TECHNISCHE UNIVERSITÄT MÜNCHEN
Fakultät für Informatik
Lehrstuhl für Informatik XVIII

# Mechanism Design for Combinatorial Allocation Problems without Quasilinear Utilities

Sören Merting

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzende: Prof. Dr. Susanne Albers

Prüfer der Dissertation: 1. Prof. Dr. Martin Bichler
2. Priv.-Doz. Dr. Georg Groh

Die Dissertation wurde am 08.04.2019 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 13.07.2019 angenommen.

# Abstract

In combinatorial allocation problems, indivisible objects have to be assigned to selfish agents. A standard assumption for those problems is that agents have quasilinear utility functions. However, in many environments either money cannot be exchanged or agents cannot be assumed to maximize payoff. We focus on two specific non-quasilinear environments. First, we analyze a course allocation problem where students have preferences over schedules and report on a large-scale course assignment application at the TU Munich. Second, we study non-quasilinear utility functions as they have been reported for display ad auctions, and propose a truthful randomized approximation mechanism.

# Zusammenfassung

In kombinatorischen Allokationsproblemen müssen unteilbare Objekte an eigennützige Agenten vergeben werden. Eine Standardannahme für solche Probleme ist, dass Agenten quasilineare Nutzenfunktionen haben. In vielen Umgebungen kann jedoch Geld nicht verwendet werden oder Agenten maximieren nicht den Gewinn. Wir fokussieren uns auf zwei spezielle nicht-quasilineare Umgebungen. Zunächst analysieren wir ein Kursvergabeproblem, bei dem Studenten Präferenzen über Stundenpläne haben und berichten über dessen Anwendung an der TU München. Außerdem analysieren wir nicht-quasilineare Nutzenfunktionen, wie für Display-Ad Auktionen und stellen einen anreizkompatiblen randomisierten Approximationsmechanismus vor.

# Acknowledgements

During my doctoral studies I got great support from many people, which I appreciate a lot.

I am thankful to my supervisor Prof. Martin Bichler for his trust and support, and for the freedom he gave me in research. I appreciate your great support and really enjoyed our collaboration. Furthermore, I want to express my gratitude to Dr. Georg Groh for reviewing my thesis.

I liked the the work at our chair and the time with the colleagues a lot. Especially the cooperation with my mentor Dr. Paul Karänke was great and I enjoyed our constructive discussions. Furthermore, I thank you for your support, feedback, and linguistic reviews.

I would like to thank all colleagues and former students who contributed to the matching project. Especially I thank our admin Oliver Jacksch for his technical support, but I also thank Aykut Uzunoglu, Kevin D. Falkenstein, and Martin Rauch for our constructive discussions and their great work in implementing the Matching System.

Last but not least I thank my parents for their incalculable support and patience during the whole time of my education.

# Contents

# 1 Introduction

In combinatorial allocation problems indivisible objects have to be distributed among selfish agents (bidders), while the agents demand several objects or subsets (bundles) of these objects. That is, there are dependencies between the different objects that may lead to inefficiencies when allocating them separately.

To allocate the objects we use mechanisms. A mechanism is a function that maps from the space of possible actions of the agents to a set of possible outcomes. Agents choose strategies in a mechanism, whereby a strategy is a function from the valuations (or preferences) to actions. A strategy is dominant if the agent chooses his actions regardless of the actions played by other agents. We assume that agents behave rational, i.e., they only participate in mechanisms making them not worse off (individual rationality).

According to the revelation principle, we focus on direct mechanisms to allocate the objects. In a direct mechanism, the only action available to the agents is to report their valuations or preferences. Furthermore, we are interested in (Pareto) efficient mechanisms, where no agent can get a better outcome without assigning any other agent to a less preferred outcome. This is necessary to ensure that agents have no incentives to subvert the mechanism while trading outside. A stronger desideratum is strategy-proofness (or incentive compatibility), where the reporting of wrong valuations or preferences never leads to a better outcome for any agent, i.e., bidding truthful is a dominant strategy.

Mechanism design seeks a way to combine the desiderata of finding an efficient and fair allocation in a tolerable amount of time, while the agents should be incentivized to submit their true preferences over the (bundles of) objects. More precisely, we need to address the following difficulties.

- Computational complexity: The problem of finding an optimal allocation that respects the demand constraints of the agents and the capacities of the objects is an $\mathcal{NP}$-hard optimization problem.
- Valuation complexity: There are exponentially many different bundles of objects. The agents need to define valuations for each of these bundles, or at least a (complete) ordering over these bundles (ordinal preferences).
- Communication complexity: Since the valuations or preferences are private information of the agents, they also need to submit their valuations or preferences over the exponentially many bundles to the allocation mechanism.
- Strategic complexity: Since there is a huge amount of different possibilities to rank bundles, agents might think of how they should report their preferences or valuations to improve their personal outcome.

During the last sixty years, much research was done in addressing these complexities for combinatorial *auctions* leading to auction formats like the strategy-proof Vickrey-Clarke-Groves (VCG) mechanism (Vickrey, 1961; Clarke, 1971; Groves, 1973) or core-selecting auctions prohibiting justified envy (Day and Milgrom, 2008). However, these results base on fundamental assumptions on the bidder model. Usually bidders with a quasilinear utility function and without any budget constraints are considered, i.e., the utility of an outcome (a bundle) is the bidders valuation minus the price she has to pay for the bundle she got allocated. If one deviates from these assumptions it is impossible to find incentive-compatible deterministic mechanisms that are not dictatorial (Gibbard, 1973; Satterthwaite, 1975).

