# The Conference Paper Assignment Problem: Using Order Weighted Averages to Assign Indivisible Goods

**Jing Wu Lian**
UNSW Sydney
Sydney, Australia
Lianjingwu@gmail.com

**Nicholas Mattei**
IBM Research AI
New York, USA
n.mattei@ibm.com

**Renee Noble**
Data61, CSIRO
Sydney, Australia
renee.noble@data61.csiro.au

**Toby Walsh**
Data61, UNSW Sydney, and TU Berlin
Sydney, Australia
toby.walsh@data61.csiro.au

## Abstract

We propose a novel mechanism for solving the assignment problem when we have a two sided matching problem with preferences from one side (the agents/reviewers) over the other side (the objects/papers) and both sides have capacity constraints. The assignment problem is a fundamental in both computer science and economics with application in many areas including task and resource allocation. Drawing inspiration from work in multi-criteria decision making and social choice theory we use order weighted averages (OWAs), a parameterized class of mean aggregators, to propose a novel and flexible class of algorithms for the assignment problem. We show an algorithm for finding an Σ-OWA assignment in polynomial time, in contrast to the NP-hardness of finding an egalitarian assignment. We demonstrate through empirical experiments that using Σ-OWA assignments can lead to high quality and more fair assignments.

## Introduction

Assigning indivisible objects to multiple agents is a fundamental problem in many fields including computer science, economics and operations research. Algorithms for matching and assignment are used in a variety of application areas including allocating runways to airplanes, residents to hospitals, kidneys to patients (Dickerson, Procaccia, and Sandholm 2014), students to schools (Budish and Cantillon 2012), assets to individuals in a divorce, jobs to machines, and tasks to cloud computing nodes (Manlove 2013). Understanding the properties of the underlying algorithms is an important aspect to ensuring that all participating agents are happy with their allocations and do not attempt to misrepresent their preferences; a key area of study for computational social choice (Brandt et al. 2016).

An area that is near to many academics' hearts is the problem of allocating papers to referees for peer review. Grant, journal, and conference reviewing has significant impact on the careers of scientists. Ensuring that papers and proposals are reviewed by the most qualified/interested referees is part of ensuring that objects are treated properly and all participants support the outcome of the processes. The importance of and methods for improving peer review have been proposed across the sciences (Merrifield and Saari 2009).

Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

There are a number of ways one can improve the quality of peer review (Price and Flach 2016). First is to ensure that reviewers are not incentivized to misreport their reviews for personal gain. Along this line there has been significant interest recently in strategyproof mechanisms for peer review (Aziz et al. 2016). Unfortunately, the method that we discuss in this paper is not strategyproof. Another way is to ensure that reviewers are competent to provide judgements on the papers they are assigned. The Toronto Paper Matching System (Charlin and Zemel 2013) is designed to improve the process from this paper-centric model. A third alternative, and the one we focus on in this study, is ensuring that reviewers are happy with the diversity and quality of papers they are asked to review (Ahmed, Dickerson, and Fuge 2017).

Formally, we study the Conference Paper Assignment Problem (CPAP) (Goldsmith and Sloan 2007) which is a special case of the Multi-Agent Resource Allocation Problem (MARA) (Bouveret, Chevaleyre, and Lang 2016), and propose a novel assignment, the Σ-OWA assignment. In the CPAP setting we have a two-sided market where on one side the agents/reviewers have preferences over the other side, the objects/papers, and both sides have (possibly infinite) upper and lower capacities. A fundamental tension in assignment settings is the tradeoff between maximizing the social welfare, also known as the utilitarian maximal assignment and the Rawlsian (Rawls 1971) fairness concept of maximizing the utility of the worst off agent, known as the egalitarian maximal assignment. These two ideas are incompatible optimization objectives and diverge in a computational sense as well: computing the utilitarian assignment for additive utilities can be done in polynomial time, while computing the egalitarian assignment is NP-complete (Demko and Hill 1988). This, perhaps, could be the reason that implementers of large conference paper assignment software often opt for utilitarian assignments, as is supposedly the case for EasyChair (Garg et al. 2010).[1] However, it is also not clear if an egalitarian assignment is desirable for CPAP.

**Contributions.** We establish a motivation for using OWA

[1] This is technically unsubstantiated as when the authors contacted EasyChair to understand the assignment process we were told, *"We do not provide information on how paper assignment in EasyChair is implemented. The information in Garg et.al. may be incorrect or out of date - none of the authors worked for EasyChair, they also had no access to the EasyChair code."*

vectors in the assignment setting and define a novel notion of allocation, the Σ-OWA assignment. We give an algorithm to compute an Σ-OWA maximal assignment in polynomial time and we show that the Σ-OWA objective generalizes the utilitarian objective. We show that Σ-OWA assignments satisfy a notion of Pareto optimality w.r.t. the pairwise comparisons of the objects by the agents. We implement an algorithm for Σ-OWA assignments and perform experiments on real world conference paper assignment data to show that the Σ-OWA objective can lead to more fair allocations.

## Preliminaries

From here we will use the more general notation agents/objects to describe our setting. In assignment settings each agent provides their preference over the objects as a reflexive, and transitive preference relation (incomplete weak order) over the set of objects, $\succsim_i$. We do not assume that $\succsim_i$ is complete; it is possible that some agents may have conflicts of interest or have no preference for a particular object; this assumption is often called "having unacceptable objects" in the literature (Manlove 2013).

Many actual CPAP settings have a fixed number of equivalence classes into which agents are asked to place the objects (Mattei and Walsh 2013). We assume that the number of equivalence classes (ranks) of objects are given as input to the problem and agents tell us within which rank each objects belong. Agents also provide a decreasing utility value for each rank[2]. Our main result can be extended to the case where the number of equivalence classes is not fixed.

Formally, the CPAP problem is defined by $(N, O, \succsim, \mathbf{u}, \Delta)$: a set of $n$ agents $N = \{a_1, \ldots, a_n\}$; a set of $m$ objects $O = \{o_1, \ldots, o_m\}$; for each $i \in N$, a reflexive and transitive preference relation (weak order) over the set of objects, $\succsim_i$, divided into $\Delta$ equivalence classes (ranks); and for each $i \in N$ a utility vector $\mathbf{u}_i$ of length $\Delta$ which assigns a decreasing utility $\mathbf{u}_i(k) \to \mathbb{R}$ for each $k \in [1, \Delta]$, i.e., $\mathbf{u}_i(1) > \mathbf{u}_i(2) > \ldots > \mathbf{u}_i(\Delta)$. Let $r_i(j)$ be the rank of object $j$ for $i$ and $\mathbf{u}_i(r_i(j))$ denote the value of $i$ for $j$.

