

How Fair are Fair Allocations?

Lone Kiilerich^{*1} and Sanne Wøhlk^{†1}

¹Cluster for Operations Research And Logistics, Department of Economics and Business Economics, Aarhus University, Denmark

December 14, 2016

Abstract

In this paper, we study a situation where a number of objects are to be allocated to a number of agents, where each agent can receive multiple objects and the number of agents that can receive a copy of a given object is limited by a capacity constraint, and utility functions are linear. The paper investigates the fairness of such allocations. Previous literature has applied numerous ways of measuring allocation fairness as well as numerous strategies for obtaining allocations. The goal of this paper is to study the fairness of allocations obtained by different allocation methods as viewed by different fairness measures. We hope that this will help guide future research towards a rational choice of allocation methods and measures.

1 Introduction

Imagine the following situation: A number of people want to attend some activities. But the activities have limited capacity and therefore some will not have their wishes fulfilled. A decision maker creates an allocation of the people to the activities, i.e. decides who gets to attend which activities. The question is now, how can this be done in a fair manner? Or rather, looking at the final allocation, how fair is it? This immediately leads us to the main questions to be investigated in this paper: 'How do we measure fairness?' and 'How do we obtain a fair allocation?'

Studying fairness is not new. In fact, several researchers have suggested different measures of fairness for different kinds of problems. Others have suggested properties that fair allocations must satisfy, and yet others have studied the fairness of the allocation process itself rather than the fairness of the final solution. However, in this study, we ask the question 'What is fair?'

We seek to answer these questions in the following way: In Phase 1, we use different fairness measures and properties from the academic literature to create allocations for a number of small data sets. Next, we ask a group of respondents to evaluate the fairness of these allocations and to provide information on what they perceive as fair in allocations in general. In Phase 2, we turn to an operations research approach and create larger data sets for which we create allocations based on a number of methods. These allocations are then cross-evaluated by all measures. Together with the knowledge obtained in Phase 1, this leads to a recommendation as regards choice of allocation method.

^{*}lone.ki.ch@econ.au.dk

[†]sanw@econ.au.dk

The contribution of this paper is threefold. Firstly, we provide an overview of different ways of measuring fairness and of obtaining fair allocations found in the academic literature. Secondly, we relate the fairness from the academic literature to the fairness perceived by humans using allocations constructed by various methods and thirdly, we cross-evaluate a large number of allocation methods and fairness measures in the search for a method that is found to be fair by a wide range of measures.

Doing this, we follow the advice of Nobel prize winner Daniel Kahneman, who stated: "The important task for students of economic fairness is not to identify ideal behavior but to find the line that separates acceptable conduct from actions that invite opprobrium and punishment", Kahneman (2012) p. 307. Rather than favoring a method that performs optimally for one single measure, we seek a method that performs well, but not necessarily optimally, based on many different measures.

The remainder of this paper is outlined as follows. In Section 2, we review the academic literature which is the basis of our study and in Section 3 we provide the terminology and exact problem definition. Section 4 provides an overview of the fairness measures found in the literature for the studied problem along with a number of new measures. Here, we also extend the ranking of some measures for indivisible objects without copies by Bouveret and Lemaître (2016) to include the case where multiple copies of each object exist. In Section 5, we present methods for obtaining fair allocations based on the fairness measures presented in Section 4. Finally, in Section 6, we present our analysis and in Section 7, we conclude our study.

2 Related Literature

Fair allocation has been studied in many different settings in the literature and we will list a few of those settings below. The problem of dividing resources fairly without any kind of transfers (e.g. money) arises in a variety of different contexts. At some universities, students apply to courses with limited capacity and the administrative unit subsequently makes the assignment decision (Budish and Cantillon (2012)). In some countries, children are assigned to schools using the parents' preferences and the priority the school assigns to that child (Abdulkadiroğlu and Sönmez (2003)). In Denmark, when students finish medical school, they have to go one year in rota. The rota places are assigned using serial dictatorship where the selection order is determined randomly (Fedders (2012)). Patients in need of an organ wait for an organ and when a new organ is available it is assigned to one of the patients (Hild (2001)). Yet another example is the allocation of resources such as medical components and food in the case of a disaster (Fiedrich et al (2000)).

All of the above examples should make it clear that fairly dividing a set of objects is an important real world problem. That is why we find it important to investigate whether the theoretical fairness concepts coincide with how fairness is regarded by humans. Next we will look into the work already made within the area of fairly dividing resources in different settings. This overview should give an impression of the theoretical methods and ideas we build upon in the following sections of this article.

Within parallel job scheduling, the time issue is a significant parameter in the definition of fairness and the classical performance metrics are response time, wait time, and slowdown (Ernemann et al (2004); Feitelson et al (2005)). The First-Come-First-Served policy can be considered fair due to the fact that all jobs are processed in order of arrival and are thereby prioritized based on seniority (Leung et al (2010)). A significantly different approach is to ensure that each job gets a fair share of the resources. This is done by Sabin and Sadayappan (2005) where the deserved time for a job is compared with the actual time given to a job. We refer the reader to Tóth (2014) for an overview of different ways of measuring fairness in this setting.

Problems where fairness is studied can generally be divided in two classes: Problems with divisible objects and problems with indivisible objects. Even though the problem studied in this paper concerns indivisible objects, much inspiration is drawn from problems with divisible objects which is a far more well-studied area.

The simplest problem with divisible objects is the so-called cake cutting problem where an inhomogeneous cake represents a single divisible good. A practical example of cake cutting is assignment of commercial time slots where different companies value the time slot before a given tv-program differently. Within this area, two fairness criteria seem to be accepted; envy-free and proportional fairness. Envy-free means that no agent prefers another agent's piece more than his own, and proportional fairness is defined as each agent receiving at least $1/m$ part of the cake (in value) where m is the number of agents. We refer the reader to Procaccia (2013) for an overview of mechanisms that make proportional fair allocations among which some are also envy-free. However, in order to ensure that the mechanism actually divides the cake fairly, it is important that the agents do not lie about their preferences. This is studied in Chen et al (2013). A variation of this problem is studied by Carvalho and Larson (2012) in a setting where each agent is asked to evaluate the other agents after teamwork, and a homogeneous reward is shared among them based on these evaluations. The paper presents a method where the agents benefit from providing truth evaluations.

Network resource allocation can be viewed as a division of several divisible objects. An example is a computer network where jobs cannot be processed without having both cpu and memory assigned. Thus the utility obtained is more complex than the additive utility seen in cake cutting. This problem has been studied from several different angles, including proportional fairness (Boche and Schubert (2009); Cole et al (2013); Kelly (1997)), Nash Bargaining (Boche and Schubert (2009); Han et al (2005)), max min fairness (Bonald et al (2006); Kelly (1997)) and Dominant Resource Fairness (DRM). The proportional fairness in this context has inspired us to an alternative fairness measure in the context of indivisible objects.

Turning the attention to problems with indivisible goods, two main aspects are relevant: The number of objects assigned to each agent (one or several) and the presence of copies of the objects. Practical examples of problems with indivisible goods include organ donations, allocation of students to courses, and allocation of children to schools. Two types of approaches are used to seek fair allocations, both of which are based on the agents defining the utility they obtain from receiving each object or bundle of objects. The first type is to formulate an overall function combining the agents' utilities in some useful manner and then optimize this function in order to obtain fair allocations. This includes maximizing the average utility obtained by the agents or maximizing the utility of the agent that gets the least utility. Such a max min approach is used by Bezáková and Dani (2005). The second type is to first define fairness criteria, i.e. criteria that an allocation must fulfill in order to be fair, and then find an allocation that satisfies these criteria if such an allocation exists. Examples of such criteria are max min fair share, proportional fair share, and envy-free (Bouveret and Lemaître (2016); Budish (2011); Procaccia and Wang (2014)). These approaches of obtaining fairness will also be used in our study and the above concepts will be elaborated upon in Section 4. Defining procedures for obtaining fair allocations with respect to these criteria is, however, not straightforward. In Othman et al (2010), the resulting allocation may be infeasible in the sense that objects may be over-allocated and in Procaccia and Wang (2014) the resulting allocation is only approximately fair. Finally, in Kesten and Yazıcı (2012), some objects are thrown away in order to avoid unfair solutions.

Because the objects are indivisible, it is difficult to achieve fair allocations and thus some studies focus on making the division procedure (ex ante) fair instead of making the actual allocation (ex-post) fair (Bezáková and Dani (2005); Nguyen and Vohra (2012)). From a game

theoretic point of view, if agents can benefit from revealing untrue preferences, they can manipulate the final outcome by giving false preferences and in that way prevent fairness. A mechanism that prevents this is referred to as being *strategy-proof* and is studied in Bezáková and Dani (2005); Budish (2011); Pápai (2001). The cost of ensuring truth telling can, however, be even higher than the cost of playing strategically in a game that does not ensure truth telling (Budish and Cantillon (2012)). A common assumption is that truthful preferences are given as input (Bezáková and Dani (2005); Procaccia and Wang (2014)). We use this assumption in our study. Another assumption that is frequently made is that the utilities are additive (Budish and Cantillon (2012); Procaccia and Wang (2014); Bouveret and Lemaître (2016); Bezáková and Dani (2005) the argument for making this very strict assumption could be that it is fairly easy for the agents involved in the problem to state their utilities in this way. As some of the results we find build upon work with additive utility functions and because we need respondents to be able to relate to the allocations, this assumption will also be made in this paper.

3 Problem Description

We consider a problem with indivisible objects where multiple copies of each object exist. Each agent can receive more than one object, but never multiple copies of the same object. The number of agents that can receive a copy of the same object is bounded by the object’s capacity. The agents obtain utility by receiving the objects and the utility obtained by receiving multiple objects is a linear function of the individual utilities. As discussed in Section 2, this problem has been studied in several papers. The problem is inspired by the assignment of students to courses.