However, there are markets where bidders are not interested in pure payoff-maximization. For example, literature on digital advertising auctions suggests that automated bidders in display ad auctions rather maximize value subject to a budget constraint as in a knapsack problem (Feldman et al., 2008; Berg et al., 2010;

Zhou et al., 2008; Lee et al., 2013; Chen et al., 2011; Zhang et al., 2014), which differs from payoff maximization assumed in a quasilinear utility function.

Only a few papers study mechanism design with non-quasilinear utility models. Kazumura and Serizawa (2016) show for auctions, where different objects are sold, that there is no strategy-proof and Pareto efficient mechanism if only one bidder has multi-unit demands. Similarly, Baisa (2017) shows that if bidders have multi-dimensional types, there is no mechanism that satisfies individual rationality, strategy-proofness, Pareto efficiency, and budget balance for homogeneous goods.

Fadaei and Bichler (2017b) introduce a model of *value bidders* who maximize the value of bundles of objects for which they are given financial limits reflecting their valuation that they must not overbid. The authors show that with a truthful and deterministic mechanism for value bidders only an $n$-approximation can be achieved, where $n$ is the number of bidders. This means, the social welfare of the computed solution, i.e., the sum of the utilities of all market participants, is not worse than $1/n$ times the optimal social welfare ignoring incentives. Furthermore, Fadaei and Bichler (2017b) show that with randomized mechanisms an $\mathcal{O}(\sqrt{m})$ approximation can be achieved, where $m$ is the number of objects. These results illustrate that without quasilinearity truthful mechanisms with good approximation ratios might often not be feasible.

Using more structural information about the bidders valuation in specific markets, we can further improve the approximation ratios of Fadaei and Bichler (2017b). Bidders in display ad markets typically have a given overall budget, which they consider as sunk cost devoted to a campaign. The task is to invest the campaign budget such that the bidder's sum of valuations of won objects is maximized and the total payment is not higher than the budget constraint. We refer to such bidders as *knapsack bidders* and propose a truthful in expectation mechanism that provides a 4-approximation in Chapter 4.

For many applications monetary transfers are not permitted (e.g. course assignment) or because of low margins the agents are not willing to pay for allocations (e.g. reservation of time-slots at warehouses). Hence, auctions cannot be used to allocate the objects. However, without the payment rule, one looses a degree of freedom to adjust the mechanisms to achieve desired properties. That is, for any mechanism solving the allocation problem all desired properties have to be ensured via the allocation rule.

*Matching with preferences* considers the allocation of objects to agents or agents to agents without payments. However, in contrast to the development in auctions, in matching theory most research is about *single-minded* agents, i.e., agents who are only interested in getting assigned one single object or agent. Only recently combinatorial markets attract attention of researchers in matching theory.

A first seminal approach to address the combinatorial assignment problem (matching problem with complementarities) was published by Budish (2011). The work was breaking new ground, but the proposed mechanism is also challenging. Nguyen et al. (2016) recently provided two randomized mechanisms for one-sided matching problems with (limited) complementarities, one for agents having cardinal and one for ordinal preferences over bundles of objects.

We implemented these and further approaches and extended a system for the allocation of courses to students such that the students can submit preferences for whole schedules (bundles of courses). We compare the mechanism of Nguyen et al. (2016) to the standard approach in course allocation, First-Come First-Served (FCFS), and observe that both approaches are surprisingly similar in various metrics.

But even if it is possible to solve the combinatorial assignment problem in an acceptable amount of time, one still has to overcome the valuation and communication complexity. We cannot expect, that students submit an ordering over exponentially many schedules. Therefore, we developed a preference elicitation tool that generates such preference lists out of only a few intuitive input parameters and knowledge about usual preferences of students, like the distribution of courses during the day and the week, or the length of breaks between courses.

## 1.1 Outline

Before discussing the combinatorial problem and the extended matching system, we first consider the single-minded case in Chapter 2. We present the one-sided and two-sided matching problem, and present truthful mechanisms computing an allocation. Thereafter, we describe the matching system for the allocation of single courses implemented at the Computer Science Department at the Technical University of Munich (TUM) and discuss why the allocation of tutorials to students is not efficient in its current state.

Motivated by the need of a mechanism that allocates places for tutorials for different lectures, while considering prefernces over whole schedules, we present the combinatorial assignment problem in Chapter 3. We first introduce design desiderata for deterministic and randomized assignments before discussing different mechanisms to solve such matching problems. Since one cannot expect that students are able to submit an ordering over exponentially many different schedules (bundles), we discuss a new way to elicit preferences. We present a preference elicitation tool where students are only required to submit a few intuitive parameter, like time constraints, preferences over weekdays or the need of breaks between courses. We compare our approach to the method proposed in Budish et al. (2017). Thereafter, we present the results of three large field experiments at TUM and compare the mechanism proposed by Nguyen et al. (2016) to the wide-spread FCFS mechanism. Furthermore, we analyze the structure of the lottery from summer term 2017 in more detail and finally present results of a survey among the participants of the matching from winter term 2017/2018. Afterwards, we present an alternative application of matching with complementarities (coordination of time-slots at warehouses), where the agents are able to express cardinal valuations for the different bundles. We conclude this chapter with a discussion, how to adapt the allocation problem if one is not allowed to over-allocate goods not even by a small amount and the capacities have to be respected.

