^{2k}(kn^2 + T(n)))$ as desired. ◀

4 Uniform Weights on Intervals

The fair OWA objectives discussed in the previous section would certainly appeal to a planner who is concerned about the worst-off individuals; however, they might not be compelling if the planner is more concerned about other slices of the population. This is especially the case if the planner wants to improve the outcomes experienced by “typical” individuals, those who do not lie within the extreme ends of the economic ladder. For example, a planner might want a strong guarantee for the median utility received in the assignment.

▶ **Example 8 (Median Utility).** Suppose that the planner is interested in obtaining a strong guarantee for the median position on the economic ladder. Consider the following setting with 3 agents and 3 items, where agent valuations are given by rows of a matrix V .

$$V = \begin{bmatrix} 100 & 50 & \epsilon \\ 50 - \epsilon & \epsilon & 0 \\ \epsilon & 0 & 0 \end{bmatrix}$$

In the above matrix, we assume that ϵ is some small positive constant, say 0.01. Note that the sorted value vectors for the max-min solution and max-welfare solutions are $(\epsilon, \epsilon, \epsilon)$ and $(0, \epsilon, 100)$ respectively. On the other hand, the solution that maximizes the median welfare has sorted value vector $(0, 50 - \epsilon, 50)$.

More generally, a planner could be interested in maximizing the average utility derived by the individuals belonging to certain specified brackets within the economic ladder. For instance, she might believe that tending to the welfare of the extremely disadvantaged is not worth compromising on the outcomes for most individuals and thus opt for an assignment that prioritizes say the top 80% of individuals. Alternatively, she could believe that an ideal assignment is one that results in a strong middle class and thus aim to maximize the average utility derived by the middle 50% of individuals.

To capture the kinds of priorities outlined above, we will consider ordered objectives $\psi_w : \mathbb{R}^n \rightarrow \mathbb{R}$ whose weight vectors $w \in \mathbb{R}_{\geq 0}^n$ are such that $w_i = 1$ for all indices i belonging to some specified subset $\mathcal{I} \subseteq [n]$ of the economic ladder and $w_i = 0$ otherwise.

Optimizing a Single Interval

Perhaps, the most natural setting of the subset $\mathcal{I}$ is a contiguous slice of the economic ladder. Note that the associated objective ψ_w would amount to maximizing the average value derived within some interval. Thus, we refer to the ordered optimization task as interval optimization.

Problem INTERVAL OPTIMIZATION

Input: A tuple (V, a, b) , where $V \in \mathbb{R}^{n \times n}$ is a square matrix encoding agent valuations for items (i.e., $V_{ij} = v_i(j)$), and $a, b \in \mathbb{N}$ are natural numbers such that $1 \leq a \leq b \leq n$.

Output: A solution $\pi \in S_n$ to the objective $\max_{\pi \in S_n} \sum_{i \in [a, b]} v(\pi)_i^\uparrow$

Recall that efficient procedures are known for several special cases: when $(a, b) = (1, 1)$, there is the bottleneck algorithm [8]; and when $(a, b) = (1, n)$, there is the Hungarian algorithm [12]. In this section, we complement those results by giving a simple, efficient combinatorial algorithm for maximizing the welfare for any arbitrary interval. In fact, we show that optimizing the more general interval objective is nearly as efficient as optimizing the traditional max-min and max-welfare objectives.

► **Theorem 9.** *There is an algorithm that, on input (V, a, b) , performs interval optimization in time $\mathcal{O}(n^4 + n^2 \cdot T(n + a - 1))$, where $T(n)$ is the running time of an algorithm that solves the maximum-weight perfect matching problem.*

Proof of Theorem 9. We first present a subroutine (Algorithm 2) that maximizes the welfare for the specified interval, given a guess about the maximum value attained within the interval for the optimal solution. We also provide some intuition behind the approach. Although it might be tempting to immediately use the maximum weight matching algorithm, an issue we encounter is potentially over-prioritizing the contribution of values that are too large. With the guess in mind, we can truncate the large values to minimize their contribution to the total weight. We ensure that the truncation is just enough, so that the large value items remain at least as valuable as the best value for the interval. Unfortunately, our approach could still fail even after truncation: the contribution of small values could also be too large. We overcome this issue by assigning highly valuable dummy items to agents who would have otherwise gotten low value items. We can easily reassign those individuals without compromising on our goal.