### Side Constraints and Feasible Assignments

We include two practical constrains in our model, making it more general than the standard MARA or CPAP problems studied in computer science (Bouveret, Chevaleyre, and Lang 2016): upper and lower capacities on both the agents and objects. **Agent Capacity:** each agent $i \in N$ has (possibly all equal) upper and lower bound on their capacity, the number of objects they can be allocated, $c^N_{min}(i)$ and $c^N_{max}(i)$. **Object Capacity:** each object $j \in O$ has a (possibly all equal) upper and lower bound on the number of agents assigned to it, $c^O_{min}(j)$ and $c^O_{max}(j)$, respectively.

We can now define a feasible assignment $A$ for an instance $(N, O, \succ, \mathbf{u}, \Delta)$. For a given assignment $A$, let $A(i,:)$ denote the set of objects assigned to agent $i$ in $A$, let $A(:, j)$ denote the set of agents assigned to object $j$, and let $|\cdot|$ denote the size (number of elements) of a set or vector. A feasible assignment $A$ must obey: $[\forall i \in N : c^N_{min}(i) \leq |A(i,:)| \leq c^N_{max}(i)] \wedge [\forall j \in O : c^O_{min}(j) \leq |A(:,j)| \leq c^O_{max}(j)]$. We write the set of all feasible assignments for an instance as $\mathbb{A}(N, O, \succ, \mathbf{u}, \Delta)$.[3]

### Individual Agent Evaluation

We first formalize how an individual agent evaluates their assigned objects. Each feasible assignment $A \in \mathbb{A}$ gives rise to a *signature vector* for each agent $i \in N$; intuitively the signature vector is the number of objects at each rank assigned to $i$. Formally let $\sigma_i(A) = (\sigma_{i,1}(A), \ldots, \sigma_{i,\Delta}(A))$ where $\sigma_{i,l}(A) = |\{j \in A(i,:) | r_i(j) = l\}|$ for each $l \in [1, \ldots, \Delta]$.

**Lexicographic:** Agent $i$ lexicographically prefers $A$ to $B$ if $\sigma_i(A)$ comes before $\sigma_i(B)$ in the lexicographic order. That is, there is an index $1 \geq l \geq \Delta$ such that for all $k > l$ we have $\sigma_{i,k}(A) = \sigma_{i,k}(B)$ and $\sigma_{i,l}(A) > \sigma_{i,l}(B)$; i.e., $i$ receives at least one more paper of a higher rank in $A$ than in $B$. The lexicographic relation over vectors has a long history in the assignment literature (Fishburn 1974).

**Additive Utility:** Agent $i$ prefers $A$ to $B$ if he has more additive utility for the objects assigned to him in $A$ than in $B$. Formally, (abusing notation) $\mathbf{u}_i(A) = \sum_{j \in A(i,:)} \mathbf{u}_i(r_i(j)) > \sum_{j \in B(i,:)} \mathbf{u}_i(r_i(j))$, or an alternative formulation using the dot product, $\mathbf{u}_i(A) = \mathbf{u}_i \cdot \sigma_i(A) > \mathbf{u}_i \cdot \sigma_i(B)$.

For indivisible (discrete) objects the lexicographic relation can be modeled by the additive utility relation by setting the agent utilities to high enough values. Formally, if the utility for rank $i < j$ is $\mathbf{u}(i) > \mathbf{u}(j) \cdot m$ then the lexicographic and additive utility relations are the same, i.e., no matter how many additional objects of rank $j$ the agent receives, one more object of rank $i$ is more preferred.

### Overall Assignment Evaluation

There are several optimization objectives for assignments can be considered; we limit our discussion to the two classical notions below. Garg et al. (2010) and Bouveret and Lemaître (2016) discuss additional features for the CPAP and MARA settings, respectivly.

**Utilitarian Social Welfare Maximal Assignment:** We maximize the total social welfare over all the agents. A utilitarian assignment is $\arg\max_{A \in \mathbb{A}} \sum_{i \in N} \sum_{j \in A(i,:)} \mathbf{u}_i(r_i(j)) = \arg\max_{A \in \mathbb{A}} \sum_{i \in N} \mathbf{u}_i \cdot \sigma_i(A)$.

**Egalitarian Social Welfare Maximal Assignment:** We want to enforce the Rawlsian notion of fairness by ensuring the worst off is as happy as possible. An egalitarian assignment is $\arg\max_{A \in \mathbb{A}} \min_{i \in N} \sum_{j \in A(i,:)} \mathbf{u}_i(r_i(j)) = \arg\max_{A \in \mathbb{A}} \min_{i \in N} \mathbf{u}_i \cdot \sigma_i(A)$.

In the discrete MARA and CPAP setting where objects are not divisible, the problem of finding an egalitarian assignment is NP-hard (Demko and Hill 1988) while finding a utilitarian assignment can be done in polynomial time (Bouveret, Chevaleyre, and Lang 2016).

There are a number of measures of inequality or fairness in the econometric literature, two of the most popular are the

[2] We assume that agents can give any utilities as input. However, often the utilities are restricted to be the same, i.e., Borda utilities in conference paper bidding, or come from some fixed budget, i.e., bidding fake currency as in course allocation at Harvard (Budish and Cantillon 2012).

[3] We will omit the arguments when they are clear from context.

Gini Index and the Hoover (Robin Hood) Index. Let $i, j$ be agents in the assignment and let $\overline{u}$ be the mean utility over all agents. The **Gini Index** measures the ration of the area that lies between the line of equality and the Lorenz curve, $G = \frac{\sum_i \sum_j |u_i - u_j|}{2n \sum_i u_i}$. The **Hoover Index** measures the amount of utility that needs to be transferred to the worse off agents in order to achieve full equality, $H = \frac{1}{2} \frac{\sum_i |u_i - \overline{u}|}{\sum_i u_i}$. Minimizing the inequality directly in the assignment setting is NP-hard for either of these measures (Schneckenburger, Dorn, and Endriss 2017). In our empirical section we will use these measures to judge the fairness of an assignment.