Chapter 4 studies a model of advertising markets, and analyzes whether truthful and prior-free approximation mechanisms with good approximation ratios of the maximal welfare are possible. Therefore, we first present our advertising model and introduce knapsack utility functions. Then, we analyze deterministic approximation mechanisms to solve the advertising model. Finally, we evaluate how randomized mechanisms can improve the approximation ratios and prove that there exists a truthful in expectation mechanism with a 4-approximation for this problem.

Chapter 5 concludes the thesis, summarizes the main findings and discusses open questions for further research.

# 2 Matching with Preferences

The term *matching* has been used quite differently in different disciplines. However, all these concepts descend from the graph-theoretic comprehension of a matching.

A *Graph* $G = (V, E)$ consists of a set of nodes $V$ and a set of edges $E \subseteq V \times V$. A *matching* (or a 1-factor) $M \subseteq E$ is a subset of edges, such that each node is incident to at most one edge $e \in M$. A matching is *maximal*, if for any edge $e \in E \setminus M$ the set $M \cup e$ would not be a matching anymore. That is, a maximal matching cannot be extended. A matching $M$ is *maximum* if $M$ is a maximal matching of maximal size and $M$ is called *perfect* if all nodes in $V$ are incident to an edge in $M$.

We usually consider matchings on bipartite graphs. $G$ is called *bipartite* if $V$ can be partitioned into sets $V_1$ and $V_2$, such that for every edge $(u, v) \in E$ the node $u \in V_1$ iff $v \in V_2$, i.e., there are no edges between nodes of the same partition.

A generalization of those matching problems is the introduction of weight functions on edges, $w : E \to \mathbb{R}_{\geq 0}$, and capacity functions on nodes, $q : V \to \mathbb{Z}$. Thereby, one is interested in finding a subset of edges $M$ such that $|M(v)| \leq q(v)$ for every node $v \in V$ and the sum of weights is maximized. This problem is computationally equivalent to matrix multiplication and therefore solveable in $\mathcal{O}\left(|V|^3\right)$ (Gabow, 1974).

Matching with preferences can be understood as finding such a maximal weight matching. Though, the agents (and objects) can be viewed as nodes, and the preferences as edges. However, in contrast to pure optimization the objective coefficients, i.e., the edges and weights are private information. Hence, one additionally has to consider strategic issues to ensure efficient solutions.

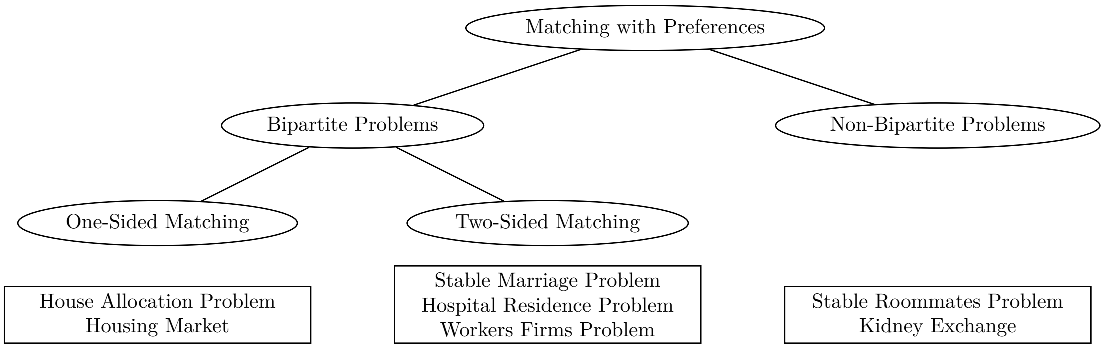


Figure 2.1: Overview of matching problems.

Figure 2.1 provides a brief overview of existing problem classes in matching theory. First, one has to differentiate if the underlying Graph is bipartite or not. That is, if we can divide the agents and objects into two sets, such that agents of one set only have preferences over agents/objects of the other set. A second structural differentiation criterion for bipartite problems is whether only one side of the market has preferences over the other side (*one-sided matching*) or both have preferences over each other (*two-sided matching*). For non-bipartite problems such a differentiation is not necessary, since all agents can have preferences over each other. Aside of these basic criteria, one can differentiate matching problems also in more detail, e.g. if the preferences have to be strict (i.e., an agent is never indifferent between two outcomes) or if the preference lists have to be complete (i.e., agents have to submit a complete ordering over all possible outcomes).

In this thesis we focus on bipartite, one-sided matching problems. However, since the matching system we build on also solves two-sided problems, we briefly introduce the two-sided matching problem too[1].

[1] We mainly follow the definition of the problems as presented in Manlove (2013). However, we use the notions and notations of our main application – the Course Allocation problem.