■ **Algorithm 2** Interval Optimization, with guess.

Input: A tuple (V, a, b) of the problem instance; guess $\rho \in \mathbb{R}$

Output: A one-to-one assignment $\pi \in S_n$

- 1: Construct a new problem instance by adding $a - 1$ dummy items and $a - 1$ dummy individuals. Define the modified valuation matrix $U \in \mathbb{R}^{(n+a-1) \times (n+a-1)}$ as follows:

$$U_{ij} = \begin{cases} \min\{V_{ij}, \rho\} & i, j \in [n] \\ \rho & \text{otherwise} \end{cases}$$

- 2: Find a maximum-weight matching $\pi^* \in S_{n+a-1}$ for the graph with adjacency matrix U .
- 3: Construct a perfect matching $\pi \in S_n$ for the original instance using the following procedure. First, remove from π^* any assignment that involves a dummy item or dummy individual. Then, arbitrarily reassign the unassigned real individuals in $[n]$ to the unassigned real items in $[n]$.
- 4: **Return** π

The algorithm that establishes the theorem is obtained by enumerating over all the possible guessed values. In other words, we iterate ρ over all possible values $\{v_i(j) : i, j \in [n]\}$, using the subroutine to obtain a candidate matching σ , and returning the candidate with the largest value of $\psi_w(v(\sigma))$.

▷ **Claim 10.** Let $\sigma^* \in S_n$ be a one-to-one assignment that maximizes the welfare for the interval with endpoints (a, b) . If $\rho = v(\sigma^*)_b^\uparrow$, then the assignment $\pi \in S_n$ returned by Algorithm 2 also maximizes the welfare on the same interval.

Proof of Claim 10. For ease of exposition, let the function u denote the values derived by individuals in the instance associated with the matrix U , so that $U_{ij} = u_i(j)$ for every pair $i, j \in [n]$. Note that this function will underestimate the value of real items, when compared to the original value function v , (i.e., $u_i(j) = \min\{v_i(j), \rho\}$ for every pair $i, j \in [n]$).

Let $\sigma \in S_{n+a-1}$ be the matching obtained by modifying $\sigma^* \in S_n$ as follows. First, we reassign the $a - 1$ individuals with smallest value (according to the matrix V) to dummy items. Then we assign the $a - 1$ unassigned items to the dummy individuals. We now examine how the items in the reassignment are valued according to the function u . Note that the dummy items and items paired to dummy individuals will have value ρ . Also, note that a real item's altered value will only be truncated to ρ if its original value (according to the matrix V) exceeds ρ ; only $(n - b)$ values will be truncated. Hence, we deduce that

$$\sum_{i=1}^{n+a-1} u_i(\sigma(i)) = \sum_{i=a}^b v(\sigma^*)_i^\uparrow + (n - b + 2(a - 1))\rho.$$

Now consider the map $\pi^* \in S_{n+a-1}$, a maximum-weight perfect matching for the adjacency matrix U (defined in step 2 of Algorithm 2). Let $X \subseteq [n]$ be the set of real individuals that are not assigned to a dummy item by the map π^* . Also, let X_{low} be a set of the $b - a - 1$ individuals in X with the smallest values (according to the matrix U). Since $\pi^* \in S_{n+a-1}$ is optimal and every entry in U is at most ρ , we have

$$\begin{aligned}
\sum_{i \in X_{\text{low}}} u_i(\pi^*(i)) &= \sum_{i=1}^{n+a-1} u_i(\pi^*(i)) - \sum_{i \in [n+a-1] \setminus X_{\text{low}}} u_i(\pi^*(i)) \\
&\geq \sum_{i=1}^{n+a-1} u_i(\pi^*(i)) - (n-b+2(a-1))\rho \\
&\geq \sum_{i=1}^{n+a-1} u_i(\sigma(i)) - (n-b+2(a-1))\rho \\
&= \sum_{i=a}^b v(\sigma^*)_i^\uparrow.
\end{aligned}$$