## Background and Related Work

One and two sided matching and assignment problems have been studied in economics (Roth and Sotomayor 1992) and computer science (Manlove 2013; Brandt et al. 2016) for over 60 years. Matching and assignment have many applications including kidneys exchanges (Dickerson, Procaccia, and Sandholm 2014) and school choice. Our problem is often called the multi-agent resource allocation (MARA) problem in computer science (Bouveret, Chevaleyre, and Lang 2016) The papers to referees formulation of this problem has some additional side constraints common in the economics literature, but not as common in computer science (Manlove 2013). In the economics literature the *Workers-Firms* problem is the most closely related analogue to our problem, modeling many-many matchings with capacities (Klaus, Manlove, and Rossi 2016).

The conference paper assignment has been studied a number of times over the years in computer science (Goldsmith and Sloan 2007), as has defining and refining notions of fairness for the assignment vectors in multi-agent allocation problems (Golden and Perny 2010). We build off the work of Garg et al. (2010), who extensively study the notion of fair paper assignments, including lexi-min and rank-maximal assignments, within the context of conference paper assignment. Garg et al. (2010) show that for the setting we study, finding an egalitarian optimal assignment and finding a leximin optimal assignment are both NP-hard when there are three or more equivalence classes; and polynomial time computable when there are only two. They also provide an approximation algorithm for leximin optimal assignments. We know that if the capacity constraints are hard values, i.e., each reviewer must review $\leq x$ papers and each paper must receive exactly $y$ reviews, then the resulting version of capacitated assignment is NP-hard (Long et al. 2013). Answer set programming for CPAP was studied by Amendola et al. (2016); they encode the CPAP problem in ASP and show that finding a solution that roughly correspond to the leximin optimal and egalitarian solutions can be done in reasonable time for large settings ($\approx 100$ agents).

CPAP also receives considerable attention in the recommender systems (Conry, Koren, and Ramakrishnan 2009) and machine learning (Charlin, Zemel, and Boutilier 2012) communities. Often though, this work takes the approach of attempting to infer a more refined utility or preference model in order to distinguish papers. Fairness and efficiency concerns are secondary. A prime example of this is the Toronto Paper Matching System designed by Charlin and Zemel (2013). This system attempts to increase the accuracy of the matching algorithms by having the papers express preferences over the reviewers themselves; where these preferences are inferred from the contents of the papers.

We make use of Order weighted averages (OWAs), often employed in multi-criteria decision making (Yager 1988). OWAs have recently received attention in computational social choice for voting and ranking (Goldsmith et al. 2014), finding a collective set of objects for a group (Skowron, Faliszewski, and Lang 2016), and multi-winner voting with proportional representation (Elkind et al. 2014; Elkind and Ismaili 2015). The key difference between CPAP and voting using OWAs in the ComSoc literature is that CPAP does not select a set of winners that all agents will share. Instead, all agents are allocated a possibly disjoint set of objects.

## Σ-OWA Assignments

An OWA is a function defined as a $K$ length vector of non-negative numbers, as a vector $\alpha^{(K)} = (\alpha_1, \ldots, \alpha_K)$. Let $\mathbf{x} = (x_1, \ldots, x_K)$ be a vector of $K$ numbers and let $\mathbf{x}^{\downarrow}$ be the non-increasing rearrangement of $\mathbf{x}$, i.e. $\mathbf{x}^{\downarrow} = \mathbf{x}_1^{\downarrow} \geq \mathbf{x}_2^{\downarrow} \geq \ldots \geq \mathbf{x}_K^{\downarrow}$. We say: $OWA_\alpha(\mathbf{x}) = \alpha \cdot \mathbf{x}^{\downarrow} = \sum_{i=1}^{K} \alpha_i \cdot \mathbf{x}_i^{\downarrow}$.

In order to apply OWAs to our setting we need to define the *weighted rank signature* of an assignment. Let $\omega_i(A)$ be defined as the sorted vector of utility that a referee gets from an assignment $A$. Formally, $\omega_i(A) = sort(\{\forall j \in A(i,:) : \mathbf{u}_i(r(j))\})$. For example, if $A(i,:)$ included two objects with utility 3, one of utility 1, and one of utility 0, we would have $\omega_i(A) = (3,3,1,0)$.

Our inspiration for applying OWAs comes from a multi-winner voting rule known as Proportional Approval Voting (PAV) (Kilgour 2010; Aziz et al. 2015b; 2015a) and generalizations studied in social choice (Faliszewski et al. 2017). In approval voting settings, each agent can approve of as many candidates as they wish. Under the standard approval voting (AV) method, all approvals from each agent have the same weight, though this is not always desirable (Aziz et al. 2015b). It intuitively does not seem fair; once a candidate that you like has been selected your next candidate in the winning set should seemingly count less. Hence in PAV, which is designed to be more fair (Aziz et al. 2015a), a voter's first approval counts for a full point, the second for $1/2$, the next for $1/3$, etc.

We want to find a way to distribute objects to agents in a "more fair" way inspired by PAV. If we desire to directly get a rank maximal assignment, completely ignoring the utilities, then we know this is polynomial by a result from Garg et al. (2010). However we can use OWAs if we wish to modulate between using the utilities and using only the ranks. We use the sum over all agents of $OWA_\alpha(\omega) = \alpha \cdot \omega$ as the optimization criteria for the assignment.

We place some restrictions on our OWA vectors. Firstly, the length of $\alpha$ needs to be at least as long as the maximum agent capacity, $|\alpha| \geq \arg\max_{i \in N}(c_{max}^N(i))$, this is to ensure that $\alpha \cdot \omega$ is well defined for every agent (we truncate $\alpha$ if necssary). Typically the literature on OWAs assumes that $\alpha$ is normalized, i.e., $\sum_{1 \leq i \leq K} \alpha_i = 1$. We do not enforce

this convention as we wish to study the PAV setting with $\alpha = (1, 1/2, \ldots)$. This is formally a relaxation and we observe that whether or not the OWAs are normalized does not affect our computational results. However, we do require that our OWA vector be non-increasing and that each entry be $\geq 0$, i.e., for any $i, j \in [|\alpha|]$, $i < j$ we have $\alpha_i \geq \alpha_j \geq 0$. We will discuss increasing OWA vectors in the experimental section.