## 2.1 Two-Sided Matching Problems

An instance $I$ of the *Course Allocation problem* (also referred to Hospitals-Residents problem, College/University/Stable Admission/Assignment problem) consists of a set of $n$ *students* $S$ and $m$ *courses* $C$. Each course $c \in C$ has a positive integral *capacity* $q_c$. We denote with $Acc_I \subseteq S \times C$ the set of *acceptabl*e student-course pairs. Each student has a set of acceptable courses $Acc_s = \{c \in C \mid (s,c) \in Acc_I\}$ and each course a set of acceptable students $Acc_c = \{s \in S \mid (s,c) \in Acc_I\}$. Both, students and courses are *agents* in a Course Allocation problem, i.e. $A = S \cup C$. Each agent $a \in A$ *ranks* the agents in $Acc_a$ in a strict order, which we call the *preference list* $(\succ_a)$ of agent $a$.

A feasible *matching* $M$ is a subset of $Acc_I$, such that each student gets at most one course ($|M_s| \leq 1$) and the capacity of each course is respected ($|M_c| \leq q_c$). With $M_a$ we denote the set of to *a assigned* agents. If $M_a = \emptyset$ we call *a unassigned*.

An important solution concept for two-sided matching problems is stability, as emphasized by Roth and his co-authors (Roth, 1984, 1990, 1991; Roth and Xing, 1994).

**Definition 2.1: Stability.** *Given an instance $I$ of a course allocation problem and a matching $M$ in $I$, a pair $(s,c) \in Acc_I \setminus M$ is a* blocking pair *for $M$, if*

*i) $s$ is unassigned or $c \succ_s M_s$, and*

*ii) $|M_c| < q_c$ or $s \succ_c s'$ for at least one $s' \in M_c$.*

*$M$ is called* stable *if no blocking pair exists for $M$.*

The problem of finding a stable matching was first presented by Gale and Shapley (1962). They also proposed a Deferred Acceptance Algorithm (DAA, see Algorithm 2.1) that computes a stable outcome and is incentive compatible for the proposing site. The function $pop\,(Acc_a)$ returns and deletes the most preferred agent from $Acc_a$, and $tail\,(M_a)$ returns the least preferred to agent $a$ matched agent. In the DAA the students propose to their most preferred course. If more students propose to a course than places are available, the courses reject the least preferred students such that the capacities are respected. The rejected students now propose to their next choice. These steps are repeated until all students are matched, or the students cannot propose to any acceptable course anymore.

**Algorithm 2.1:** (student proposing) Deferred Acceptance Algorithm (Gale and Shapley, 1962).

**Input :** instance $I$ of a (two-sided) Course Allocation problem.
$M = \emptyset$
**while** $\exists s \in S : M_s = \emptyset$ *and* $Acc_s \neq \emptyset$ **do**
  $c = pop\,(Acc_s)$
  **if** $|M_c| = q_c$ **then**
    $s' = tail\,(M_c)$
    **if** $s \succ_c s'$ **then** $M_c = (M_c \cup s) \setminus s'$, $M_s = c$ and $M_{s'} = \emptyset$
  **else** $M_c = M_c \cup s$ and $M_s = c$
**Output:** stable matching $M$ for $I$

Algorithm 2.1 computes a stable matching that is the best possible stable matching for the students and the worst possible for the courses. That means, the courses prefer every other stable matching over the outcome of the student proposing DAA. Analogues one can define the course proposing DAA, which is optimal for courses and returns the worst possible stable matching for the students (Gale and Shapley, 1962; Gusfield and Irving, 1989).

Even if Algorithm 2.1 is strategy proof for the proposing side, there exists no mechanism for the stable matching problem for which truthful reporting is a dominant strategy for all participating agents (Roth, 1982).

The course optimal and the student optimal stable matchings are two extreme cases. In general there may be exponentially many other matchings *in between.* However, regardless of the differences of these matchings, some structural properties are the same among all of them.

**Theorem 2.2: Rural Hospitals Theorem (Roth, 1984; Gale and Sotomayor, 1985; Roth, 1986).** *Given an instance of the Course Allocation problem, the following properties hold:*

*i) the same students are assigned in all stable matchings;*

*ii) each course gets assigned the same number of students in all stable matchings;*

*iii) any course that has free capacity in one stable matching is assigned exactly the same set of students in all stable matchings.*

For their seminal work on *the theory of stable allocations and the practice of market design* E. Roth and Lloyd S. Shapley were awarded with the Nobel Memorial Price in Economic Science in 2012 (Economic Sciences Prize Committee of the Swedish Academy of Sciences, 2012).

## 2.2 One-Sided Matching Problems

An instance of a one-sided matching problem is similar to the two-sided version. The only structural difference is that the members of the second partition, in our application the courses $C$, are (only) objects and have no preferences over the agents of the first partition, the students $S$. That is, $A = S$. The definition of a (feasible) matching remains unchanged. This problem is also known as (Capacitated) House Allocation problem, first introduced by Shapley and Scarf (1974).

Since the courses have no preferences, stability reduces to *envy-freeness* in one-sided matching problems. A matching is called *envy-free* if no agent prefers the outcome of an other agent over his own assignment. However, a (meaningful) deterministic assignment that is envy-free does not necessarily had to exist. Consider a simple example where we have a rock and a diamond, and two agents, each preferring the diamond over the rock. Equally who gets assigned the diamond, the agent getting the rock always will envy the other agent[2].

Therefore, (Pareto) efficiency became an important design desideratum for one-sided matching. We formally will introduce Pareto efficiency, envy-freeness and further design desiderata for one-sided matching problems in Section 3.2.2.

[2]The only envy-free assignment in this example is the empty matching.