This, along with the following chain of inequalities, would imply our desired result.

$$\begin{aligned}
\sum_{i=a}^b v(\sigma^*)_i^\uparrow &\geq \sum_{i=a}^b v(\pi)_i^\uparrow \\
&\geq \sum_{i \in X_{\text{low}}} v_i(\pi^*(i)) \\
&\geq \sum_{i \in X_{\text{low}}} u_i(\pi^*(i))
\end{aligned}$$

The first inequality follows from the optimality of $\sigma^* \in S_n$ and the third follows from the fact that the function u was constructed to underestimate the function v . Thus, it remains to justify the second inequality. Suppose there are d real individuals in $[n]$ who were assigned a dummy item under π^* . When we modify π^* to get π , an individual in X_{low} will now be among the $(b-a+1) + d \leq b$ individuals with smallest value (according to the matrix U). Therefore, the sum of values for the individuals in X_{low} (a set of size $b-a-1$) is at most the sum $\sum_{i=a}^b v(\pi)_i^\uparrow$. $\triangleleft$

Claim 10 implies that the assignment π_ρ that has the highest value for the interval for all values of ρ maximizes the social welfare of the interval $[a, b]$. Note that we can obtain the right value of ρ by iterating over all the possible values in the matrix V . Having established correctness, we turn our attention to the running time. Note that steps 1 and 3 of Algorithm 2 can be done in time $\mathcal{O}(n^2)$ and step 2 can be done in time $(n+a-1)$. Note also that computing our objective value for the returned matching can also be done in time $\mathcal{O}(n^2)$. Thus, the cost for a single iteration is $\mathcal{O}(n^2 + T(n+a-1))$. Since we iterate over n^2 possible values of ρ , the overall runtime is $\mathcal{O}(n^4 + n^2 \cdot T(n+a-1))$. $\blacktriangleleft$

► **Remark 11.** There might not always be a unique solution to a given interval optimization task. For instance, it is possible to have two solutions that optimize the same interval, but the total social welfare is higher in one of them. Note that our algorithm outlined above makes no distinction between these optimal solutions and returns one of them arbitrarily.

Optimizing a Union of Intervals

Next, we consider the more general setting in which the subset $\mathcal{I}$ is the union of k disjoint intervals of the economic ladder. We call the ordered optimization task k -intervals optimization.

Problem k -INTERVALS OPTIMIZATION

Input: A tuple $(V, \mathcal{I} = \cup_{i=1}^k I_r)$, where $V \in \mathbb{R}^{n \times n}$ is an integer-valued square matrix and $I_1 = [a_1, b_1], I_2 = [a_2, b_2], \dots, I_k = [a_k, b_k]$ are k disjoint intervals on the economic ladder such that $b_r < a_{r+1}$ for every $r \in [k-1]$.

Output: A solution $\pi \in S_n$ to the objective $\max_{\pi \in S_n} \sum_{r=1}^k \sum_{i \in I_r} v(\pi)_i^\uparrow$

For the above optimization problem, there is a simple algorithm with modest approximation guarantees when the set $\mathcal{I}$ is comprised of intervals of the same size: we can run the interval optimization algorithm on instances (V, a_r, b_r) for each $r \in [k]$ and return the best assignment. Since one of the intervals must account for at least $1/k$ of the optimal value, a naive averaging argument would have suggested an approximation of $\mathcal{O}(k)$; however, we can show that the approach would actually result in a logarithmic multiplicative approximation. Our proof can be found in Appendix A.