To formally define the problem, let \mathcal{A} be the set of m agents and let \mathcal{O} be the set of n objects. Let $\mathbf{q} \in \mathbb{N}^n$ be a capacity vector such that each object $j \in \mathcal{O}$ has a capacity $0 < q_j \leq m$ bounding the number of agents who can be assigned to the object. We use $w_{ij} \geq 0$ to denote the weight that agent $i \in \mathcal{A}$ puts on object $j \in \mathcal{O}$ and say that objects with $w_{ij} > 0$ are requested by agent i . This weight describes the desire that agent i has for object j .

A feasible allocation of objects to agents is denoted $S = \{S_1, \dots, S_m\}$ where $S_i \subseteq \mathcal{O}$ represents the set of objects assigned to agent i . Each agent i can be assigned to each object j at most once. There are no restrictions on the assignment of agents such as e.g. a maximum or minimum number of objects to assign to each agent or conditions such as at most one object from a set of objects can be assigned to an agent. We denote by \mathcal{S} the set of all feasible allocations.

Each agent has a utility function that describes the utility that agent i receives by the allocation $S = \{S_1, \dots, S_m\}$. For any allocation S , we use $u_i(S_i)$ to denote the total utility obtained by agent i when receiving the bundle S_i . It is assumed that the agents’ utility functions are additive and determined by the weight that the agent puts on the objects in S_i . Therefore $u_i(S_i) = \sum_{j \in S_i} w_{ij}$ for all $i \in \mathcal{A}$, $S_i \subseteq \mathcal{O}$. Agents do not gain utility by being assigned to unrequested objects.

We use three different types of utilities in our experiments. In the first type, the agents express their preferences for the objects by distributing preference points in the form of actual weights among the objects to indicate their relative desire for the objects. The weights are normalized such that $\sum_{j \in \mathcal{O}} w_{ij} = t$ for all $i \in \mathcal{A}$ for some constant t . In our experiments, we use $t = 100$. The second type is 0/1-utilities in which the agents merely state if they want the object. Here, $w_{ij} = 1$ if agent i wants object j , and zero otherwise. These 0/1-utilities are not normalized. However, in the third type, the agents also express their preferences in the form of 0/1-utilities, but now weights are assigned evenly to the objects requested by each agent such that $\sum_{j \in \mathcal{O}} w_{ij} = t$ for all $i \in \mathcal{A}$. This is referred to as normalized 0/1-utilities.

4 Measuring Fairness

In this section, we present a number of ways to measure the fairness of an allocation. The measures presented are general and independent of the choice of utility function. Given an allocation $S = \{S_1, \dots, S_m\} \in \mathcal{S}$, we use $\lambda^\Delta(S)$ to denote the fairness of S determined by the measure Δ , where Δ is replaced by the name of the measure under consideration. When it is clear from the context, we will leave out the allocation and simply write λ^Δ . Firstly, in Section 4.1, we consider measures where an objective function are sought to be optimized, whereas Sections 4.2 and 4.3 consider measures that are based on fairness concepts which are either satisfied or not.

4.1 Utilitarian Fairness Measures

Utilitarian measures are based on ideas such as minimizing the average or maximizing the minimum of a joint objective function. Even though they are not directly focused on optimizing fairness, these general ideas have been used for several decades for a wide range of problems within Operations Research. We therefore survey such measures in this section.

When considering an allocation S , a natural approach is to maximize the total utility of that allocation. This is equivalent to maximizing the average utility and is reflected by our first measure, referred to as *AverageUtility*. We have

$$\lambda^{AverageUtility} = \frac{1}{m} \sum_{i \in \mathcal{A}} u_i(S_i)$$

An alternative view is to focus the attention on the agent who obtained the least utility. One might claim that the higher the utility for this agent, the fairer the allocation. We measure this by *MinimumUtility* which is determined as

$$\lambda^{MinimumUtility} = \min_{i \in \mathcal{A}} u_i(S_i)$$

Instead of looking at the utility obtained by the unlucky agent, one could simply count the number of objects received by the agent who got the lowest number of objects. I.e. we consider the amount of objects rather than the utility. The motivation is that from a human perspective, it can seem unfair if one agent receives very few objects while another agent receives many objects even though their utilities may almost be the same. The *MinimumNumber* measure reflects this and is determined as

$$\lambda^{MinimumNumber} = \min_{i \in \mathcal{A}} |S_i|$$

Within the same line of reasoning, people may find an allocation unfair if the number of objects received differs highly among the agents even if the utilities are similar. This is the motivation behind *SpanNumber* which is calculated as

$$\lambda^{SpanNumber} = \max_{i \in \mathcal{A}} |S_i| - \min_{i \in \mathcal{A}} |S_i|$$

Furthermore, the measure *UnassignedObjects* considers the number of objects that are left unassigned even though at least one agent requests the objects. The motivation is that some fairness criteria found in the literature, in particular those presented in Section 4.3, obtain fairness by leaving some objects unallocated. From a human perspective, this does not appear

to be particularly fair. Especially in cases where several copies are available, as for example when assigning students to courses. We have

$$\lambda^{Unassigned} = \sum_{j \in \mathcal{O}} \left(q_j - \sum_{i \in \mathcal{A}} |S_i \cap \{j\}| \right)$$

Finally, when evaluating the fairness of an allocation, some people will consider not what the agents got, but rather what they did not get. We present two measures with this focus. We define the lost utility of an agent to be the utility of the objects he requested but did not receive and define *MaxLostUtility* as

$$\lambda^{MaxLostUtility} = \max_{i \in \mathcal{A}} (u_i(\mathcal{O}) - u_i(S_i))$$

When using preference points or normalized 0/1-utilities, we have

$$\lambda^{MaxLostUtility} = t - \lambda^{MinUtility}$$

However, this is generally not true for 0/1-utilities without normalization.

Alternatively, we can consider the number of objects not received by the agent who loses most objects. This is done in *MaxLostNumber* and is defined as

$$\lambda^{MaxLostNumber} = \max_{i \in \mathcal{A}} (|\{j \in \mathcal{O} | w_{ij} > 0\}| - |S_i|)$$

For 0/1-utilities, $\lambda^{MaxLostUtility} = \lambda^{MaxLostNumber}$. This is, however, not the case for preference points and normalized 0/1-utilities.

4.2 Fair Share Measures

The concept of fair share is that each agent shall receive the portion of all objects or all utility which fairly belongs to him. The reasoning behind this is that in this way, the agents are treated the same. However, when the objects are indivisible, and in particular when copies are present, this is not straightforward.

One fair share criterion found in the literature is the proportional fair share property which is found in relation to cake cutting (Procaccia (2013)), but also suggested for division of indivisible objects without copies (Bouveret and Lemaître (2016)). Here, the proportional fair share value for agent i is calculated as $u_i^{prop} = u_i(\mathcal{O})/m$ and the allocation $S = \{S_1, \dots, S_m\} \in \mathcal{S}$ is said to be proportional share fair if $u_i^{prop} \leq u_i(S_i)$ for all agents $i \in \mathcal{A}$.

The sharing incentive is a generalization of the proportional fair share and is used within division of several homogeneous divisible objects in Parkes et al (2012). It states that each agent should get at least as much utility as he would if he received a fraction of $\frac{1}{m}$ of each object in order to be willing to accept allocations other than the equal one. Formally, we have $u_i^{SI} = u_i(\langle 1/m, \dots, 1/m \rangle)$ and the allocation is fair according to the sharing incentive if $u_i^{SI} \leq u_i(S_i)$ for all agents $i \in \mathcal{A}$.

The presence of copies complicates matters and causes u_i^{prop} to be underestimated and the sharing incentive loses its meaning when goods are not divisible. This means that both measures are of little direct use in our case of indivisible goods with copies. We therefore present two proportional fair share variations for indivisible goods with copies. To the best of our knowledge, this generalization has not been made elsewhere in the literature.

Since there are q_j copies of j and m agents to share them, each agent should rightfully have a fraction of $\frac{q_j}{m}$ of j . Generalizing this, we define

$$u_i^{prop_{sum}} = \sum_{j \in \mathcal{O}} \frac{q_j}{m} u_i(\{j\})$$

and say that an allocation, S , is proportional sum share fair if $u_i^{prop_{sum}} \leq u_i(S_i)$ for all agents $i \in \mathcal{A}$. When $q_j = 1$ for all $j \in \mathcal{O}$ and utilities are additive, this definition coincides with u_i^{prop} . Furthermore, it has some additional theoretical features which we will describe in Section 4.3. In order to obtain a measure to evaluate the degree of fairness of an allocation rather than simply stating whether the allocation is fair, we define the *ProportionalSumFairShare* measure as

$$\lambda^{prop_{sum}} = \min_{i \in \mathcal{A}} \frac{u_i(S_i)}{u_i^{prop_{sum}}}$$

In this way, we evaluate to which extent the fair share condition has been met by the agent who has received the lowest fraction of his or her fair share.