Like for two-sided matching problems, we are interested in efficient mechanisms that incentivize the students to submit their preferences truthfully. However, similar to the two-sided problem, the design space for those mechanisms is limited. Gibbard (1973) and Satterthwaite (1975) showed that with general ordinal preferences every deterministic mechanism is susceptible to manipulation. Later, Gibbard (1977) has shown that every strategy-proof mechanism is a lottery over deterministic mechanisms that are dictatorial. Hence, we are limited to a form of Random Serial Dictatorship (RSD, see Algorithm 2.2).

**Algorithm 2.2:** Random Serial Dictatorship

```
Input  : instance I of a (one-sided) Course Allocation problem
order S randomly, M = ∅
forall i ∈ S do
    while Acc_s ≠ ∅ do
        c = pop(Acc_s)
        if |M_c| < q_c then
            M_c = M_c ∪ s
            M_s = c
            break
Output: deterministic matching M
```

Algorithm 2.2 orders the students randomly and assigns each student to his most preferred course that is still available. RSD is proven to be (ex post) efficient, symmetric and strategy-proof. We discuss this mechanism together with further mechanisms, e.g. Probabilistic Serial (Bogomolnaia and Moulin, 2001) and the Competetive Equilibrium from Equal Incomes (Hylland and Zeckhauser, 1979) in more detail in Section 3.2.3.

## 2.3 Matching at TUM

Since 2014 every term more than 1500 students at the Computer Science Department of the TUM get assigned places in practical courses and seminars with a student optimal DAA. Before introducing a matching system, those course seats were allocated with the First-Come First-Served (FCFS) mechanism. In this system students usually had to register for multiple courses and after the decision of the course organizers whom to accept and whom to reject, they withdrew their application from the less preferred assigned courses. Hence, course organizer had places left, which they could offer other students who were more interested in this course. Altogether this system was inefficient, not strategy-proof and unfair.

| Semester | Practical Courses | Seminars | Pro-Seminars |
|---|---|---|---|
| SS 17 | 796 | 907 | 330 |
| WS17/18 | 825 | 672 | 131 |
| SS18 | 962 | 1248 | 68 |
| WS18/19 | 1040 | 908 | - |

Table 2.1: Number of participating students in the matching for seminars and practical courses.

Now, the mechanism is centralized. The students have a one week period to express their ordinal preferences over different seminars via drag-and-drop in our matching system (see Figure 2.3). Afterwards, the course organizer see, who is interested in their course[3] and can rank the students too. After this period, a student optimal stable matching is computed using the DAA (see Algorithm 2.1).

Table 2.2 reports some metrics of the instances in winter term 2018. E.g., there were 908 students each interested in one of 967 course seats for seminars. Since students as well as course organizer were allowed to rank participants as unacceptable, 116 students were not assigned, although there were still 175 places left. This can easily be explained with students, who are only interested in very specific courses. Furthermore some courses were quite unpopular and got less applicants than available places (e.g. courses for theoretical computer science). However, it was possible to assign 525 students to their top choice and more than 700 students got a place in one of their three most preferred seminars. Diebold et al. (2014) and Diebold and Bichler (2017) provide a more detailed analysis of the matching of students to seminars and practical courses at TUM.

[3]However, the course organizer cannot see how the students ranked the course.

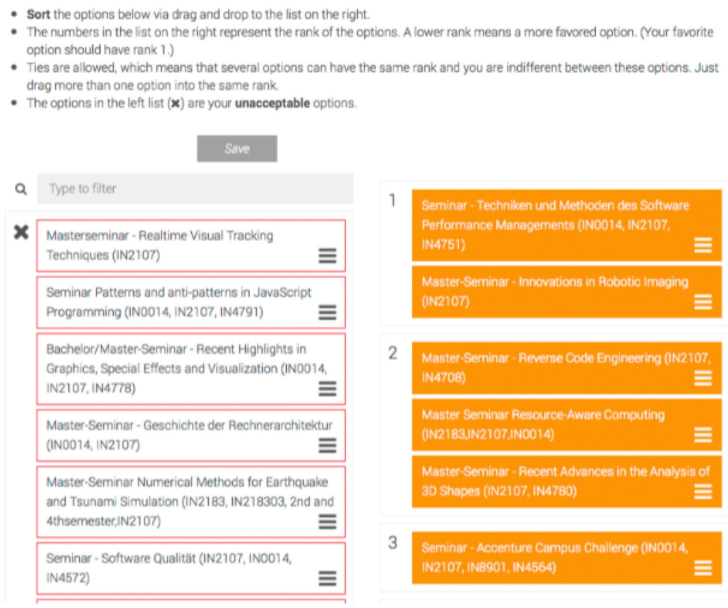


Figure 2.2: Ranking of seminars via drag and drop.

| Instance | students | not assigned | capacity | places left | Rank 1 | Rank top 3 |
|---|---|---|---|---|---|---|
| Seminars | 908 | 116 | 967 | 175 | 525 | 706 |
| Practical Courses | 1040 | 190 | 921 | 71 | 603 | 785 |

Table 2.2: Number of participating students, number of students that were not assigned in the first round, the total capacity of the instances, the places that were not assigned and some rank statistics for the instances in winter term 2018.