Σ-OWA ASSIGNMENT

Input: Given an assignment setting $(N, O, \succ, \mathbf{u}, \Delta)$ with agent capacities $[c^N_{min}(i), c^N_{max}(i)]$ for all $i \in N$, and object capacities $[c^O_{min}(j), c^O_{max}(j)]$ for all $j \in O$, and a non-increasing OWA vector $\alpha_i$ with $|\alpha| \geq \max_{\forall_i \in N}(c^N_{max}(i))$.

Question: Find a feasible assignment $A$ such that

$$A = \arg\max_{A \in \mathbb{A}} \sum_{i=1}^{|n|} \alpha_i \cdot \omega_i(A).$$

In our formulation, the OWA operator is applied to the vector of agent utilities and then we aggregate (or sum) these modified utilities to give the assignment objective. Hence, the Σ-OWA name. We observe that this formulation strictly generalizes the utilitarian assignment objective; if we set $\alpha = (1)^n$ we recover the utilitarian assignment.

One may also wish to consider applying the OWA over the sorted vector of total agent utility for their allocation, which one could call the OWA-Σ version of our problem. Indeed, this formulation of the problem has been considered before and proposed in the earliest writings on OWAs for decision making (Yager 1988). Taking the OWA-Σ formulation allows one to recover both the utilitarian assignment, $\alpha = (1/n, \ldots, 1/n)$, as well as the egalitarian assignment, $\alpha = (0, \ldots, 0_{n-1}, 1)$. However, because the OWA-Σ formulation is a generalization of the egalitarian assignment, it becomes NP-hard in general (Demko and Hill 1988).

We think of the $\alpha$ vector as a kind of control knob given to the implementer of the market, allowing them to apply a sub-linear transform to the agent utilities. This ability may be especially useful when agents are free to report their (normalized) utilities for ranks via bidding or other mechanisms (Budish and Cantillon 2012). In many settings the utility vector is controlled by the individual *agents*, while the OWA vector is under the control of the *market implementers*. Consider the following example.

**Example 1** *Consider a setting with four agents $N = \{a_1, a_2, a_3, a_4\}$ agents and four objects $O = \{o_1, o_2, o_3, o_4\}$. For all agents let $c^N_{min} = c^N_{max} = 2$ and for all objects let $c^O_{min} = c^O_{max} = 2$. For the Σ-OWA assignment, let $\alpha = (1, 1/2)$.*

| | $o_1$ | $o_2$ | $o_3$ | $o_4$ |
|---|---|---|---|---|
| $a_1$ | 11 | 9 | 0 | 0 |
| $a_2$ | 8 | 8 | 2 | 2 |
| $a_3$ | 7 | 7 | 3 | 3 |
| $a_4$ | 6 | 6 | 4 | 4 |

*We get the following allocations.*

*Utilitarian:* $A(a_1,:) = \{o_1, o_2\}, u_1(A) = 20$; $A(a_2,:) = \{o_1, o_2\}, u_2(A) = 16$; $A(a_3,:) = \{o_3, o_4\}, u_3(A) = 6$; $A(a_4,:) = \{o_3, o_4\}, u_4(A) = 8$;
$\sum_i u_i(A) = 50$.
*OWA with* $\alpha = (1, 1/2)$: $A(a_1,:) = \{o_1, o_2\}, u_1(A) = 20$, $\alpha \cdot \omega_1 = 15.5$; $A(a_2,:) = \{o_2, o_3\}, u_2(A) = 10$, $\alpha \cdot \omega_2 = 9.0$; $A(a_3,:) = \{o_1, o_4\}, u_3(A) = 10$, $\alpha \cdot \omega_3 = 8.5$; $A(a_4,:) = \{o_3, o_4\}, u_4(A) = 8$, $\alpha \cdot \omega_4 = 5.0$;
$\sum_i u_i(A) = 48$.
*Egalitarian:* $A(a_1,:) = \{o_1, o_4\}, u_1(A) = 11$; $A(a_2,:) = \{o_2, o_4\}, u_2(A) = 10$; $A(a_3,:) = \{o_2, o_3\}, u_3(A) = 10$; $A(a_4,:) = \{o_1, o_3\}, u_4(A) = 10$; $\sum_i u_i(A) = 41$.

Inspecting the results of Example 1, we observe that in the set of all utilitarian maximal assignments have $a_1$ and $a_2$ each being assigned to $o_1$ and $o_2$, in the set of all Σ-OWA maximal assignments $a_3$ is assigned one of $o_1$ or $o_2$ while $a_2$ is assigned one of $o_3$ or $o_4$, while in the set of all egalitarian maximal assignments each of the agents receives one of either $o_1$ or $o_2$ along with one of $o_3$ or $o_4$. Thus we observe the following.

**Observation 2** *The set of assignments returned by each of the three objective functions, utilitarian, egalitarian, and OWA, can be disjoint.*

There are instances where the set of Σ-OWA assignments is the same as the set of egalitarian assignments, but disjoint from the set of utilitarian assignments. An interesting direction for future work to fully characterize Σ-OWA assignments and discover OWA vectors with nice properties.

## Pareto Optimality

An allocation $S$ is *more preferred by a given agent with respect to pairwise comparisons* than allocation $T$ if $S$ is a result of replacing an object in $T$ with a strictly more preferred object (Aziz et al. 2014). Note that we use the word *allocation* to refer to the allocation to a single agent and *assignment* to all agents. Also note that the pairwise comparison relation is transitive. An assignment is *Pareto optimal with respect to pairwise comparisons* if there exists no other assignment that each agent weakly prefers and at least one agent strictly prefers.

**Lemma 3** *Consider an agent i and two allocations S and T of equal size. Then if S is weakly preferred to T by i with respect to pairwise comparison, then S yields at least as much OWA value as T for* any *OWA vector no matter if it is increasing or decreasing.*

**Proof:** Note that $S$ can be viewed as a transformation from $T$ where each object $j$ is replaced by some other object $j'$ that is weakly preferred. If the utility of $j' = j$, then $j'$ is multiplied by the same entry of the OWA vector. If the utility of $j' > j$ then $j'$ may rise in the weighted rank signature, being multiplied by a different entry in the OWA vector. However, since we sort this signature the values may only stay the same or increase. Since the OWA transform is bilinear, the total OWA score of $S$ is at least as much as that of $T$. ⋄

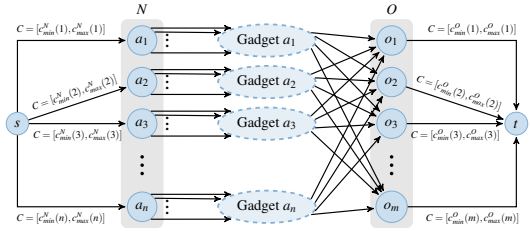



Figure 1: Main gadget for the reduction which enforces the agent and object capacity constraints.