An alternative is to consider the average number of copies of the objects and use this to adjust u_i^{prop} . In that way, the utility of getting everything is adjusted by the reasonable fraction of the full set of objects an agent should expect. Formally we have

$$u_i^{prop_{av}} = \frac{\sum_{j \in \mathcal{O}} q_j}{m n} u_i(\mathcal{O})$$

This also coincides with u_i^{prop} when $q_i = 1$ even if utilities are not additive, but does not have similar theoretical properties as $u_i^{prop_{sum}}$. We say that an allocation, S , is proportional average share fair if $u_i^{prop_{av}} \leq u_i(S_i)$ for all agents $i \in \mathcal{A}$, and define the *ProportionalAverageFairShare* measure as

$$\lambda^{prop_{av}} = \min_{i \in \mathcal{A}} \frac{u_i(S_i)}{u_i^{prop_{av}}}$$

In Budish (2011), the max min fair share was introduced as an alternative to proportional fair share when the objects are indivisible. It is also used in Bouveret and Lemaître (2016) and in Procaccia and Wang (2014). The intuition is to consider the utility an agent can ensure himself if he divides the objects into m bundles, but is the last to choose. Formally, let

$$u_i^{mFS} := \max_{S \in \mathcal{S}} \min_{k \in \mathcal{A}} u_i(S_k)$$

An allocation of objects $S = \{S_1, \dots, S_m\}$ is max min share fair if $u_i^{mFS} \leq u_i(S_i)$ for all agents and the fairness measure *MaxMinFairShare* is defined as

$$\lambda^{mFS} = \min_{i \in \mathcal{A}} \frac{u_i(S_i)}{u_i^{mFS}}$$

In a symmetrically way, if an agent i chooses a bundle first after another agent has divided the objects, the worst that agent i can risk to get is the min max fair share suggested by Bouveret and Lemaître (2016) and given by

$$u_i^{MFS} := \min_{S \in \mathcal{S}} \max_{k \in \mathcal{A}} u_i(S_k)$$

The allocation is then min max share fair if $u_i^{MFS} \leq u_i(S_i)$ for all agents and we define the fairness measure *MinMaxFairShare* as

$$\lambda^{MFS} = \min_{i \in \mathcal{A}} \frac{u_i(S_i)}{u_i^{MFS}}$$

4.3 Envy-Free Fairness Measures

An important aspect to consider when allocating objects is that the agents should not be envious of each other. It is said that an allocation is envy-free if no agent strictly prefers another agent's bundle. That is

$$u_i(S_i) \geq u_i(S_k) \quad \forall i, k \in \mathcal{A}$$

In an envy-free allocation, every agent would therefore feel that the bundle he received is the best (or at least as good as the best) and would hence be satisfied with what he got. Envy-free allocations are studied in various settings in Kesten and Yazıcı (2012), Parkes et al (2012), and Procaccia (2013). In a setting with indivisible objects this is a very strict criterion. Therefore, Budish (2011) introduced a less strict version denoted 'envy-free bounded by a single good'. The idea is that if agent i strictly prefers another agent k 's bundle, then there should be an object $j \in S_k$ in k 's bundle such that i does not prefer $S_k \setminus \{j\}$ compared to his own bundle. I.e. if $\forall i, k \in \mathcal{A} : \exists j \in S_k : u_i(S_i) \geq u_i(S_k \setminus \{j\})$, then the allocation S is envy-free bounded by a single good.

Based on the ideas of envy-free allocations, we introduce two new fairness measures. The first is referred to as *TotalEnvy* and reflects the total amount of envy present in the system. We define

$$\lambda^{envytotal} = \sum_{i \in \mathcal{A}} \left(\max_{k \in \mathcal{A}} (u_i(S_k) - u_i(S_i)) \right)$$

where the envy of each agent is nonnegative by definition because the agent himself appears in the formula. An alternative is to consider the amount of envy felt by the agent who is the most envious. We denote this measure *MaxEnvy* and define it as

$$\lambda^{envymax} = \max_{i \in \mathcal{A}} \left(\max_{k \in \mathcal{A}} (u_i(S_k) - u_i(S_i)) \right)$$

Nice relations exist among the properties introduced in this section and in Section 4.2. For allocation problems with indivisible goods where several object can be allocated to each agent but no copies are allowed and utility functions are additive, Bouveret and Lemaître (2016) showed the following.

Proposition 1. *Let S be an allocation with $S_i \cap S_k = \emptyset \quad \forall i \neq k, \cup_{i \in \mathcal{A}} S_i = \mathcal{O}$. Then $u_i^{mFS}(S_i) \leq u_i^{prop}(S_i) \leq u_i^{MFS}(S_i)$ and the following holds: $CEEI \preceq$ envy-free \preceq min max share fair \preceq proportional share fair \preceq max min share fair.*

The notation $\Psi \preceq \Upsilon$ is used to denote that if the allocation satisfies property Ψ , then it also satisfies property Υ and hence, Ψ is stronger than Υ . CEEI refers to Competitive Equilibrium from Equal Income and originates from economic equilibrium theory. CEEI is presented in Budish (2011) and will not be considered further in this paper.

The above scale cannot directly be extended to the case with copies of the objects. The complication lies in the fact that an agent does not receive extra utility by receiving an additional copy of some object. With the *ProportionalSumFairShare* definition above, we can, however, generalize most of the statement if we require that all copies of all objects are allocated (an object can be allocated to an agent that gets zero utility for that object or excess objects can be removed). As in Proposition 1, we require utilities to be additive. We have the following.

Theorem 2. *Let S be an allocation with $\sum_{i \in \mathcal{A}} |S_i \cap \{j\}| = q_j \quad \forall j \in \mathcal{O}$. Then the following statements hold when utilities are additive*

- a) $u_i^{mFS} \leq u_i^{propsum}$ so proportional sum share fair \preceq max min share fair

<i>AverageUtility</i>	$\lambda^{AvUtility}$	↑
<i>MinimumUtility</i>	$\lambda^{MinUtility}$	↑
<i>MinimumNumber</i>	$\lambda^{MinNumber}$	↑
<i>SpanNumber</i>	$\lambda^{SpanNumber}$	↓
<i>UnassignedObjects</i>	$\lambda^{Unassigned}$	↓
<i>MaxLostUtility</i>	$\lambda^{MaxLostUtility}$	↓
<i>MaxLostNumber</i>	$\lambda^{MaxLostNumber}$	↓
<i>ProportionalSumFairShare</i>	$\lambda^{prop_{sum}}$	↑
<i>ProportionalAverageFairShare</i>	$\lambda^{prop_{av}}$	↑
<i>MaxMinFairShare</i>	λ^{mFS}	↑
<i>MinMaxFairShare</i>	λ^{MFS}	↑
<i>TotalEnvy</i>	$\lambda^{envy_{total}}$	↓
<i>MaxEnvy</i>	$\lambda^{envy_{max}}$	↓

Table 1: Overview of the measures used to evaluate the fairness of allocations.

b) $u_i^{prop_{sum}} \leq u_i^{MFS}$ so *min max share fair* \preceq *proportional sum share fair*

c) *envy-free* \preceq *min max share fair*

Proof. a) Let S be an arbitrary allocation with $\sum_{i \in \mathcal{A}} |S_i \cap \{j\}| = q_j \forall j \in \mathcal{O}$ and let i be any agent. Define x_{kj} to be 1 if bundle k includes object j and zero otherwise. As we have copies, an object j will be included in a bundle exactly q_j times, hence $\sum_{k \in \mathcal{A}} x_{kj} = q_j$ for all $j \in \mathcal{O}$. Combined with additive preferences this gives

$$\sum_{k \in \mathcal{A}} u_i(S_k) = \sum_{k \in \mathcal{A}} \sum_{j \in \mathcal{O}} w_{ij} x_{kj} = \sum_{j \in \mathcal{O}} w_{ij} \sum_{k \in \mathcal{A}} x_{kj} = \sum_{j \in \mathcal{O}} w_{ij} q_j$$

A bundle from an allocation that yields the lowest utility must give no more utility than the average utility of bundles in that allocation. So now we get

$$\min_{k \in \mathcal{A}} u_i(S_k) \leq \frac{1}{m} \sum_{k \in \mathcal{A}} u_i(S_k) = \frac{1}{m} \sum_{j \in \mathcal{O}} w_{ij} q_j = u_i^{prop_{sum}}$$

As S was an arbitrary allocation, the above also holds if maximum is taken over all allocations, hence it follows that

$$u_i^{mFS} = \max_{S \in \mathcal{S}} \min_{k \in \mathcal{A}} u_i(S_k) \leq u_i^{prop_{sum}}$$

The proof of b) follows in much the same manner and c) is proved exactly as in Bouveret and Lemaître (2016). \square

4.4 Overview of Fairness Measures

To summarize, Table 1 provides an overview of the measures used to evaluate allocations of objects to agents. To the right in the table, an arrow indicates if it is preferred to have high or low values of the specific measure.

5 Obtaining Fair Allocations

In this section, we present 14 methods for obtaining fair allocations. The methods can be categorized into four types. In Section 5.1, we present methods that seek to optimize the

measures presented in Section 4.1. In Section 5.2, the focus is on the fair share concept and Section 5.3 presents methods that seek to find envy-free allocations. Section 5.4 presents a method which simulates the practice of letting the agents take turns at choosing. Finally, in Section 5.5, we provide an overview of the methods used in this study.

In order to model the allocation problem as a classical transportation problem, we define a parameter for each agent $i \in \mathcal{A}$ and each object $j \in \mathcal{O}$ to indicate if it is possible to assign agent i to object j . We only allow agents to be assigned to objects they request, so $a_{ij} = 1$ if $w_{ij} > 0$ and 0 otherwise.

For each allocation S , we define a corresponding vector $\mathbf{x} = [x_{ij}] \in \{0, 1\}^{m \times n}$ of binary variables indicating if a copy of object j is assigned to agent i . Formally, we have

$$x_{ij} = \begin{cases} 1 & \text{if } |S_i \cap \{j\}| > 0 \\ 0 & \text{otherwise} \end{cases}$$

Corresponding to the set of feasible allocations \mathcal{S} , we define \mathcal{X} as the feasible set of vectors such that for each feasible allocation $S \in \mathcal{S}$, there is a unique vector $\mathbf{x} \in \mathcal{X}$ and vice versa. The set \mathcal{X} is defined as

$$\mathcal{X} = \{\mathbf{x} \in \{0, 1\}^{m \times n} \mid \sum_{i \in \mathcal{A}} a_{ij} x_{ij} \leq q_j \quad \forall j \in \mathcal{O}\}$$

where the constraints ensure that the capacities of the objects are not exceeded.

5.1 Optimizing Utility

The first method, referred to as *MaxAverage*, seeks to maximize the average utility gained by all agents, or equivalently, to maximize the total utility and thereby meet the $\lambda^{AvUtility}$ measure. This is done by solving the following model:

$$\begin{aligned} \max \quad & \sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{O}} w_{ij} x_{ij} \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X} \end{aligned}$$

The allocation obtained by this model will always be Pareto efficient so no agent can be better off by swapping with another agent without hurting some agent.