Because of a constantly rising number of students over the last years, a second application became more and more important - the assignment of tutor groups. For some of the large lectures in the first semesters of the computer science studies, there are more than 50 different tutorial groups available. Each student is interested in at most one of them. This problem can be seen as a one-sided matching problem and we use RSD (see Algorithm 2.2) to compute an assignment. Again the students can rank the different tutorials via drag and drop before we compute the allocation. In contrast to the matching of seminars and practical courses, the participation on this matching is not mandatory for the courses. However, the last years we observed an increasing demand for those instances.

| Course | IN0002 | IN0003 | IN0006 | IN0007 | IN0008 | IN0009 | IN0022 | IN2028 | MA0901/02 |
|---|---|---|---|---|---|---|---|---|---|
| Participants | 1700 | 900 | 1200 | 1000 | 1500 | 900 | 300 | 400 | 800 |

Table 2.3: Average number of participating students for the matching of tutorials in different courses.

Even if the usage of the matching system improves the quality of the assignments and circumvents the flaws of FCFS, one problem remains. Usually students have to visit up to four of those huge lectures in each term. Since each lecture has its own matching instance, the students have to submit preferences to up to four different instances, which all can have different deadlines and match the students to tutorials independent form each other. Now, the students have to decide how to rank the courses in each instance. If they rank all courses truthfully in each instance, it can happen that they get assigned overlapping tutorials, which is not feasible for them. However, if they are aware of this problem and they adjust their ranking, the outcome can be less efficient. Hence, one needs a system where students can report preferences not only over single courses, but over whole schedules.

# 3 Matching with Complementarities

## About Preference Elicitation, Fairness, and Truthfulness [4]

Course assignment is arguably one of the most wide-spread assignment problems where money cannot be used to allocate scarce resources. Such problems appear at most educational institutions. Matching with preferences has received significant attention in the recent years. While simple First-Come First-Served (FCFS) rules are still wide-spread, many organizations adopted matching mechanisms such as the deferred acceptance algorithm (Gale and Shapley, 1962; Diebold et al., 2014) or course bidding (Sönmez and Ünver, 2010; Krishna and Ünver, 2008) to allocate scarce course seats. Although many course assignment problems are similar to the widely studied school choice problems with students having private preferences for one out of many courses, other applications differ significantly. In particular, students are often interested in schedules of courses across the week. Assigning schedules of courses has been referred to as the *combinatorial assignment problem* (CAP) (Budish, 2011). Similar problems arise when siblings should be assigned to the same schools in school choice (Abdulkadiroğlu et al., 2006), or couples in the context of the hospital residency matching (Ashlagi et al., 2014).

[4]This chapter is mainly based on Bichler et al. (2018) and Merting et al. (2016). Especially the introduction, the presentation of the matching system and the discussion of the field experiments are mainly identical to the respective parts of Bichler et al. (2018). For the section about the Combinatorial Assignment Problems the respective section from Bichler et al. (2018) is extended by revised results of Merting et al. (2016) and additional proofs. Bichler et al. (2018) is still a working paper. However, a shortend version is already available (Bichler et al., 2019).

The need to assign course schedules rather than courses individually became apparent in an application of matching with preferences at the Technical University of Munich that we will discuss. The Department of Informatics is using the deferred acceptance algorithm for two-sided matching problems and random serial dictatorship for one-sided matching problems. These algorithms are used to assign seminars or practical courses, and every semester about 1500 students are being matched centrally. For seminars and practical courses students need to get assigned one out of many courses offered per semester.

In the initial three semesters the situation is different. There are large courses with hundreds of students (e.g. on linear algebra or algorithms). These courses include a lecture and small tutor groups. Students need to attend one tutor group for three to four courses in each semester. These tutor groups should not overlap and they should be adjacent to each other such that students do not have a long commute for each of the tutor groups individually. For example, students might want to have two tutorials in the morning and one after lunch on a particular day to reduce their commute time, and they would have a strong preference for this schedule over one where the tutorials are scattered across the week. In any case, students have timely preferences over course schedules, which need to be considered. This makes it a combinatorial assignment problem.

A first and seminal approach to address this challenging problem, the *approximate competitive equilibrium from equal incomes mechanism* (A-CEEI), was published by Budish (2011). Budish et al. (2017) reports the empirical results at the Wharton School of Business. In addition, Budish and Kessler (2017) summarize the results of lab experiments.

The work was breaking new ground, but the A-CEEI mechanism is also challenging. First, it is not guaranteed that a solution exists that satisfies all capacity constraints. Second, the problem of computing the allocation in A-CEEI is PPAD-complete and the algorithms proposed might not scale to larger problem sizes required in the field (Othman et al., 2016). Third, students might not be able to rank-order an exponential set of bundles, which is a well-known problem (aka. missing bids problem) in the literature on combinatorial auctions (with money) (Milgrom, 2010; Bichler et al., 2011, 2014). The latter is a general problem in CAP not restricted to A-CEEI, which we will discuss in much more detail in Section 3.3.

Randomization can be a powerful tool in the design of algorithms, but also in the design of economic mechanisms. Nguyen et al. (2016) recently provided two randomized mechanisms for one-sided matching problems, one with cardinal and one with ordinal preferences for bundles of objects. The mechanism for ordinal preferences is a generalization of Probabilistic Serial (Bogomolnaia and Moulin, 2001), called Bundled Probabilistic Serial (BPS). Nguyen et al. (2016) show that this randomized mechanism is ordinally efficient, envy-free, and weakly strategy-proof. These appealing properties come at the expense of feasibility, but the constraint violations are limited by the size of the bundles. In course assignment problems the size of the bundles is typically small (e.g., bundles with three to four tutor groups) compared to the capacity of the courses or tutor groups (around 30 seats or more). Computationally, the mechanism is still very fast, which is important for large instances of the course allocation problem that can frequently be found. This makes BPS a practical approach to many problems that appear in practice.