**Proposition 4** *The Σ-OWA maximal assignment is Pareto optimal with respect to pairwise comparison irrespective of the OWA.*

**Proof:** Assume for contradiction that a Σ-OWA maximal assignment $A$ is not Pareto optimal with respect to pairwise comparisons. From Lemma 3, there exists another assignment $A'$ that each agent weakly prefers and at least one agent strictly prefers. But this means that in $A'$ each agent gets at least as much OWA score and at least one agent gets strictly more. This contradicts the fact that $A$ is OWA maximal. Lemma 3 applies here as we can enforce the assignments to be the same size using the capacity constraints. ⋄

## An Algorithm for Σ-OWA assignments

We give an algorithm for finding Σ-OWA assignments using flow networks. In this proof we use the most general formulation of our problem by allowing the values of the upper and lower per-agent capacities, $[c^N_{min}(i), c^N_{max}(i)]$, to vary for each agent; and the upper and lower object capacities, $[c^O_{min}(j), c^O_{max}(j)]$, to vary for each object.

**Theorem 5** *An Σ-OWA assignment can be found in polynomial time.*

**Proof:** We reduce our problem to the problem of finding a minimum cost feasible flow in a graph with upper and lower capacities on the edges, which is a polynomial time solvable problem. In addition to being polynomial time solvable, we know that the flow is integral as long as all edge capacities are integral, even if we have real valued costs (Ahuja, Magnanti, and B.Orlin 1993). Figures 1 and 2 provide a high level view of the flow network that we will construct.

In Figure 1 we first build a tripartite graph with two sets of nodes and one set of gadgets per agent: the agent nodes, one for each agent $a_i$; the agent gadgets, one (illustrated in Figure 2) for each agent $a_i$; and the object nodes, one for each object $o_j$. There is an edge from the source node $s$ to each of the agent nodes, each with cost 0, minimum flow capacity $c^N_{min}(i)$ and a maximum flow capacity $c^N_{max}(i)$. This set of edges and nodes enforces the constraint that each $a_i$ has capacity $[c^N_{min}(i), c^N_{max}(i)]$. We also construct an edge from each object node to the sink $t$. Each of these edges has a cost 0, a minimum capacity $c^O_{min}(j)$, and a maximum capacity $c^O_{max}(j)$. This set of edges enforces the constraint that each $o_j$ has capacity $[c^O_{min}(j), c^O_{max}(j)]$.

We now turn to the agent gadget depicted in Figure 2 for arbitrary $a_i$. The leftmost node and the rightmost set of nodes

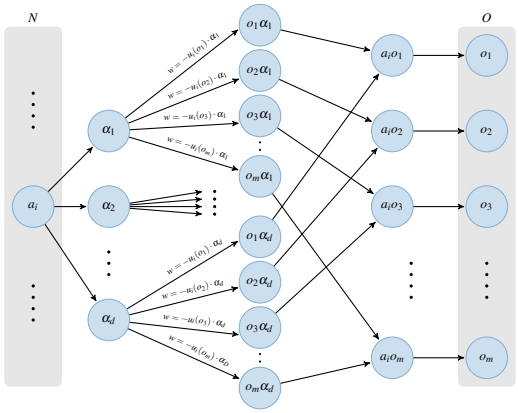



Figure 2: The per agent gadget. Note that all costs on edges are 0 and all capacities are [0,1] unless otherwise noted.

in Figure 2 correspond to the agent nodes $N$ and object nodes $O$ in Figure 1, respectively. In each agent gadget we create a tripartite sub-graph with the agent node $a_i$ serving as the source and the set of object nodes $O$ serving as the sinks.

We create three layers of nodes which we describe in turn from left to right. First, we create a set of decision nodes with labels $\alpha_1, \ldots, \alpha_d$ where $c^N_{max}(i) \leq d \leq |\alpha| = c^N_{max}(a'_i)$ for some $a'_i$ with maximal capacity. We use $d$ here to generally handle the case when some agent $a'_i$ has capacity $c^N_{max}(a_i) << c^N_{max}(a'_i)$ and hence we need to deal with a longer $\alpha$ vector than the allowable capacity constraint allows for agent $a_i$. Intuitively, we will be multiplying the OWA value $\alpha_1$ by the utility for some object, so we need to keep track of all the values that could result. The arcs from $a_i$ to each of the nodes in this set has upper capacity 1, minimum capacity 0, and cost 0. If we have the case that $c^N_{max}(i) < d$ then we set the maximum capacity of the edges to node(s) $\alpha_j, j > c^N_{max}(i)$ to 0. This enforces that each value in the OWA vector can modify at most one utility value.

For each of the decision nodes $\alpha_1, \ldots, \alpha_d$ constructed, we create a set of object/decision nodes for each $o_j$ which we denote $o_j\alpha_k$. From each of the decision nodes $\alpha_1, \ldots, \alpha_d$ we create an edge to each of the object/decision nodes created for this particular decision node $\alpha_k$, i.e., $o_1\alpha_1, o_2\alpha_1, \ldots, o_m\alpha_1$ for $\alpha_1$. Each of these edges has maximum capacity 1 and a cost equal to $-1 \cdot u_i(o_j) \cdot \alpha_1$ for rank 1 and object $o_j \in O$. These costs are the (negative) cost that matching agent $a_i$ with object $o_j$ at weighted rank $d_k$ contributes to the OWA objective.

Finally, we create one set of agent/object nodes, one for each $o_j$ denoted $a_io_j$. From all the object/decision nodes we connect all nodes with a label of $o_j$ to the corresponding agent/paper node, i.e., $o_1\alpha_1, o_1\alpha_2, \ldots o_1\alpha_d$ all connect to $a_io_1$ with cost 0 and maximum capacity 1. We then connect the agent/object node to the corresponding object node in the main construction from $O$, i.e., $a_io_1$ to $o_1$ with cost 0 and maximum capacity 1. This set of nodes and edges enforces that each agent can be assigned each object once.