When 0/1-utilities are used, we work with a second version of the *MaxAverage* method which we refer to as *MaxAverageNormalized*. In this version, we incorporate the fact that there can be significant variation in the number of different objects an agent desires. Using normalized weights, we avoid favoring agents requesting many objects above those requesting only few, because larger weight will be given to each object of the latter when weights are normalized. For each agent i , we let the normalized weight for object j be

$$w'_{ij} = \frac{w_{ij}}{\sum_{j \in \mathcal{O}} w_{ij}} \tag{1}$$

The *MaxAverageNormalized* method works as *MaxAverage* but uses the normalized weights from (1).

The next method, referred to as *MaxMin*, seeks to maximize the utility of the agent that is worst off and thereby seeks to obtain the highest possible value for the $\lambda^{MinUtility}$ measure and secondarily for $\lambda^{AvUtility}$. This is done by a 2-step procedure where we first determine the highest utility we can possibly assign to the agent obtaining the lowest utility, i.e. $c_1 =$

$\max_{\mathbf{x} \in \mathcal{X}} \min_{i \in \mathcal{A}} \{\sum_{j \in \mathcal{O}} w_{ij} x_{ij}\}$. Next, we add c_1 as a lower bound on the utility obtained by any agent and maximize the average utility of the agents while respecting this bound. Formally, we have the following model:

$$\begin{aligned} & \text{lex max}(c_1, \sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{O}} w_{ij} x_{ij}) \\ & \text{s.t. } c_1 \leq \sum_{j \in \mathcal{O}} w_{ij} x_{ij} \quad \forall i \in \mathcal{A} \\ & \mathbf{x} \in \mathcal{X} \\ & c_1 \in \mathbb{R} \end{aligned}$$

Ideally, one would iteratively solve a model maximizing c_1 m times and fix the utility for one agent in each iteration, but the above approach allows us solve two models rather than m and still obtain good allocations. However, in Phase 1 of our analysis, we have used the iterative approach of solving the model m times because m is significantly smaller than in Phase 2 of the analysis.

For 0/1-utilities, we again supplement with a version of this method where we work with normalized weights based on (1). We refer to this method as *MaxMinNormalized*. Again, the fairness is measured directly based on the 0/1-utilities.

We create a variation of the above method that considers the number of objects assigned to the agent receiving the lowest number of objects, c_2 , rather than the agent receiving the smallest utility. Thereby, the method seeks to favor $\lambda^{MinNumber}$. This method, which is only relevant for preference points and normalized 0/1-utilities, is referred to as *MaxMinNumber* and can be described as follows:

$$\begin{aligned} & \text{lex max}(c_2, \sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{O}} w_{ij} x_{ij}) \\ & \text{s.t. } c_2 \leq \sum_{j \in \mathcal{O}} x_{ij} \quad \forall i \in \mathcal{A} \\ & \mathbf{x} \in \mathcal{X} \\ & c_2 \in \mathbb{R} \end{aligned}$$

Furthermore, we create a combination of *MaxMin* and *MaxMinNumber*, which we refer to as *MaxMinComb*, because the results in Phase 1 indicated that those two methods complement each other very well. As a result, this method is first introduced in Phase 2. We first maximize the number of objects, c_3 , assigned to the agent who receives the fewest. Then, adding c_3 as a constraint, we maximize the utility, c_4 , of the agent obtaining the lowest utility. Finally, adding both c_3 and c_4 as constraints, the sum of the utilities is maximized. This gives us:

$$\begin{aligned} & \text{lex max}(c_3, c_4, \sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{O}} w_{ij} x_{ij}) \\ & \text{s.t. } c_3 \leq \sum_{j \in \mathcal{O}} x_{ij} \quad \forall i \in \mathcal{A} \\ & c_4 \leq \sum_{j \in \mathcal{O}} w_{ij} x_{ij} \quad \forall i \in \mathcal{A} \\ & \mathbf{x} \in \mathcal{X} \\ & c_3, c_4 \in \mathbb{R} \end{aligned}$$

Finally, we have created two allocation methods which are based on minimization of the objects requested, but not received by an agent. In the first, which we refer to as *MinMaxLostUtility*, we seek to minimize the lost utility of the agent who loses the most, c_5 , and thereby favor $\lambda^{MaxLostUtility}$. For data with preference points, this is equivalent to maximizing the utility of the agent who obtains the smallest utility because all agents have the same potential utility t . When considering 0/1-utilities, it is, however, not equivalent. As before, our secondary object is to maximize the sum of utilities. We have

$$\begin{aligned} & \text{lex min}(c_5, \sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{O}} -w_{ij}x_{ij}) \\ & \text{s.t. } c_5 \geq \sum_{j \in \mathcal{O}} w_{ij}(a_{ij} - x_{ij}) \quad \forall i \in \mathcal{A} \\ & \mathbf{x} \in \mathcal{X} \\ & c_5 \in \mathbb{R} \end{aligned}$$

With *MinMaxLostNumber*, we seek to minimize the number of objects lost by the agent who loses most objects, c_6 , and thereby favor $\lambda^{MaxLostNumber}$. This is not equivalent to maximizing the number of objects received by the agent who gets the most. But for 0/1-utilities, it is equivalent to *MinMaxLostUtility*. We have:

$$\begin{aligned} & \text{lex min}(c_6, \sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{O}} -w_{ij}x_{ij}) \\ & \text{s.t. } c_6 \geq \sum_{j \in \mathcal{O}} (a_{ij} - x_{ij}) \quad \forall i \in \mathcal{A} \\ & \mathbf{x} \in \mathcal{X} \\ & c_6 \in \mathbb{R} \end{aligned}$$

5.2 Fair Share

In this section, we present two methods inspired by the fair share criteria. In the first method, referred to as *FairShare*, we recall the scale of the fairness criteria presented in Section 4.3 and based on that we seek to satisfy each of the three fair share criteria, *MaxMinFairShare*, *ProportionalSumFairShare*, and *MinMaxFairShare*, as well as possible in an iterative manner.

We start by determining the following three values for each agent $i \in \mathcal{A}$:

$$\begin{aligned} u_i^{mFS} &= \max \left\{ p_i \in \mathbb{R} \mid \begin{array}{l} p_i \leq \sum_{j \in \mathcal{O}} w_{ij}x_{kj} \quad \forall k \in \mathcal{A} \\ \mathbf{x} \in \mathcal{X} \end{array} \right\} \\ u_i^{propsum} &= \sum_{j \in \mathcal{O}} \frac{q_j}{m} u_i(\{j\}) \end{aligned}$$

and

$$u_i^{MFS} = \min \left\{ r_i \in \mathbb{R} \mid \begin{array}{l} r_i \geq \sum_{j \in \mathcal{O}} w_{ij}x_{kj} \quad \forall k \in \mathcal{A} \\ \sum_{i \in \mathcal{A}} a_{ij}x_{i,j} = q'_j \quad \forall j \in \mathcal{O} \\ x_{ij} \in \{0, 1\} \quad \forall i \in \mathcal{A}, j \in \mathcal{O} \end{array} \right\}$$

where q'_j is defined for all objects j as $q'_j = \min\{q_j, \sum_{i \in \mathcal{A}} a_{ij}\}$. That is, if the capacity of an object exceeds the number of agents requesting the object, then q'_j equals the number of agents

requesting the object. Alternatively, q'_j is simply the capacity. The second set of constraints in the definition of u_i^{MFS} ensures that all objects are distributed. Alternatively, the zero-allocation would be feasible and optimal.

Finally, the allocation is found by solving the following model. The variables, c_1, c_2 , and c_3 being iteratively maximized, ensure that each of the three fair share criteria are satisfied as much as possible before the next is considered. The last criterion seeks to maximize the total utility obtained by the agents.

$$\begin{aligned}
& \text{lex max} \left(c_1, c_2, c_3, \sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{O}} w_{ij} x_{ij} \right) \\
& \text{s.t. } u_i^{mFS} c_1 \leq \sum_{j \in \mathcal{O}} w_{ij} x_{ij} \quad \forall i \in \mathcal{A} \\
& \quad u_i^{propsum} c_2 \leq \sum_{j \in \mathcal{O}} w_{ij} x_{ij} \quad \forall i \in \mathcal{A} \\
& \quad u_i^{MFS} c_3 \leq \sum_{j \in \mathcal{O}} w_{ij} x_{ij} \quad \forall i \in \mathcal{A} \\
& \quad 0 \leq c_i \leq 1 \quad i = 1, 2, 3 \\
& \quad \mathbf{x} \in \mathcal{X}
\end{aligned}$$

The next method seeking to satisfy the fair share criteria is referred to as *PropFairShare* and is based on the *ProportionalAverageFairShare* measure. Here, we first determine the following value for each agent $i \in \mathcal{A}$:

$$u_i^{propav} = \frac{\sum_{j \in \mathcal{O}} q_j}{mn} u_i(\mathcal{O})$$

We then find the allocation by solving the following model where the variable c_4 being maximized first, ensure that the *ProportionalAverageFairShare* criterion is satisfied as well as possible and the second objective maximizes the total utility.

$$\begin{aligned}
& \text{lex max} \left(c_4, \sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{O}} w_{ij} x_{ij} \right) \\
& \text{s.t. } u_i^{propav} c_4 \leq \sum_{j \in \mathcal{O}} w_{ij} x_{ij} \quad \forall i \in \mathcal{A} \\
& \quad 0 \leq c_4 \leq 1 \\
& \quad \mathbf{x} \in \mathcal{X}
\end{aligned}$$

In Phase 1 of the analysis, we used $\sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{O}} x_{ij}$ rather than $\sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{O}} w_{ij} x_{ij}$ as the last objective in these two methods. However, additional tuning between the two phases indicated that using maximization of utility in the methods as described above, resulted in better allocations.