## 3.1 Contributions

We report on a first large-scale field study of BPS and address important problems in the implementation of mechanisms for the combinatorial assignment problem that are beyond a purely theoretical treatment. In particular, preference elicitation is a central concern in combinatorial mechanisms with a fully expressive bid language and we provide a practical approach that addresses the combinatorial explosion of possible bundles for many applications in Section 3.3. Theoretical contributions of assignment mechanisms largely focus on envy-freeness and efficiency as primary design desiderata. In Section 3.4 we report properties of matchings such as their size, their average rank, the probability of matching, the profile, and the popularity. These properties are of central importance for the choice of mechanisms. It is important to understand the trade-offs with other mechanisms, in particular with the wide-spread FCFS.

Implementing and testing new IS artifacts for coordination in organizations is challenging and we are grateful for the possibility to run a large-scale field experiment at the Department of Informatics of the Technical University of Munich (TUM). This is particularly true for a non-trivial mechanism such as BPS, which involves advanced optimization and randomization. Yet, we can report on the assignment of 1415 students in the summer term 2017 to 67 tutor groups for 4 classes, the assignment of 1736 students in the winter term 2017/2018 to 66 tutor groups for 4 classes, and the assignment of 1683 students in the winter term 2018/2019 to 68 tutor groups for 4 classes using BPS.

For such a large application we could not elicit preferences of students for BPS and let them participate in FCFS simultaneously. Instead we simulated FCFS via a version of Random Serial Dictatorship that allows for bundles (BRSD), which is of independent interest as an assignment mechanism. In our numerical experiments we simulated FCFS via a large number of random order arrivals in BRSD using the preferences elicited in BPS and average across all of them. This approach allows for a comparison between BPS and BRSD (FCFS) on equal footing.

FCFS only collects limited information about the preferences of participants, a single bundle only. Mechanisms for the combinatorial assignment problem allow participants to specify preferences for all possible bundles. However, a fully enumerative bid language requires participants to submit preferences for an exponential set of bundles which is impractical.

Preference elicitation and user interface design have long been a topic in IS research (Santos and Bariff, 1988; Lee and Benbasat, 2011). We contribute an approach that is applicable in a wide array of CAP applications where timely preferences matter. We elicit a small number of parameters about breaks and preferred times and days of the week. Together with some prior knowledge about student preferences this allows us to score and rank-order all possible bundles. Students could iteratively adapt the parameters and the ranking, which then served as an input for BPS. While such ranking algorithms will differ among types of applications, adequate decision support that aids the ranking of exponentially many bundles is a crucial prerequisite to actually achieve the benefits of combinatorial assignment in real-world applications.

In our empirical analysis, we show that BPS has many advantages over BRSD in all of the properties introduced earlier. While the differences in these criteria are small, envy-freeness turns out to be the most compelling advantage of BPS. The level of envy that we find in BRSD is substantial in spite of the limited complementarities in student preferences, who are only interested in bundles with at most four tutor groups. This has to be traded off with the simplicity of FCFS. Overall, we *empirically test and illustrate theory* that has been developed only recently.

## 3.2 Combinatorial Assignment Problems

Let us first define the combinatorial assignment problem in the context of course assignment applications. We introduce desirable properties for deterministic and random assignments, and analyze (randomized) mechanisms.

### 3.2.1 Assignment Problems

Assigning objects to agents with preferences but without money is a fundamental problem referred to as *assignment problem with preferences* or *one-sided matching with preferences*. In the following, we use the terms assignment and matching interchangeably. In course assignment, students express ordinal preferences, which need to be considered in the assignment. A *one-sided one-to-many course assignment problem* consists of a finite set of $n$ students (or agents) $S$ and a finite set of $m$ courses (or objects) $C$ with the *maximum capacities* $q = (q_1, q_2, \ldots, q_m)$.

In the *combinatorial assignment problem* in the context of course allocation, every student $i \in S$ has a transitive preference relation $\succeq_i \in \mathcal{P}$ over subsets (or bundles) $b \in B$ of elements of $C$. Let $Acc_i \subseteq B$ be the set of *acceptable bundles* of student $i$ and $Acc_S = \bigcup_{i \in S} Acc_i$ the overall set of all acceptable bundles. For all bundles $b \notin Acc_i$ we call $b$ *unacceptable* and note $b =_i \emptyset$. A preference profile $\succeq = (\succeq_1, \ldots, \succeq_n) \in \mathcal{P}^{|S|}$ is an $n$-tuple of preference relations. For most of the thesis we assume strict preferences, but we also discuss indifferences in the conclusions.

**Definition 3.1: Deterministic Matching.** *A* deterministic combinatorial assignment *(deterministic matching) is a mapping $M \subseteq S \times B$ of students $S$ to bundles $B$ of courses $C$ with:*

*i)* $\forall i \in S : M_i \in Acc_i \cup \emptyset$,

*ii)* $\forall c \in C : M_c \subseteq S$,

*iii)* $\forall i \in S \wedge c \in C : c \in M_i \Leftrightarrow i \in M_c$.