We can extract an assignment from the minimum cost feasible flow by observing that paper $o_j$ is allocated to agent $a_i$

if and only if there is a unit of flow passing from the particular agent/object node $a_i o_j$ to the object node $o_j$. We now argue for the correctness of our algorithm in two steps, (1) that all constraints for the Σ-OWA assignment problem are enforced and (2) that a minimum cost feasible flow in the constructed graph gives an Σ-OWA assignment. For (1) we note that since the units of flow across the graph represent the assignment and we have explained how the capacity constraints on all edges enforce each of the particular constraints imposed by our definition of a feasible assignment, there is a feasible flow iff the flow satisfies the constraints.

For (2) observe that for each agent, the $\alpha$ nodes fill with flow in order from $\alpha_1$ to $\alpha_d$ as the OWA vector is non-increasing and the utilities are decreasing, i.e., for each agent, the edge costs monotonically increase from the edges associated with $\alpha_1$ to the edges associated with $\alpha_d$. Any unit of flow from $s$ to $t$ must take the least cost (most negative) path across the network that has remaining capacity. This path will move through some agent gadget where it will cross an edge associated with the smallest $\alpha$ value not yet used. If this were not the case then we would end up selecting a path which assigned an object with higher utility to a smaller $\alpha$ value than necessary, meaning the flow would no longer be minimum cost. From the capacity constraints we know there is only one unit of flow that enters each decision node $\alpha_i$ and there is only one unit of flow that can leave each agent/paper node $a_i o_j$. Hence, each $\alpha_i$ can modify only one $o_j$ and each $o_j$ selected must be unique for this agent.

As the decision nodes are filled in order and $\alpha_i$ can only modify the value for a single object, we know the total cost of the flow across the agent gadget for each $a_i$ is equal to $-1 \cdot \alpha_i \cdot \omega_i$. Hence, the price of the min cost flow across all agents is equal to $-1 \cdot \sum_{\forall i \in N} \alpha_i \cdot \omega_i(A)$. Thus, the min cost flow in the graph is an Σ-OWA assignment. ⋄

## Generalizations

We observe two possible generalizations of Theorem 5 that allows us to use the construction for more general instances. First, the construction can be generalized to allow for $\alpha$ to vary for each agent. Observe that the decision nodes for each agent $a_i$ are independent from all other agents. Thus, for each agent (or a class of agents) we could use an OWA vector $\alpha^{a_i}$. This ability may be useful, e.g., when a group of agents reports the same extreme utility distribution and the organizer wishes to apply the same transform.

The second generalization that we can make to the above construction is to allow each agent to be assigned to each object more than once. While this ability does not make sense in the reviewers/papers setting (unless there are sub reviewers) there could be other capacitated assignment settings where we may wish to assign the agents to objects multiple times e.g., if there are discrete jobs that need to be done a certain number of times and a single agent can be assigned the same job multiple times.

To generalize the capacity constraint from 1 for each agent $i$ for each object $j$ we introduce a capacity upper bound $z_{i,j}$ which encodes the number of times that agent $i$ can be assigned to object $j$. Taking $z_{i,j} = 1$ for all $i$ and $j$ gives us the original CPAP setting. In order to enforce this constraint, within each agent gadget (Figure 2) we add a capacity constraint equal to $z_{i,j}$ from each edge $a_i o_j$ to $o_j$. If we want a lower bound for the number of copies of $o_j$ assigned to $a_i$ we can encode this lower bound on this edge as well.

We can extract an assignment from the minimum cost feasible flow by observing that paper $o_j$ is allocated to agent $a_i$ $z_{ij}$ times if and only if there are units of flow passing from the particular agent/object node $a_i o_j$ to the object node $o_j$. The argument for correctness follows exactly from the proof of Theorem 5 above.

**Corollary 6** *An Σ-OWA assignment can be found in polynomial time even if each agent $a_i$ has a unique OWA vector $\alpha^{a_i}$ and each object $o_j$ can be assigned to each agent $a_i$ any number of times (not just once).*

## Experiments

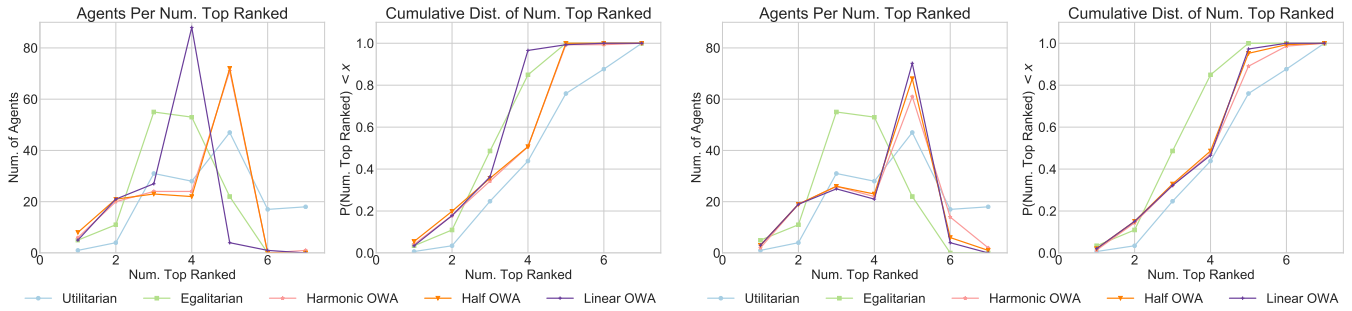
We evaluate the quality of Σ-OWA using real-world data from WWW.PREFLIB.ORG (Mattei and Walsh 2013). We implemented the algorithm described in Theorem 5 using networkX for Python and Lemon for C++. While the network flow algorithm gives a nice theoretical proof of polynomial solvability, it does not perform very well on practical instances. Instead, we modeled the problem as a MIP in Gurobi 7.0 and it ran in under 1 minute for all instances and settings using 4 cores. Our MIP is similar to the one given by Skowron, Faliszewski, and Lang (2016). However, as we have capacity constraints and individual/variable length OWAs, our MIP is more general.