5.3 Avoiding Envy Among the Agents

In Section 4.3, we introduced the concept of envy-free allocations. Here, we present a method that will construct an envy-free allocation. We refer to this method as *NoEnvy*. Recall that an allocation is envy-free if $u_i(S_i) \geq u_i(S_k)$ for all $i, k \in \mathcal{A}$. Among all envy-free allocations, we

prefer one that maximizes the average utility of the agents. This is obtained by the following model:

$$\begin{aligned}
& \max \sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{O}} w_{ij} x_{ij} \\
& \text{s.t.} \quad \sum_{j \in \mathcal{O}} w_{ij} x_{ij} \geq \sum_{j \in \mathcal{O}} w_{ij} x_{kj} \quad \forall i, k \in \mathcal{A}, i \neq k \\
& \quad \mathbf{x} \in \mathcal{X}
\end{aligned}$$

A drawback of requiring that an allocation is envy-free is that it may leave some objects unassigned even though they are desired by some agents. This is studied in Kesten and Yazıcı (2012) where envy-free allocations are obtained by deleting all objects that are requested by more agents than the number of copies available.

The next method, which is referred to as *EnvyBounded*, is inspired by the concept of envy-free bounded by a single good, which was discussed in Section 4.3. However, instead of modeling this concept directly, we have chosen to model a more strict requirement. Consider the bundle assigned to agent k . In our model, we require that by removing one object from k 's bundle, none of the other agents may feel that k 's remaining bundle is worth more than their own bundle. So in our model, the object that is removed from k 's bundle does not vary with the agents.

Among the allocations that satisfy this requirement, the model will select the one that maximizes the total utility. Thereby, it seeks to favor $\lambda^{AvUtility}$. This means that when two agents are competing for a copy of the same object, the agent who will receive the highest utility from that object will get it provided that the resulting allocation is envy-free bounded by a single good.

In the following model, we use the binary variables y_{ij} as the fictive objects j assigned to agent i while not counting the one that is removed when comparing envy. In that way, $\sum_{j \in \mathcal{O}} y_{ij}$ represents the bundle of goods received by agent i minus the one object that is taken out of the actual bundle. We have the following:

$$\begin{aligned}
& \max \sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{O}} w_{ij} x_{ij} \\
& \text{s.t.} \quad \sum_{j \in \mathcal{O}} w_{kj} x_{kj} \geq \sum_{j \in \mathcal{O}} w_{kj} y_{ij} \quad \forall i, k \in \mathcal{A} i \neq k \\
& \quad y_{ij} \leq x_{ij} \quad \forall i \in \mathcal{A}, j \in \mathcal{O} \\
& \quad \sum_{j \in \mathcal{O}} y_{ij} \geq \sum_{j \in \mathcal{O}} x_{ij} - 1 \quad \forall i \in \mathcal{A} \\
& \quad \mathbf{x} \in \mathcal{X} \\
& \quad y_{ij} \in \{0, 1\} \quad \forall i \in \mathcal{A}, j \in \mathcal{O}
\end{aligned}$$

Furthermore, when preference points are used, we use an additional method which is a variation of *EnvyBounded* where we seek to maximize the total number of assigned objects rather than the utility that those objects provide. Therefore, in the *EnvyBoundedNumber* method, we replace the objective function by $\max \sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{O}} x_{ij}$.

5.4 Take Turn to Choose

The final method is referred to as *SequentialChoice* and is the only one that is purely algorithmic without any mathematical modeling.

The background is that Budish and Cantillon (2012) shows that a procedure where the agents take turns to choose an object performs relatively well. Here, we simulate this by creating a method where agents are iteratively assigned the available object they desire the most. This is a very intuitive way of distributing available objects and it is therefore interesting to compare its performance to that of the previous methods which focused directly on the performance measures. Algorithm 1, which is a slight modification of the one presented in Budish and Cantillon (2012), provides the details of the method.

Algorithm 1 Sequential Choice

Let $L_1 = [a_1, a_2, \dots, a_m]$ be a random permutation of the agents in \mathcal{A} .
Set $L_2 = [a_m, a_{m-1}, \dots, a_1]$
Set $L_3 = [a_{\lceil m/2 \rceil}, \dots, a_m, a_1, \dots, a_{\lfloor m/2 \rfloor}]$
Set $L_4 = [a_{\lfloor m/2 \rfloor}, \dots, a_1, a_m, \dots, a_{\lceil m/2 \rceil}]$
 $h = 1$
while $\mathcal{A} \neq \emptyset$ or $\mathcal{O} \neq \emptyset$ **do**
 for $t = 1$ to $\text{length}(L_h)$ **do**
 Let i be the agent at location $L_h[t]$
 $j' = \text{argmax}_{j \in \mathcal{O}} \{w_{ij}\}$
 Assign agent i to object j'
 Set $q_{j'} = q_{j'} - 1$
 if $q_{j'} = 0$ **then**
 Set $\mathcal{O} = \mathcal{O} \setminus \{j'\}$
 if $w_{ij} = 0 \forall j \in \mathcal{O}$ **then**
 Delete agent i from $L_h, h = 1, \dots, 4$
 $h = h + 1 \pmod{4}$

5.5 Overview of Allocation Methods

To summarize, Table 2 provides an overview of the different methods we use to obtain fair allocations. The second column of the table indicates the fairness measures favored by each method and the third column gives some comments when needed.

6 Evaluating Fairness

In Section 4, we presented 13 measures for evaluating the fairness of allocations and in Section 5, we presented 14 methods for obtaining such allocations.

In this section, we create a large number of allocations with the methods presented above and evaluate the fairness of the outcomes. This is done in two phases. First, in Phase 1, we use small data sets for creating allocations. The resulting allocations have been presented to 25 respondents who rated their fairness. All respondents are adults but not Operations Research specialists. The results of this analysis are presented in Section 6.1. The main purpose of Phase 1 is to obtain a better understanding of the way fairness is perceived and of characteristics that are evaluated as being important, such that this knowledge can guide us in Phase 2. In Phase 2, allocations are made for significantly larger data sets and the fairness of each allocation is evaluated by the measures presented in Section 4. This leads to recommendations as regards the choice of allocation model for different kinds of data.

Allocation method	Favors	Comment
<i>MaxAverage</i>	$\lambda^{AvUtility}$	
<i>MaxAverageNormalized</i>	$\lambda^{AvUtility}$	Only 0/1-utility
<i>MaxMin</i>	$\lambda^{MinUtility}, \lambda^{AvUtility}$	
<i>MaxMinNormalized</i>	$\lambda^{MinUtility}, \lambda^{AvUtility}$	Only 0/1-utility
<i>MaxMinNumber</i>	$\lambda^{MinNumber}, \lambda^{AvUtility}$	Introduced in Phase 2
<i>MaxMinComb</i>	$\lambda^{MinNumber}, \lambda^{MinUtility}, \lambda^{AvUtility}$	Introduced in Phase 2
<i>MinMaxLostUtility</i>	$\lambda^{MaxLostUtility}, \lambda^{AvUtility}$	Introduced in Phase 2
<i>MinMaxLostNumber</i>	$\lambda^{MaxLostNumber}, \lambda^{AvUtility}$	Introduced in Phase 2
<i>FairShare</i>	$\lambda^{mFS}, \lambda^{prop_{sum}}, \lambda^{MFS}, \lambda^{Unassigned}$	
<i>PropFairShare</i>	$\lambda^{prop_{av}}, \lambda^{Unassigned}$	
<i>NoEnvy</i>	$\lambda^{envy_{total}}, \lambda^{envy_{max}}, \lambda^{AvUtility}$	
<i>EnvyBounded</i>	$\lambda^{envy_{total}}, \lambda^{envy_{max}}, \lambda^{AvUtility}$	
<i>EnvyBoundedNumber</i>	$\lambda^{envy_{total}}, \lambda^{envy_{max}}, \lambda^{Unassigned}$	Only preference points Only Phase 1
<i>SequentialChoice</i>		

Table 2: Overview of the methods for obtaining allocations.

Agent	Object			
	1	2	3	fraction
1	1	1	1	0.67
2	1	1	0	0.50
3	1	1	0	0.50
4	1	0	1	0.50
5	0	1	1	1.00
6	1	0	0	1.00
Capacity	3	6	2	
# assigned	2	4	2	
$ \{i \in \mathcal{A} : w_{ij} > 0\} $	5	4	3	

Agent	Object			
	1	2	3	$u_i(S_i)$
1	30	1	69	30
2	30	70	0	70
3	45	55	0	55
4	20	0	80	20
5	0	35	65	35
6	100	0	0	100
Capacity	3	6	2	
# assigned	3	3	0	
$ \{i \in \mathcal{A} : w_{ij} > 0\} $	5	4	3	

Figure 1: Example of the way an allocation was presented in the survey. Left: 0/1-utilities. Right: Preference points.

6.1 Phase 1

In the first part of our analysis, we created a survey showing the allocations obtained for 8 small manually constructed data sets using a total of 10 methods for obtaining fair allocations. In 3 of the data sets, the agents' preferences were given as 0/1-utilities, and in the remaining sets a total of 100 preference points were distributed among the activities for each agent. The data was constructed in such a way that each measure should fail to perform well on at least one data set. This resulted in a total of 67 data-allocation instances for the respondents to evaluate.

Examples of instances are shown in Figure 1. Each instance contained information about the agents' preferences. For 0/1-utilities (left hand side), a 1 indicates that the agent request the object and zero means that the object is not requested by the agent. For preference points (right hand side), the values given are the weights that the agents puts on the objects. The assignment is shown in green, and red indicates requests that were not satisfied. Each instance also provided information on the number of available copies of each object (capacity) and some calculated characteristics which could help the evaluation.

Allocation	0	1	2	3	4	5	6	7	8	9	10	Mean	Var	St.Dev
1	0	0	0	0	0	3	3	3	7	3	6	7.88	2.86	1.69
2	0	0	1	0	0	1	4	3	7	3	6	7.80	3.75	1.94
3	0	0	0	0	0	0	1	7	6	4	7	8.36	1.66	1.29
4	0	0	0	0	0	0	1	6	7	4	7	8.40	1.58	1.26

Figure 2: Example of results obtained from the survey.