► **Theorem 12.** *Let $(V, \mathcal{I} = \cup_{r=1}^k I_r)$ be an instance of the k -intervals optimization problem such that for some real number $\alpha > 0$ and every $r \in [k]$,*

$$\frac{1}{r} (|I_1| + |I_2| + \dots + |I_r|) \geq \alpha \cdot |I_r|.$$

There exists an interval $[a, b]$ such that the following holds. Any solution $\sigma \in S_n$ to the interval optimization problem on instance (V, a, b) is an $\mathcal{O}\left(\frac{\log k}{\alpha}\right)$ -approximation for the k -intervals optimization problem on instance $(V, \mathcal{I})$. In particular, if $|I_1| = |I_2| = \dots = |I_k|$, the solution σ is a $\mathcal{O}(\log k)$ -approximation.

The above theorem highlights a potential upside of using interval optimization as a tool for (approximately) solving more ordered objectives. In the next section, we explore how well this approach fares on more general ordered objectives.

5 General Weights

Although the ordered objectives ψ_w discussed so far are arguably quite natural from a fairness perspective, we recognize that a central planner could want to work with a more complicated objective. This is especially the case if she attempts to simultaneously reconcile the concerns of multiple stakeholders. Thus, we consider ordered optimization with general weights.

Problem GENERAL ORDERED OPTIMIZATION

Input: A tuple (V, w) , where $V \in \mathbb{R}^{n \times n}$ is an integer-valued square matrix, $w \in \mathbb{R}^n$ is an arbitrary vector of non-negative weights

Output: A solution $\pi \in S_n$ to the objective $\max_{\pi \in S_n} \sum_{i=1}^n w_i v(\pi)_i^\uparrow$

Since the FOWAMA problem discussed in Section 3 is already known to be NP-complete [14], it is highly unlikely that an efficient algorithm will exist when the weights $w \in \mathbb{R}^n$ are allowed to be arbitrary. Nevertheless, it is still worthwhile to consider simple heuristic algorithms that could lead to reasonable approximations.

A possible approach, inspired by works on simultaneous optimization [7, 13], is to consider how well a solution for one objective approximates another. In the previous section, we showed that arbitrary interval optimization is efficient. One might reasonably hope that a suitable interval could approximate a more complicated ordered objective. For instance, it could be the case that a max-welfare or max-min approach (or some other interval optimization) captures enough of the objective's complexity. We show, however, that a logarithmic multiplicative error is about the best one can hope to achieve using a single interval for approximation for general instances. The proof can be found in Appendix B.

► **Theorem 13.** *There exists an instance (V, w) of the general ordered optimization problem such that the following holds. For any interval J , there exists an assignment $\pi_J \in S_n$ that optimizes interval J such that*

$$\frac{\max_{\pi \in S_n} \psi_w(v(\pi))}{\psi_w(v(\pi_J))} = \Omega\left(\frac{\log n}{\log \log n}\right).$$

Recall that there could potentially be many assignments that optimize a given interval and we do not make any assumption about the one returned by our algorithm for interval optimization (see Remark 11). In the worst-case, our algorithm returns an assignment that yields the above lower bound.

6 Conclusion

In this work, we provide efficient exact and approximation algorithms for several ordered weighted average (OWA) operators that could appeal to a central planner with fairness considerations. Our work also motivates the study of OWA operators with arbitrary weights, which may arise as the result of aggregating the private OWA functions held by stakeholders. An interesting direction is to design and study aggregation functions for these privately held OWA functions. Consider the specific case in which each agent has some interval that she cares about. Is there an aggregation rule that motivates the agent to truthfully report the interval that they care about? Is there any such rule which is not some variation of dictatorship? An additional fascinating direction is determining how well OWA operators can capture or approximate other notions of fairness (e.g. Nash social welfare).