We introduce a binary variable $x_{a,o}$ indicating that agent $a$ is assigned object $o$ as well as binary variable $r_{a,o,p}$ for the OWA matrix which notes that agent $a$ is assigned object $o$ at OWA rank $p$. Given this we maximize Σ-OWA utility using $\alpha^a$ as an OWA vector for each agent.

| | | |
|---|---|---|
| max | $\sum_{a \in A} \sum_{o \in O, p \in P} u_a(o) \cdot \alpha_p^a \cdot r_{a,o,p}$, s.t. | |
| (1) | $c^O_{min}(o) \leq \sum_{a \in A} x_{a,o} \leq c^O_{max}(o)$ | $\forall o \in O$ |
| (2) | $c^N_{min}(a) \leq \sum_{o \in O} x_{a,o} \leq c^N_{max}(a)$ | $\forall a \in A$ |
| (3) | $\sum_{p \in P} r_{a,o,p} \leq 1$ | $\forall a \in A, \forall o \in O$ |
| (4) | $\sum_{o \in O} r_{a,o,p} \leq 1$ | $\forall a \in A, \forall p \in P$ |
| (5) | $\sum_{p \in P} r_{a,o,p} \geq x_{a,o}$ | $\forall a \in A, \forall o \in O$ |
| (6) | $\sum_{o \in O} r_{a,o,p} \geq \sum_{o \in O} r_{a,o,p+1}$ | $\forall a \in A, \forall p \in P$ |
| (7) | $\sum_{o \in O} r_{a,o,p} \cdot u_a(o) \geq \sum_{o \in O} r_{a,o,p+1} \cdot u_a(o)$ | $\forall a \in A, \forall p \in P$ |

Constraints (1)–(4) enforce the cardinality constraints on the agents, objects, and OWA rank matrix. Constraint (5) links the agent and object assignments to be positions in the OWA rank matrix. Line (6) enforces that the rank matrix fills from the first position to the $c^N_{max}$ position. And finally (7) enforces that the Σ-OWA value of the assignment positions in the rank matrix must be decreasing. We then maximize the sum over all agents of the OWA objective value.

We focus our discussion on MD-00002-00000003 from PREFLIB which is the largest dataset available.This dataset contains 146 agents and 175 objects. We evaluate the utilitarian, egalitarian, and Σ-OWA assignments when each object must receive 3–4 reviews and each agent must review 6–7 objects. In the data, each agent sorts the papers into 4 equivalence classes which we gave utility values $(5, 3, 1, 0)$. We study three different OWA vectors and their reversals




(a) Decreasing OWA Vectors

(b) Increasing OWA Vectors

Figure 3: Count of agents receiving $x$ top ranked papers and the cumulative distribution function (CDF) for the number of agents being assigned $x$ top ranked objects for decreasing OWAs (left) and increasing OWAs (right). Though 100% of the agents receive between 1 and 5 top ranked objects for the egalitarian and Σ-OWA assignments (CDF), the most agents receive the most top ranked objects under the decreasing Linear OWA assignment. More agents receive fewer top ranked objects than under the increasing vectors.

| | Mean | Hoover | Gini |
|---|---|---|---|
| Utilitarian | 21.71 (4.59) | 0.0916 | 0.1184 |
| Egalitarian | 19.12 (1.13) | 0.0234 | 0.0315 |
| Inc. Harmonic OWA | 18.99 (3.43) | 0.0680 | 0.0961 |
| Inc. Geometric OWA | 18.52 (3.00) | 0.0626 | 0.0841 |
| Inc. Linear OWA | 18.38 (2.75) | 0.0617 | 0.0793 |
| Dec. Harmonic OWA | 17.92 (2.90) | 0.0643 | 0.0831 |
| Dec. Geometric OWA | 17.74 (2.77) | 0.0667 | 0.0819 |
| Dec. Linear OWA | 16.25 (2.51) | 0.0576 | 0.0797 |

Table 1: Mean utility (and standard deviation) per agent with the Hoover and Gini indices over all assignment objectives for the real-world data. The decreasing OWA vectors are strictly better in terms of maximizing fairness under both indices with the decreasing Linear OWA being the most fair besides the egalitarian objective.

such that they are increasing or decreasing: (1) the Harmonic OWA vector $(1, 1/2, 1/3, \ldots)$ which has provably nice fairness properties in the multi-winner setting (Faliszewski et al. 2017), the Geometric OWA vector $(1, 1/2, 1/4, 1/8 \ldots)$ which is a steeper redistribution function, and a Linear OWA where for a vector of length $n$ we have a linear spaced vector between 1 and 0, e.g., $n = 4$ gives $(1, 2/3, 1/3, 0)$.

Figure 3 shows the counts and the cumulative distribution function (CDF) for the number of top ranked objects the agents receive. On the left side, we see that $\approx 70$ agents receive 5 top ranked papers under the decreasing Harmonic and Geometric OWA assignment but $\approx 85$ agents receive 4 top ranked objects under the Linear OWA assignment. For the increasing OWA papers the performance is the same across all OWA vectors checked. Under the utilitarian assignment, several agents receive an entire set of top ranked objects, while the egalitarian assignment modulates this so that most agents only receive 3–4 top ranked objects. In contrast, all Σ-OWA assignments are balanced between these with the most agents receiving 5 top ranked objects.

Table 1 shows the mean and standard deviation for the agent utilities and fairness scores. Looking first at the egalitarian objective we see that the mean agent utility is indeed quite high and from investigating the standard deviation, quite tightly packed together. This is also seen when looking at both the Hoover and the Gini index, which are both significantly lower for the egalitarian objective than they are for the utilitarian objective. Using the fairness criteria as a measure it is easy to see that the decreasing OWA vectors are all strictly more fair than the increasing versions. Either increasing or decreasing, the Σ-OWA assignments are more fair than the utilitarian assignment. Hence, an important direction is to establish what OWA vectors are best.

## Conclusions

We propose and provide algorithms for the novel notion of Σ-OWA assignments that give a central organizer a "slider" to efficiently transition from utility maximizing to rank maximal assignments. Finding axiomatic characterizations for good OWA vectors and to fully understand the difference between increasing and decreasing OWA vectors are important next steps. It is generally the case that reviewers want to review fewer, not more, papers. It would be interesting to study CPAP from the point of view of *chores*, as they are called in the economics literature.