	Mean				Standard deviation			
	1	2	3	av	1	2	3	av
<i>MaxAverage</i>	1.6	1.2	1.2	1.3	2.1	1.5	1.5	1.7
<i>MaxAverageNormalized</i>	5.0	1.4	5.8	4.1	2.5	1.7	2.5	2.2
<i>MaxMin</i>	5.0	6.8	6.1	6.0	2.5	2.2	2.7	2.5
<i>MaxMinNormalized</i>	8.0	5.9	7.3	7.0	1.7	2.2	1.7	1.9
<i>FairShare</i>	7.7	5.2	7.3	6.7	1.6	2.8	1.7	2.1
<i>PropFairShare</i>	5.1	5.2	7.3	5.9	2.6	2.8	1.7	2.4
<i>NoEnvy</i>	5.0	6.0	5.0	5.3	2.7	2.1	3.5	2.8
<i>EnvyBounded</i>	2.3	5.1	5.2	4.2	2.5	3.0	2.9	2.8
<i>SequentialChoice</i>	7.8	6.9	6.8	7.1	1.8	1.8	2.8	2.1

Table 3: Mean and standard deviation of rating distributions for all data-allocation instances with 0/1-utilities.

For each instance, the respondents evaluated the fairness of the allocation. The fairness was rated on a scale from 0 to 10, where 0 should be given for the completely unfair allocation and 10 for the completely fair allocation. An analysis of this is presented in Section 6.1.1. Finally, the respondents were asked to state in their own words what they found to be important in their evaluation of fairness. A summary of this is presented in Section 6.1.2. 25 respondents answered the survey.

6.1.1 Quantitative Analysis and Findings

The results obtained from the survey take the form illustrated in Figure 2, which shows the distribution of the ratings given by the 25 respondents for each of four methods used to find allocations of a data set with 6 agents and 3 objects based on preference points.

Based on the rating distribution, the mean and standard deviation are calculated. These are shown for all data-allocation combinations in Table 3 for data with 0/1-utilities and in Table 4 for data with preference points. The tables also show the average values.

We first consider the ratings given to the data sets with 0/1-utilities, shown in Table 3. The methods that obtain high average ratings are *MaxMinNormalized*, *SequentialChoice*, and *FairShare*. If we consider the standard deviations, we see that the respondents generally agree on the high performance. Furthermore, it is clear that *MaxAverage* obtains a poor average rating, in particular when the utilities are not normalized. This is as we expected as there is no incentive to distribute the objects evenly among the agents when only the average is considered. The respondents seem to agree on this. On the other hand, the respondents strongly disagree when evaluating the two methods seeking to avoid envy.

Table 4 shows the means and standard deviations of the ratings given to the allocations made for the data sets with preference points. The first thing to note is that the respondents generally seem to agree on giving high ratings to the allocations obtained by *MaxMin*, *EnvyBounded*, and

	Mean						Standard deviation					
	1	2	3	4	5	av	1	2	3	4	5	av
<i>MaxAverage</i>	6.2	7.9	1.8	7.5	7.0	6.1	3.0	1.7	2.0	1.7	1.7	2.0
<i>MaxMin</i>	6.5	7.8	7.3	7.0	6.4	7.0	2.6	1.9	2.2	2.0	1.7	2.1
<i>FairShare</i>	6.5	8.4	6.4	6.8	6.9	7.0	2.3	1.3	2.4	1.9	1.7	1.9
<i>PropFairShare</i>	5.2	8.4	5.5	6.0	6.4	6.3	2.3	1.3	2.2	2.2	1.8	2.0
<i>NoEnvy</i>	2.8	2.9	5.6	5.9	6.1	4.7	2.9	2.7	3.0	2.5	2.7	2.8
<i>EnvyBounded</i>	6.5	8.4	5.4	7.5	7.6	7.1	2.6	1.3	2.5	1.7	1.2	1.9
<i>EnvyBoundedNumber</i>	2.4	2.9	4.8	2.5	2.8	3.1	2.1	1.9	2.6	2.2	2.1	2.2
<i>SequentialChoice</i>	6.1	4.9	6.5	5.5	6.9	6.0	2.5	2.1	2.3	2.2	1.2	2.0

Table 4: Mean and Standard deviation of rating distributions for all data-allocation instances with preference points.

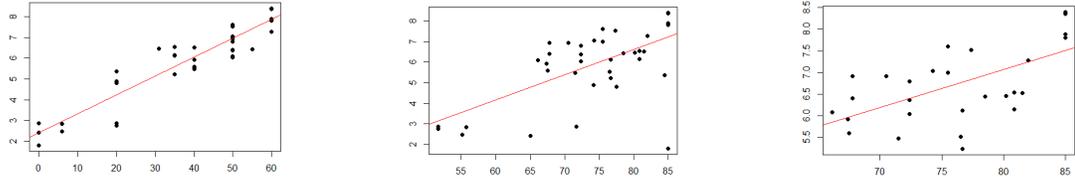


Figure 3: The average rating plotted against the Minimum Utility measure $\lambda^{MinUtility}$ (left) and the Average Utility measure $\lambda^{AvUtility}$ (middle and right), respectively.

FairShare. There is also a general agreement on giving a very low rating to *EnvyBoundedNumber*, while the respondents disagree more when evaluating *NoEnvy* even though the average rating for this method is low. *EnvyBoundedNumber* is discarded in Phase 2 due to this very low rating.

For the two utility based measures, *MinimumUtility* and *AverageUtility*, we check for a linear relation between the measure and the ratings obtained. This analysis is based on the data sets with preference points. Figure 3 shows the plots. It is quite clear from the left plot in Figure 3 and the supporting statistical test (not included in the paper), that a linear relation exists between $\lambda^{MinUtility}$ and the rating. Testing $\lambda^{AvUtility}$ against the rating does not support any obvious relationship. The plot is shown in the middle part of Figure 3. As we know that the rating is related to the utility of the agent who is worst off, we have performed an additional test in which the instances with $\lambda^{MinUtility} < 30$ are discarded. The resulting plot is shown in the right side of Figure 3. But even with these removed, a linear relation between the $\lambda^{AvUtility}$ measure and the rating is rejected. We conclude that the *MinimumUtility* measure yields a relatively good indication of the fairness viewed by humans, whereas *AverageUtility* does not. This will be valuable information when we turn to Phase 2 of our analysis.

6.1.2 Qualitative Analysis and Findings

The true benefits of the survey are, however, the free text information where the respondents were asked what they value as important when rating an allocation. Here we will discuss the main issues.

Half of the respondents stated that they found it important that the objects were distributed evenly among the agents, irrespective of the number of objects that each agent requested. This indicates that the respondents find it difficult to handle the heterogeneity of the agents as regards

the amount of objects they desire. The only methods that seek to meet this requirement of the allocations are the *MaxMinNumber* and *MaxMinComb*, both of which are new methods that we added to the study in Phase 2 based on this finding. In the situation with 0/1-utilities, the *MaxMin* (without normalization) will have a similar effect.

Almost 40% of the respondents stated that they find it important that all objects wanted by some agents are distributed. This is such an intuitive notion that we believe that more respondents agree with the statement without having mentioned it. After all, if an object is undistributed while some agents would like to have it, it just feels unfair. Interestingly enough, in order to obtain envy-free allocations, it can be necessary to leave some objects undistributed. All other methods will always distribute all objects that are requested.

Many respondents mentioned the importance of the number of points, but they did so in various ways. Some respondents stated that if an object is desired above its capacity, the copies should be given to the agents who have assigned the highest number of points to this object. In other words, those not receiving a copy should be those who value the object the least. Some respondents stated that it is important that every agent received the one object that he or she values the most. Once that happens, the rest is not very important. The only method that truly seeks to fulfill this is the *MaxAverage*, but this method conflicts significantly with the goal of distributing the objects evenly and the allocations created with this method received very low ratings, in particular when preference points were involved.

The final issue, which was mentioned by about 25% of the respondents, is that every agent should get at least one object in order for the allocation to be fair. This is a relaxed version of the goal of distributing the objects evenly and will be obtained by most of the methods if it is possible. Exceptions are *MaxAverage* and the fair share methods.

It was interesting to observe the difficulty the respondents had rating the fairness of the allocations. Some told us so directly, but it also became clear when we analyzed the results. Consider again the ratings given to the four allocations shown in Figure 2. The four allocations that the respondents were presented with here were in fact symmetric, i.e. they were identical except from reordering of agents and objects. It is clear from the ratings given that it is very hard to be completely consistent with one's own criteria for rating.

One explanation for this difficulty may be found in conflicts among the characteristics that the respondents are looking for. An example of this conflict can be seen in the following characteristics provided by one of the respondents:

1. If possible, each agent should receive at least one object.
2. When creating the allocation, higher preferences should (always) count more than lower preferences, as long as 1. is still satisfied.
3. Overall, the objects should be evenly distributed.
4. All objects should always be used if they are wanted by agents.

It should be noted that this respondent provides well-structured feedback, but still ends up with conflicts. Here, point 3 conflicts with point 4, and point 2 conflicts with point 3.

6.2 Phase 2

Having obtained a better insight in the way people rate the fairness of allocations, we now turn our attention to a larger analysis. For each data instance, we create an allocation using each of the allocation methods described above. For each of these allocations, we compute the value of each fairness measure. The purpose of this analysis is to identify one or several allocation

methods which perform well on all (or most of) the fairness measures - and in particular on those that were found to be important in Phase 1.

The average values of these fairness measures over 50 data sets are presented in Figures 4 through 6 below. For each measure we have used Excel's red-green-color scale to indicate the quality obtained by each of the methods, irrespective of whether high or low values are preferred. Consequently, method-measure combinations with dark green color are relatively good whereas those in red color are relatively bad.

A good allocation method is one that generally produces fair allocations when measured by the different fairness measures, in particular those that were found in Phase 1 to be important. In Figures 4 through 6 such a method is characterized by a column containing as many green cells as possible and preferably no red cells.