References

- 1 Fateme Abbasi, Sandip Banerjee, Jarosław Byrka, Parinya Chalermsook, Ameet Gadekar, Kamyar Khodamoradi, Dániel Marx, Roohani Sharma, and Joachim Spoerhase. Parameterized Approximation Schemes for Clustering with General Norm Objectives. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1377–1399, 2023. doi:10.1109/FOCS57990.2023.00085.
- 2 Atila Abdulkadiroğlu, Parag A Pathak, and Alvin E Roth. The new york city high school match. *American Economic Review*, 95(2):364–367, 2005.
- 3 Hyung-Chan An and Ola Svensson. Recent developments in approximation algorithms for facility location and clustering problems. In *Combinatorial Optimization and Graph Algorithms: Communications of NII Shonan Meetings*, pages 1–19. Springer, 2017.
- 4 Nick Arnosti and Peng Shi. Design of lotteries and wait-lists for affordable housing allocation. *Management Science*, 66(6):2291–2307, 2020. doi:10.1287/MNSC.2019.3311.
- 5 Deeparnab Chakrabarty and Chaitanya Swamy. Approximation algorithms for minimum norm and ordered optimization problems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 126–137, Phoenix AZ USA, June 2019. ACM. doi:10.1145/3313276.3316322.
- 6 Shichuan Deng and Qianfan Zhang. Ordered k-Median with Outliers. In Amit Chakrabarti and Chaitanya Swamy, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2022)*, volume 245 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 34:1–34:22, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPICS.APPROX/RANDOM.2022.34.
- 7 Ashish Goel and Adam Meyerson. Simultaneous Optimization via Approximate Majorization for Concave Profits or Convex Costs. *Algorithmica*, 44(4):301–323, April 2006. doi:10.1007/S00453-005-1177-7.
- 8 O Gross. *The bottleneck assignment problem*. Rand, 1959.

- 9 S.K. Gupta and A.P. Punnen. k-sum optimization problems. *Operations Research Letters*, 9(2):121–126, 1990.
- 10 Swati Gupta, Jai Moondra, and Mohit Singh. *Balancing Notions of Equity: Trade-offs Between Fair Portfolio Sizes and Achievable Guarantees*, pages 1136–1165. Society for Industrial and Applied Mathematics, 2025. doi:10.1137/1.9781611978322.33.
- 11 Thomas Kesselheim, Marco Molinaro, and Sahil Singla. Supermodular approximation of norms and applications. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1841–1852, 2024. doi:10.1145/3618260.3649734.
- 12 Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.
- 13 A. Kumar and J. Kleinberg. Fairness measures for resource allocation. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 75–85, 2000.
- 14 Julien Lesca, Michel Minoux, and Patrice Perny. The fair owa one-to-one assignment problem: Np-hardness and polynomial time special cases. *Algorithmica*, 81(1):98–123, January 2019. doi:10.1007/S00453-018-0434-5.
- 15 Albert W. Marshall, Ingram Olkin, and Barry C. Arnold. *Inequalities: Theory of Majorization and Its Applications*. Springer Series in Statistics. Springer New York, New York, NY, 2011.
- 16 M. Minoux. Solving combinatorial problems with combined min-max-min-sum objective and applications. *Math. Program.*, 45(1-3):361–372, August 1989. doi:10.1007/BF01589111.
- 17 John Rawls. A theory of justice. *Cambridge (Mass.)*, 1971.
- 18 Alvin E Roth. The origins, history, and design of the resident match. *Jama*, 289(7):909–912, 2003.
- 19 P.T. Sokkalingam and Y.P. Aneja. Lexicographic bottleneck combinatorial problems. *Operations Research Letters*, 23(1):27–33, 1998. doi:10.1016/S0167-6377(98)00028-5.
- 20 John A Weymark. Generalized gini inequality indices. *Mathematical Social Sciences*, 1(4):409–430, 1981. doi:10.1016/0165-4896(81)90018-4.
- 21 Ronald R Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on systems, Man, and Cybernetics*, 18(1):183–190, 1988. doi:10.1109/21.87068.
- 22 Ronald R. Yager, Janusz Kacprzyk, Gleb Beliakov, and Janusz Kacprzyk, editors. *Recent Developments in the Ordered Weighted Averaging Operators: Theory and Practice*, volume 265 of *Studies in Fuzziness and Soft Computing*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

A Missing Proofs from Section 4

► **Theorem 12.** Let $(V, \mathcal{I} = \cup_{r=1}^k I_r)$ be an instance of the k -intervals optimization problem such that for some real number $\alpha > 0$ and every $r \in [k]$,

$$\frac{1}{r} (|I_1| + |I_2| + \dots + |I_r|) \geq \alpha \cdot |I_r|.$$

There exists an interval $[a, b]$ such that the following holds. Any solution $\sigma \in S_n$ to the interval optimization problem on instance (V, a, b) is an $\mathcal{O}\left(\frac{\log k}{\alpha}\right)$ -approximation for the k -intervals optimization problem on instance $(V, \mathcal{I})$. In particular, if $|I_1| = |I_2| = \dots = |I_k|$, the solution σ is a $\mathcal{O}(\log k)$ -approximation.