### Acknowledgements

We thank Haris Aziz, Serge Gaspers, and Joachim Gudmundsson for their helpful discussions. Data61 is supported by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program. Toby Walsh is supported by the European Research Council and by AOARD Grant FA2386-12-1-4056.


## References


Ahmed, F.; Dickerson, J. P.; and Fuge, M. 2017. Diverse weighted bipartite b-matching. In *Proc. of the 26th IJCAI*.

Ahuja, R. K.; Magnanti, T. L.; and B.Orlin, J. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall.

Amendola, G.; Dodaro, C.; Leone, N.; and Ricca, F. 2016. On the application of answer set programming to the confer-

ence paper assignment problem. In *Proc. of the 15th AI*IA Conference*. 164–178.

Aziz, H.; Gaspers, S.; Mackenzie, S.; and Walsh, T. 2014. Fair assignment of indivisible objects under ordinal preferences. In *Proc. of the 13th AAMAS Conference*, 1305–1312.

Aziz, H.; Brill, M.; Conitzer, V.; Elkind, E.; Freeman, R.; and Walsh, T. 2015a. Justified representation in approval-based committee voting. In *Proc. of the 29th AAAI Conference*.

Aziz, H.; Gaspers, S.; Gudmundsson, J.; Mackenzie, S.; Mattei, N.; and Walsh, T. 2015b. Computational aspects of multi-winner approval voting. In *Proc. of the 14th AAMAS Conference*.

Aziz, H.; Lev, O.; Mattei, N.; Rosenschein, J. S.; and Walsh, T. 2016. Strategyproof peer selection: Mechanisms, analyses, and experiments. In *Proc. of the 30th AAAI Conference*.

Bouveret, S., and Lemaître, M. 2016. Characterizing conflicts in fair division of indivisible goods using a scale of criteria. *Autonomous Agents and Multi-Agent Systems* 30(2):259–290.

Bouveret, S.; Chevaleyre, Y.; and Lang, J. 2016. Fair allocation of indivisible goods. In Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A. D., eds., *Handbook of Computational Social Choice*. Cambridge University Press. chapter 12, 284–311.

Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A. D., eds. 2016. *Handbook of Computational Social Choice*. Cambridge University Press.

Budish, E., and Cantillon, E. 2012. The multi-unit assignment problem: Theory and evidence from course allocation at harvard. *The American Economic Review* 102(5):2237–2271.

Charlin, L., and Zemel, R. S. 2013. The Toronto paper matching system: an automated paper-reviewer assignment system. In *Proc. of the ICML Workshop on Peer Reviewing and Publishing Models (PEER)*.

Charlin, L.; Zemel, R. S.; and Boutilier, C. 2012. A framework for optimizing paper matching. *CoRR* abs/1202.3706.

Conry, D.; Koren, Y.; and Ramakrishnan, N. 2009. Recommender systems for the conference paper assignment problem. In *Proc. of the 3rd ACM RecSyS*, 357–360.

Demko, S., and Hill, T. P. 1988. Equitable distribution of indivisible objects. *Mathematical Social Sciences* 16:145–158.

Dickerson, J. P.; Procaccia, A. D.; and Sandholm, T. 2014. Price of fairness in kidney exchange. In *Proc. of the 13th AAMAS Conference*, 1013–1020.

Elkind, E., and Ismaili, A. 2015. Owa-based extensions of the Chamberlin-Courant rule. In *Proc. of the 4th ADT Conference*, 486–502.

Elkind, E.; P.; Faliszewski; Skowron, P.; and Slinko, A. 2014. Properties of multiwinner voting rules. In *Proc. of the 13th AAMAS Conference*, 53–60.

Faliszewski, P.; Skowron, P.; Slinko, A.; and Talmon, N. 2017. Multiwinner rules on paths from k-borda to chamberlin–courant. In *Proc. of the 26th IJCAI*.

Fishburn, P. C. 1974. Lexicographic orders, utilities and decision rules: A survey. *Management science* 20(11):1442–1471.

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

Golden, B., and Perny, P. 2010. Infinite order lorenz dominance for fair multiagent optimization. In *Proc. of the 9th AAMAS Conference*, 383–390.

Goldsmith, J., and Sloan, R. 2007. The AI onference paper assignment problem. In *Proc. of the 22nd AAAI Conference-Workshop on Preference Handling for Artificial Intelligence (MPREF)*.

Goldsmith, J.; Lang, J.; Mattei, N.; and Perny, P. 2014. Voting with rank dependent scoring rules. In *Proc. of the 28th AAAI Conference*, 698–704.

Kilgour, D. M. 2010. Approval balloting for multi-winner elections. In *Handbook on Approval Voting*. Springer.

Klaus, B.; Manlove, D. F.; and Rossi, F. 2016. Matching under preferences. In Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A. D., eds., *Handbook of Computational Social Choice*. Cambridge University Press. chapter 14, 333–356.

Long, C.; Wong, R.; Peng, Y.; and Ye, L. 2013. On good and fair paper-reviewer assignment. In *Proc. of the 13th IEEE ICDM*, 1145–1150.

Manlove, D. 2013. *Algorithmics of Matching Under Preferences*. World Scientific.

Mattei, N., and Walsh, T. 2013. Preflib: A library for preferences, http://www.preflib.org. In *Proc. of the 3rd ADT Conference*, 259–270.

Merrifield, M. R., and Saari, D. G. 2009. Telescope time without tears: a distributed approach to peer review. *Astronomy & Geophysics* 50(4):4–16.

Price, S., and Flach, P. A. 2016. Computational support for academic peer review: A perspective from artificial intelligence. *CACM* 60(3):70–79.

Rawls, J. 1971. *A Theory of Justice*. Harvard University Press.

Roth, A. E., and Sotomayor, M. A. O. 1992. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Cambridge University Press.

Schneckenburger, S.; Dorn, B.; and Endriss, U. 2017. The atkinson inequality index in multiagent resource allocation. In *Proc. of the 16th AAMAS Conference*.

Skowron, P.; Faliszewski, P.; and Lang, J. 2016. Finding a collective set of items: From proportional multirepresentation to group recommendation. *AIJ* 241:191–216.

Yager, R. 1988. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man and Cybernetics* 18(1):183–190.