6.2.1 Data

A total of 150 data instances are created for this analysis. They are partitioned into three main groups, each containing 50 instances. In the data of group 1, the agents express their desire for the objects in the form of preference points, i.e. $\sum_{j \in \mathcal{O}} w_{ij} = t \forall i \in \mathcal{A}$ where $t = 100$. In the data of group 2, the agents merely provide information about which objects they would like - without prioritizing the objects. Hence, the data in this group uses 0/1-utilities. Finally in the data in group 3, the agents also provide 0/1 information, but the weights are normalized to $\sum_{j \in \mathcal{O}} w_{ij} = t \forall i \in \mathcal{A}$ where $t = 100$. The reasoning behind the separation between groups 2 and 3 is the following. Consider two agents, x and y . Agent x requests two objects (1 and 2) whereas agent y requests 4 objects (1, 2, 3, and 4). In group 2, agent y obtains the same utility by receiving objects 1 and 2 as agent x would, and thereby the potential utility obtainable by agent y is higher (receiving all four objects) than that of agent x . But in group 3, the total utility that each agent can get is assumed to be the same. Therefore, agent x must value objects 1 and 2 higher than agent y . Without the presence of preference points, we therefore say that an agent values his requested objects equally.

Each group contains a total of 50 instances, with 10 instances having $m = 20, 50, 100, 200$, and 500 agents, respectively. Some agents request few objects whereas others request many. The requested objects are determined stochastically along with the number of objects and the capacities of the objects, as will be explained below.

In each data instance, the number of objects, n , is determined randomly based on a normal distribution with mean $\mu = \frac{m}{3}$ and variance $\sigma^2 = \frac{m+100}{20}$. However, the distribution is truncated from below at the value 3 and from above by m .

The capacity q_j of each object $i \in \mathcal{O}$ was drawn randomly as an integer between minCap and maxCap based on a uniform distribution, where

$$\text{minCap} = \begin{cases} 2, & 6 \leq m < 50 \\ 10, & 50 \leq m < 500 \\ 20, & 500 \leq m \end{cases}$$

and

$$\text{maxCap} = \begin{cases} m, & 6 \leq m < 50 \\ 50, & 50 \leq m < 500 \\ 100, & 500 \leq m \end{cases}$$

In the 50 instances with preference points, we have created three different agent types, T_1, T_2 , and T_3 , divided evenly (up to an integer) among the m agents in the data set. For each agent a total weight of $t = 100$ is distributed among the objects.

We iteratively select an object to be desired by the agent and assign a weight to it. All objects not yet requested by the agent are equally likely to be selected. We implicitly round all numbers to integers and possibly assign less than the computed weight to the last object selected in order to reach $t = 100$. We define the parameter $P = \frac{100}{\min(30,n)}$ and consider each agent type in turn. $U(\cdot, \cdot)$ is used to denote a uniform distribution.

T1: The number of objects requested by this type of agent is drawn from a stochastic variable X with distribution

$$X = \begin{cases} U(2, n), & n < 10 \\ U(5, n), & 10 \leq n \leq 25 \\ U(5, 25), & 25 < n \end{cases}$$

The weights of the objects are the same, except from possibly the last one.

T2: This type of agent puts high weights on a few much desired objects and has little interest in the remaining objects. For this agent, we uniformly select an unrequested object and draw the weight to assign to it from a stochastic variable with distribution

$$U(3P, \min(8P, 100)).$$

This is repeated until at least 90 weight points are distributed (and never more than $t = 100$). The remaining 0-10 points are given to less valued objects where the weight is drawn from a stochastic variable with distribution:

$$U(1, P).$$

T3: The last type of agent is less extreme. The weight given to an object is draw from the distribution:

$$U(0.5P, 3P)$$

except from the last object which may receive a lower weight.

We now turn our attention to the 50 data instances with 0/1-utilities. For each agent, we draw the number of objects to be requested from normal distribution with mean $\mu = \frac{\min(n,20)}{2}$ and variance $\sigma^2 = 7$ and truncated from below at 3 and from above at $\min(n, 30)$. As we know the amount of objects to request, the selection of objects is done uniformly at random.

Finally, to generate the 50 data instances with normalized 0/1-utilities, the objects requested by each agent are determined like for the 0/1-utility instances. Next, the $t = 100$ weight points are shared evenly among these objects.

6.2.2 Results with Preference Points

Figure 4 presents the results obtained for the data sets with preference points. For each method, the figure provides the average value of each fairness measure based on 50 data sets. The first thing to notice is that the *MinMaxLostNumber* allocation method performs relatively poorly on the majority of measures and can therefore be discarded. We can draw the same conclusion as regards *MaxAverage* because the only measure where this method performs better than the others is $\lambda^{AvUtility}$. However, this measure was found in Phase 1 not to be of significant importance. Furthermore, the variation as regards this measure is relatively small across the methods.

	MaxAverage	MaxMin	MinMaxLostUtility	MaxMinNumber	MaxMinComb	MinMaxLostNumber	FairShare	PropFairShare	NoEnvy	EnvyBounded	SequentialChoice
AverageUtility	88.1	86.3	87.9	86.4	84.8	88.0	88.1	87.6	88.1	85.6	
MinimumUtility	55.7	79.4	54.8	78.0	58.4	56.8	57.1	57.7	57.5	58.1	
MinimumNumber	3.8	3.5	4.3	4.3	3.0	3.8	3.9	3.9	3.8	4.0	
SpanNumber	11.3	12.7	10.6	11.7	13.8	11.0	11.2	10.5	10.8	10.0	
UnassignedObjects	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	
MaxLostUtility	44.3	20.6	45.2	22.0	41.6	43.2	42.9	42.3	42.5	41.9	
MaxLostNumber	9.0	5.3	9.0	5.4	3.0	8.7	8.7	8.4	8.5	7.8	
ProportionalSumFairShare	2.4	2.8	2.3	2.8	2.3	2.4	2.4	2.5	2.4	2.6	
ProportionalAverageFairShare	2.4	3.6	2.4	3.5	2.6	2.4	2.5	2.5	2.5	2.6	
MaxMinFairShare	3.0	3.7	3.0	3.6	3.5	3.1	3.1	3.2	3.1	3.1	
MinMaxFairShare	1.0	1.2	1.0	1.2	0.9	1.1	1.1	1.2	1.1	1.1	
TotalEnvy	65.0	95.4	60.4	63.6	275.6	60.7	61.8	0.0	35.6	45.4	
MaxEnvy	10.2	11.6	10.0	10.3	24.6	9.2	9.2	0.0	6.0	10.2	

Figure 4: Average performance of each of the methods used to obtain allocations for the 50 data sets based on preference points. Each column corresponds to one of the methods whereas each row provides a performance measure.

The remaining methods can be partitioned into two groups. As expected, the two envy-based methods perform best on the envy-based measures $\lambda^{envy_{total}}$ and $\lambda^{envy_{max}}$, but not particularly well on the others.

The methods *MaxMin*, *MaxMinComb*, and *MinMaxLostUtility* perform very well on all measures except the two envy-based ones and $\lambda^{AvUtility}$ and $\lambda^{SpanNumber}$. As regards *MaxMinComb*, and *MinMaxLostUtility*, the difference across the methods is generally small (few utility points) and $\lambda^{AvUtility}$ was found in Phase 1 to be a less important measure. Note that in the current situation with preference points, the two methods *MaxMin* and *MinMaxLostUtility* are identical because, based on 100 points, one maximizes the utility of the agent who receives fewest points, whereas the other minimizes the utility lost by the one who loses the most.

The remaining methods show average performance for all measures except $\lambda^{AvUtility}$. With a few exceptions, these methods are outperformed by *MaxMin*, *MaxMinComb*, and *MinMaxLostUtility* on all performance measures except $\lambda^{AvUtility}$, which was found in Phase 1 not to be important. These measures are therefore not candidates for being the overall best.

When we compare *MaxMin* (or equivalently *MinMaxLostUtility*) to *MaxMinComb*, which show equal performance, the latter is found to have an advantage because the total envy in the allocations made with this method is smaller. When we compare the two envy-based methods, we favor *EnvyBounded* because Phase 1 showed that it is unacceptable to have unassigned objects which are wanted by some agent and *NoEnvy* cannot meet this requirement.

The final choice is between *MaxMinComb* and *EnvyBounded*. Here, our preferences go towards *MaxMinComb* because it is clear from the measures $\lambda^{MinUtility}$, which was found in Phase 1 to be important, and $\lambda^{MaxLostNumber}$ that this method is better at distributing the objects evenly, which was also found in Phase 1 to be important.

6.2.3 Results with 0/1-Utilities and Normalized 0/1-Utilities

Figure 5 presents the results obtained for the data sets based on normalized 0/1-utilities, which are equivalent to assuming that the agents prefer the objects they request equally much. The

	<i>MaxAverage</i>	<i>MaxMin</i>	<i>MaxMinNumber</i>	<i>MaxMinComb</i>	<i>MinMaxLostUtility</i>	<i>MinMaxLostNumber</i>	<i>NoEnvy</i>	<i>EnvyBounded</i>	<i>FairShare</i>	<i>PropFairShare</i>	<i>SequentialChoice</i>
<i>AverageUtility</i>	87.7	85.1	87.7	85.3	85.1	84.3	87.2	87.7	87.6	87.6	85.8
<i>MinimumUtility</i>	52.7	77.6	53.8	77.4	77.6	66.8	56.9	55.8	56.0	56.8	62.7
<i>MinimumNumber</i>	2.9	2.9	3.0	3.0	2.9	2.8	3.0	3.0	3.0	3.0	2.6
<i>SpanNumber</i>	13.1	15.4	13.1	15.3	15.4	17.1	12.7	12.9	13.0	13.0	14.3
<i>UnassignedObjects</i>	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0
<i>MaxLostUtility</i>	47.3	22.4	46.2	22.6	22.4	33.2	43.1	44.2	44.0	43.2	37.3
<i>MaxLostNumber</i>	8.7	3.8	8.6	3.8	3.8	2.2	8.1	8.3	8.2	8.2	6.4
<i>ProportionalSumFairShare</i>	1.8	2.1	1.8	2.1	2.1	1.9	1.9	1.9	1.9	1.9	2.0
<i>ProportionalAverageFairShare</i>	1.8	2.7	1.8	2.6	2.7	2.3	1.9	1.9	1.9	1.9	2.1
<i>MaxMinFairShare</i>	4.0	5.1	4.1	5.1	5.1	5.1	4.3	4.2	4.2	4.3	4.8
<i>MinMaxFairShare</i>	0.9	1.1	1.0	1.1	1.1	1.0	1.0	1.0	1.1	1.0	1.1
<i>TotalEnvy</i>	47.0	69.5	36.7	65.3	69.7	242.8	0.0	23.2	43.7	29.6	44.1
<i>MaxEnvy</i>	9.3	12.8	8.5	11.9	12.8	22.0	0.0	4.3	7.6	7.0	13.4

Figure 5: Average performance of each of the 11 methods used to obtain allocations for the 50 data sets based on normalized 0/1-utilities.

conclusions to be drawn from these results are very similar to those based on the data with preference points and indeed the two situations are also comparable. When the 0/1-utility data is treated and measured in a normalized form, *MaxMin* (or equivalently *MinMaxLostUtility*) and *MaxMinComb* are superior to the other methods. An interesting point to note in this respect is that the *SequentialChoice* method provides a good balance between focusing on avoiding envy and favoring the remaining measures.