Proof of Theorem 12. For ease of exposition we will abuse notation, letting $\psi(I; \cdot)$ for a given subset $I \subseteq [n]$ denote the OWA operator $\psi_w(\cdot)$ where $w_i = 1$ if $i \in I$ and $w_i = 0$ otherwise. We also remind the reader that for the the k -intervals optimization problem, the intervals are ordered so that I_1 is a higher region on the economic ladder (and hence corresponds to individuals with higher utility) than interval I_2 , I_2 is higher than I_3 , and so on.

21:16 OWA for Bipartite Assignments

Let σ be an assignment that optimizes the union of intervals $\mathcal{I} = \cup_{r=1}^k I_r$. For each index $r \in [k]$, let σ_r be the assignment that optimizes the interval I_r . Suppose that $f(k)$ is a function such that for every $r \in [k]$, we have $\psi(\mathcal{I}; v(\sigma)) > f(k) \cdot \psi(\mathcal{I}; v(\sigma_r))$. Then, observe that for every $r \in [k]$, we have

$$\begin{aligned} \psi(\mathcal{I}; v(\sigma)) &> f(k) \cdot \psi(\mathcal{I}; v(\sigma_r)) \\ &\geq f(k) \cdot (\psi(I_1; v(\sigma_r)) + \psi(I_2; v(\sigma_r)) + \cdots + \psi(I_r; v(\sigma_r))) \\ &\geq f(k) \cdot \frac{(|I_1| + |I_2| + \cdots + |I_r|)}{|I_r|} \cdot \psi(I_r; v(\sigma_r)) \\ &\geq f(k) \cdot (\alpha r) \cdot \psi(I_r; v(\sigma_r)) \end{aligned}$$

The third inequality follows from recalling the fact that every value in an interval that precedes I_r is at least as large as the maximum value within the interval I_r and comparing the average value attained within each interval. Now, rearranging our final expression and summing over r , we see that

$$\begin{aligned} \sum_{r=1}^k \psi(I_r; v(\sigma_r)) &< \left(\frac{1}{f(k)} \sum_{r=1}^k \frac{1}{\alpha r} \right) \psi(\mathcal{I}; v(\sigma)) \\ &\leq \frac{C \cdot \log k}{\alpha \cdot f(k)} \cdot \psi(\mathcal{I}; v(\sigma)) \end{aligned}$$

for some absolute constant $C > 0$. We claim that $f(k) \geq \frac{C \log k}{\alpha}$ would lead to a contradiction. On one hand, we would have $\sum_{r=1}^k \psi(I_r; v(\sigma_r)) < \psi(\mathcal{I}; v(\sigma))$. Yet, the optimality of each assignment σ_r implies that

$$\psi(\mathcal{I}; v(\sigma)) = \sum_{r=1}^k \psi(I_r; v(\sigma)) \leq \sum_{r=1}^k \psi(I_r; v(\sigma_r))$$

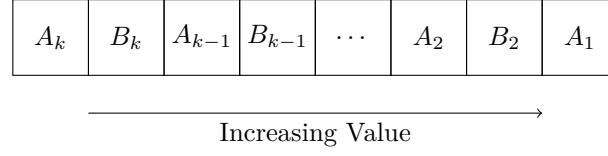
Hence, we must have $f(k) < \frac{C \log k}{\alpha}$. Moreover, there must exist an interval I_r such that $\psi(\mathcal{I}; v(\sigma)) \leq \frac{C \log k}{\alpha} \cdot \psi(\mathcal{I}; v(\sigma_r))$, as desired. $\blacktriangleleft$

B Missing Proofs from Section 5

► **Theorem 13.** *There exists an instance (V, w) of the general ordered optimization problem such that the following holds. For any interval J , there exists an assignment $\pi_J \in S_n$ that optimizes interval J such that*

$$\frac{\max_{\pi \in S_n} \psi_w(v(\pi))}{\psi_w(v(\pi_J))} = \Omega\left(\frac{\log n}{\log \log n}\right).$$

Proof of Theorem 13. We will construct an instance involving the union $\mathcal{I}$ of k -disjoint intervals, where k is later defined in terms of the number of agents n . At a high level, we design the valuation matrix V in such a way that each specified interval $I \in \mathcal{I}$ will correspond to a fixed set of agents A_I . Each agent in A_I will have a unique item they prefer most; their value for this item will significantly exceed that of any other item. In order to optimize the union $\mathcal{I}$, an assignment must ensure that for each $I \in \mathcal{I}$, every agent in A_I gets her unique preferred item. An assignment that optimizes some other region of the ladder may fail to provide maximum utility for some set A_I , resulting in a significant decrease in the contribution to the target objective. In the paragraphs that follow, we discuss the aspects of our construction and analyze how well a single interval is able to approximate the union objective.



■ **Figure 1** Positions on the economic ladder induced by the valuation matrix V .

Construction

We construct an instance with $n = 1 + 2 \sum_{r=1}^k k^{2r-2}$ agents. These agents are partitioned into sets $A_1, B_2, A_2, \dots, B_k, A_k$ such that $|A_1| = 1$ and $|B_r| = |A_r| = k^{2r-2}$ for every $r \in \{2, \dots, k\}$. For ease of exposition, we also let $B_1 = A_1$. Let f be an injective function such that for each $r \in \{1, \dots, k\}$, every agent $b \in B_r$ is mapped to some agent $f(b) \in A_r$. We design a valuation matrix $V \in \mathbb{R}^{n \times n}$ as follows:

$$V_{ij} = \begin{cases} 1/k^{2r-3} & i \in B_r, j = f(i) \\ 1/k^{2r-2} & i \in B_r, j \neq f(i) \\ 1/k^{2r-2} & i \in A_r, j = i \\ 1/k^{2r-1} & i \in A_r, j \neq i \end{cases}$$

By design of the function f and matrix V , the following claims are immediate.

▷ **Claim 14.** For every $r \in \{1, \dots, k\}$, the item most preferred by an agent $b \in B_r$ is also the item most preferred by the agent $f(b) \in A_r$.

▷ **Claim 15.** The following hold for any matching $\sigma \in S_n$:

- For any $r \in \{2, \dots, k\}$, and any pair $(a, b) \in A_r \times B_r$, we have $v(\sigma(b)) \geq v(\sigma(a))$
- For any $r \in \{2, \dots, k\}$, and any pair $(a, b) \in A_{r-1} \times B_r$, we have $v(\sigma(a)) \geq v(\sigma(b))$.

As a result of Claim 15, we can treat each set of agents A_r or B_r as occupying a fixed interval in the economic ladder, regardless of the chosen assignment. Moreover, the ladder can be viewed as an interleaving of the sets $A_1, B_2, A_2, B_3, A_3, \dots, B_k, A_k$ (see Figure 1).

Analysis

Let $I_1, I_2, \dots, I_k$ be the intervals respectively corresponding to the agents in the sets $A_1, A_2, \dots, A_k$ and let $\mathcal{I} = \cup_{r=1}^k I_r$ be the union of those intervals. We first analyze the assignment that optimizes the union $\mathcal{I}$. For ease of exposition we will abuse notation, letting $\psi(I; \cdot)$ denote the OWA operator $\psi_w(\cdot)$ where $w_i = 1$ if $i \in I$ and $w_i = 0$ otherwise for each subset $I \subset [n]$. Recall that for each $r \in \{1, \dots, k\}$, the unique item preferred most by agent $a \in A_r$ has value $1/k^{2r-2}$. An assignment $\pi \in S_n$ that gives each agent in A_r their most preferred item would maximize the value derived for the union $\mathcal{I}$. Computing the objective value $\psi(\mathcal{I}; v(\pi))$, we therefore get

$$\psi(\mathcal{I}; v(\pi)) = \sum_{r=1}^k |A_r| \cdot (1/k^{2r-2}) = k$$

We now analyze how well a single interval $J \subseteq \{1, \dots, n\}$ is able to approximate this optimal value. For each J , we will consider an assignment π_J that optimizes the interval J , but whose contribution to the objective $\psi(\mathcal{I}, v(\pi_J))$ is worst-case (in the sense that as