Finally, we turn our attention to the results for the 0/1-utility data, which is presented in Figure 6. When considering 0/1-utility, the three methods *MaxMin*, *MaxMinNumber*, and *MaxMinComb* are equivalent and are therefore presented jointly in the table. The same goes for *MinMaxLostUtility* and *MinMaxLostNumber*.

It is clear that the *MaxMinNormalized* method is superior to the others. This is an interesting finding because this method treats the data as if the 0/1-utilities were normalized but measuring the quality of the allocations based on 0/1-utilities without normalization. Hence, using normalization in the planning is advantageous even though it is not evaluated based on normalization.

The two methods, *MinMaxLostUtility* and *MinMaxLostNumber*, that focus on the objects that the agents do not get rather than on what they get, perform relatively poorly on most of the measures. However, the two envy-based methods and *SequentialChoice* perform relatively well, whereas *MaxMinComb*, being superior for the other types of data, is not particularly good for this type. The superior performance of *MaxMinNormalized* and *SequentialChoice* supports the findings of Phase 1.

The results presented in Figures 5 and 6 are based on the same data sets, the only difference being the normalization of the data in Figure 5. In a situation where a planner is presented with data based on 0/1-utilities, a decision must be made about whether to normalize the data and thereby actively respond to the fact that not all agents request the same number of objects. It is interesting to observe the large difference in the relative performance of the different methods in the two figures.

However, it seems that a decision to normalize the data and then seek to maximize the utility of the agent receiving the least number of objects is generally a good choice independently of whether the subsequent scale of measuring is based on preference points or 0/1-utilities.

	MaxAverage	MaxAverageNormalized	MaxMin, MaxMinComb, MaxMinNumber	MaxMinNormalized	MinMaxLastUtility, MinMaxLastNumber	FairShare	PropFairShare	NoEnvy	EnvyBounded	SequentialChoice
AverageUtility	8.7	8.7	8.7	8.7	8.7	8.7	8.7	8.7	8.7	8.7
MinimumUtility	1.7	2.9	3.0	2.9	1.4	1.7	1.9	2.3	2.0	2.6
SpanNumber	18.1	13.1	17.5	15.4	18.9	18.6	18.3	15.6	17.0	14.4
UnassignedObjects	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0
MaxLostUtility	5.6	8.7	5.8	3.8	2.2	5.6	5.6	5.8	5.4	6.7
ProportionalSumFairShare	1.6	1.8	1.8	2.1	1.5	1.7	1.8	2.0	1.9	2.0
ProportionalAverageFairShare	1.5	1.8	1.7	2.7	1.5	1.4	1.6	1.9	1.7	2.1
MaxMinFairShare	4.7	4.0	4.7	5.1	5.1	4.9	4.9	4.7	4.7	4.6
MinMaxFairShare	0.7	0.9	0.9	1.1	0.7	1.0	1.0	1.1	1.0	1.0
TotalEnvy	24.0	4.4	30.4	5.5	30.6	32.8	31.6	0.0	8.0	2.5
MaxEnvy	2.4	1.1	2.6	0.8	2.0	2.7	2.5	0.0	0.8	0.7

Figure 6: Average performance of each of the 13 methods used to obtain allocations for the 50 data sets based on 0/1-utilities.

This approach corresponds to *MaxMin* in Figure 5 and *MaxMinNormalized* in Figure 6. Apart from this general conclusion, the performance quality of the methods differs significantly across the two approaches, which stresses the importance of actively considering how fairness is best measured in a given situation.

7 Concluding Remark

We set out to answer the two questions: 'How do we measure fairness?' and 'How do we obtain a fair allocation?'. Realizing that the academic literature provides many different fairness measures and that many papers invented their own measures - often without specific explanation - we started our study by a thorough review of measures used for this and related problems. This was supplemented by a few new ways of viewing fairness of allocations and we ended up with a total of 13 fairness measures. A similar approach was followed as regards the methods for obtaining allocations and led us to 14 allocation methods, 13 of which are based on linear programming.

Our analysis contained two parts. First, we presented a number of allocations to a number of human beings and asked them to rate the fairness of these allocations, as well as provide information about their general perception of fairness as regards allocations. Second, we constructed a large number of allocations based on the allocation methods reviewed and evaluated their fairness based on each of the reviewed fairness measures.

Our main conclusion is that, in general, allocations constructed by the method *MaxMinComb* or *MaxMinNormalized*, depending on the type of data, are fair by the majority of the measures and in particular by those found to be important by humans. Furthermore, we found that the only method that can ensure that no agent is envious on some other agent, namely *NoEnvy*, is not particularly fair when evaluated by other measures.

It is our hope that the review and analysis presented in this paper will help guide future researchers and practitioners within the area to select which allocation method and fairness measures to use on a more informed basis.

References

- Abdulkadiroğlu A, Sönmez T (2003) School choice: A mechanism design approach. *Amer Econom Rev* 93(3):729–747
- Bezáková I, Dani V (2005) Allocating indivisible goods. *ACM SIGecom Exchanges* 5(3):11–18
- Boche H, Schubert M (2009) Nash bargaining and proportional fairness for wireless systems. *IEEE/ACM Trans Networking* 17(5):1453–1466
- Bonald T, Massoulié L, Proutière A, Virtamo J (2006) A queueing analysis of max-min fairness, proportional fairness and balanced fairness. *Queueing Systems* 53(1-2):65–84
- Bouveret S, 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
- Budish E (2011) The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *J Political Econom* 119(6):1061–1103
- Budish E, Cantillon E (2012) The multi-unit assignment problem: Theory and evidence from course allocation at harvard. *Amer Econom Rev* 102(5):2237–2271
- Carvalho A, Larson K (2012) Sharing rewards among strangers based on peer valuations. *Decision Anal* 9(3):253–273
- Chen Y, Lai J, Parkes D, Procaccia A (2013) Truth, justice, and cake cutting. *Games Econom Behavior* 77(1):284–297
- Cole R, Gkatzelis V, Goel G (2013) Mechanism design for fair division: Allocating divisible items without payments. In: *Proc.14th Conf. Electronic Commerce, ACM, New York*, pp 251–268
- Ernemann C, Hamscher V, Yahyapour R (2004) Benefits of global grid computing for job scheduling. In: *Proc. 5th IEEE/ACM Internat. Workshop on Grid Comput.*, pp 374–379
- Fedders L (2012) Ungt par finder metode til bedre fordeling af kbu-forløb. <http://www.dagensmedicindk>
- Feitelson D, Rudolph L, Schwiegelshohn U, Sevcik K, Wong P (2005) Theory and practice in parallel job scheduling. In: *Job Scheduling Strategies for Parallel Processing, Springer, LNCS*, pp 1–34
- Fiedrich F, Gehbauer F, Rickers U (2000) Optimized resource allocation for emergency response after earthquake disasters. *Safety Science* 35(13):41–57
- Han Z, Ji Z, Liu K (2005) Fair multiuser channel allocation for ofdma networks using nash bargaining solutions and coalitions. *IEEE Trans Communications* 53(8):1366–1376
- Hild M (2001) Fair kidney allocation based on waiting time. *Analyse und Kritik* 23:173–190
- Kahneman D (2012) *Thinking, Fast and Slow*. Penguin Random House, UK
- Kelly F (1997) Charging and rate control for elastic traffic. *Eur Transactions on Telecommunications*

- Kesten O, YazıcıA (2012) The pareto-dominant strategy-proof and fair rule for problems with indivisible goods. *Econom Theory* 50(2):463–488
- Leung V, Sabin G, Sadayappan P (2010) Parallel job scheduling policies to improve fairness: A case study. In: 39th Internat. Conf. Parallel Processing Workshops, pp 346–353
- Nguyen T, Vohra R (2012) The allocation of indivisible objects via rounding. Tech. rep.
- Othman A, Sandholm T, Budish E (2010) Finding approximate competitive equilibria: Efficient and fair course allocation. In: Proc. 9th Internat. Conf. Autonomous Agents and Multiagent Systems, vol 1, pp 873–880
- Pápai S (2001) Strategyproof and nonbossy multiple assignments. *J Public Econom Theory* 3(3):257–271
- Parkes D, Procaccia A, Shah N (2012) Beyond dominant resource fairness: Extensions, limitations, and indivisibilities. In: Proc. 13th ACM Conf. on Electronic Commerce, pp 808–825
- Procaccia A, Wang J (2014) Fair enough: Guaranteeing approximate maximin shares. In: Proc. 15th ACM Conf. Econom. and Computation, pp 675–692
- Procaccia D (2013) Cake cutting: Not just child’s play. *Communications of the ACM* 56(7)
- Sabin G, Sadayappan P (2005) Unfairness metrics for space-sharing parallel job schedulers. In: *Job Scheduling Strategies for Parallel Processing*, Springer, pp 238–256
- Tóth Š (2014) A case for a multifaceted fairness model: An overview of fairness methods for job queuing and scheduling. *Memics* 2014 pp 113–124