21:18 OWA for Bipartite Assignments

many agents in $\cup_{r=1}^k B_r$ receive their unique favorite item as possible). Since π_J is worst case, we claim there is at most one set A_r whose agents are assigned their most preferred items. Indeed, this follows from Claim 14 and considering several cases:

- If J corresponds to a region of the ladder that contains all the agents in B_r and all the agents in A_r for some r , then modifying the assignment π_J to give each agent in B_r their most preferred item (instead of the agents in A_r) does not decrease the value of $\psi(J; v(\pi_J))$.
- If J corresponds to a region of the ladder that contains some of the agents in B_r and none of the agents in A_r for some r , then modifying the assignment π_J to give each agent in B_r their unique favorite item does not decrease the value of $\psi(J; v(\pi_J))$.
- If J corresponds to a region of the ladder that contains at most $|B_r| - 1$ agents from B_r and some of the agents in A_r , then π_J could potentially have given some of the agents in A_r their most preferred items.

Since there is at most one set A_r whose agents are assigned their most preferred items, the objective value $\psi(\mathcal{I}; v(\pi_J))$ is

$$\begin{aligned} \psi(\mathcal{I}; v(\pi_J)) &\leq |A_r| \cdot (1/k^{2r-2}) + \sum_{i \neq r} |A_i| \cdot (1/k^{2i-1}) \\ &= 1 + (k-1)/k \\ &= 2 - 1/k \end{aligned}$$

Comparing to the assignment π , we see that

$$\frac{\psi(\mathcal{I}; v(\pi))}{\psi(\mathcal{I}; v(\pi_J))} \geq \frac{k}{2 - 1/k} \geq \frac{k}{2}$$

We use the following claim to complete our proof.

▷ **Claim 16.** $k = \Omega\left(\frac{\log n}{\log \log n}\right)$

Proof. Recall that the number of agents is

$$n = 1 + 2 \sum_{r=1}^{k-1} k^{2r} = \frac{2k^{2k} - k^2 - 1}{k^2 - 1}$$

By taking the last term of the summation and assuming that $k > 2$, it follows that

$$k^{2k-2} \leq n \leq \frac{2k^{2k}}{k^2/2} = 4k^{2k-2}.$$

Rearranging terms (and taking k sufficiently large), we derive that

$$\frac{\log n - \log 4}{\log k} \leq 2k - 2 \leq \frac{\log n}{\log k} \quad (3)$$

Note that the second inequality in (3) implies that $\log(2k - 2) + \log \log k \leq \log \log n$. For n sufficiently large, we therefore have

$$\log k < \log(2k - 2) + \log \log k \leq \log \log n$$

Applying the above inequality to the first inequality in (3), we see that

$$2k - 2 \geq \frac{\log n - \log 4}{\log k} \geq \frac{\log n - \log 4}{\log \log n}$$

which implies that $k = \Omega\left(\frac{\log n}{\log \log n}\right)$

◁

◀

Revision Notice

This is a revised version of the eponymous paper that appeared in the proceedings of FORC 2025 (LIPIcs, volume 329, <https://www.dagstuhl.de/dagpub/978-3-95977-367-6>, published in June, 2025), in which it was erroneously claimed that the fair OWA problem belongs to the class FPT. This claim is withdrawn and replaced by a simple proof of the fact that fair OWA belongs to the class XP in this version (Theorem 1). All references to this error in the introduction and abstract have also been updated accordingly.

Dagstuhl Publishing – September 02, 2025.