*$M$ is* feasible *if $|M_c| \leq q_c$ for all $c \in C$. $\mathcal{M}$ describes the set of all deterministic matchings.*

We can model feasible assignments also as integer programs (IP). Thereby, bundles are described with binary vectors $b \in \{0, 1\}^m$, where $b_j = 1$ if course $j$ is included in bundle $b$. We define the size of $b$ with $size(b) = \sum_{j=1}^{m} b_j$, the number of different courses included in the bundle. Let $x_{ib}$ be a binary variable describing if bundle $b$ is assigned to student $i$. Then we can model the demand and supply as linear constraints. The supply constraints make sure that the capacity of the courses are not exceeded, and the demand constraints determine that each student can get at most one bundle.

$$\sum_{i \in S, b \in B} x_{ib} b_j \leq q_j \qquad \forall j \in C \qquad \text{(supply)}$$

$$\sum_{b \in B} x_{ib} \leq 1 \qquad \forall i \in S \qquad \text{(demand)}$$

$$x_{ib} \in \{0, 1\} \qquad \forall i \in S, b \in B \qquad \text{(binary)}$$

That is, a deterministic matching is feasible if it corresponds to a feasible integer solution to the constraints (demand) and (supply). *Random combinatorial assignments* (random matchings) are related to fractional assignments with $0 \leq x_{ib} \leq 1$. The fractional solution $x_{ib}$ to the (demand) and (supply) constraints is then equal to the probability that student $i$ obtains bundle $b$. While random matchings can be seen as probability distributions over bundles, *lotteries* are probability distributions over whole matchings.

**Definition 3.2: Lottery.** *A* Lottery *$L$ is a probability distribution over feasible deterministic matchings. The set of all lotteries is denoted as $\mathcal{L}$.*

Nguyen et al. (2016) show that a lottery of bundles induces probability distributions over these bundles that satisfy the constraints (demand) and (supply). Thus a lottery coincides with a random matching. However, a random matching does not need to be implementable into a lottery over feasible deterministic assignments in general if the bundle size is greater than one.

For assignment problems with single-unit demands ($size(b) = 1$) the Birkhoff-von-Neumann theorem (Birkhoff, 1946; Von Neumann, 1953) says that every fractional allocation can be written as a unique probability distribution over feasible deterministic assignments. That is, any random assignment can be implemented as a lottery over feasible deterministic assignments, such that the expected outcome of this lottery equals the random assignment. One can describe a random assignment as a bistochastic matrix $P$, where $p_{ic}$ is the probability that student $i$ is assigned to course $c$. The Birkhoff-von-Neumann theorem shows that such a bistochastic matrix can be decomposed into a convex combination of permutation matrices, which describe feasible deterministic assignments. However, the Birkhoff-von-Neumann theorem fails when bundles $b$ with $size(b) > 1$ need to be assigned. Nguyen et al. (2016) generalize the Birkhoff-von-Neumann theorem and show that any fractional solution respecting the (demand) and (supply) constraints can be implemented as a lottery over integral allocations that violate the (supply) constraints only by at most $k - 1$ course seats, where $k = max\,\{size(b) \mid b \in Acc_S\}$, i.e. the maximal size of any acceptable bundle.

Courses in our application are actually tutor groups and each tutor group belongs to one of $\ell$ classes. Students in our application can only select bundles with at most one tutor group in each of these classes. For example, a student might select a bundle with a course seat in a tutor group for mathematics on Monday at 1 pm, and another tutor group in software engineering two hours later, but no additional tutor group in mathematics or software engineering in this bundle. As a result, the possible size of a bundle $b$ is $size(b) \leq \ell \ll m$.

### 3.2.2 Design Desiderata

Efficiency, envy-freeness, and strategy-proofness are design desiderata of first-order importance typically considered in the theoretical literature on assignment problems. In this section we introduce these and further concepts for deterministic and random matchings.

#### Deterministic Matchings

We start with deterministic assignments. An important property for deterministic matchings is Pareto efficiency.

**Definition 3.3: Pareto Efficiency.** *Given two matchings $M', M \in \mathcal{M}$, $M'$* dominates *$M$ if*

*i)* $\forall\, i \in S : M'_i \succeq_i M_i$, *and*

*ii)* $\exists\, i \in S : M'_i \succ_i M_i$.

*We denote a matching $M \in \mathcal{M}$ as* Pareto optimal *or* Pareto efficient *if there exists no matching $M' \in \mathcal{M}$ that dominates $M$.*

Pareto efficiency means that no student can be better off without making another student worse off. This is not only an optimality criterion, but also a necessary fairness condition. Without Pareto efficiency the agents would have incentives to subvert the matching to receive a more preferred assignment.

One may also be interested in matchings with maximal size, i.e., matchings that assign as many students as possible. We call a matching $M$ *maximum* if it is Pareto efficient and $|M| \geq |M'|$ for all Pareto efficient matchings $M' \in \mathcal{M}$.

We present a characterisation of Pareto efficiency using conflict trees similar to the result in Sng (2008) for the single-unit demand case. Sng (2008) shows that a matching is Pareto efficient iff it is maximal, trade-in-free and cyclic-coalition-free.

A Matching $M \in \mathcal{M}$ is *maximal* if there is no unassigned student $i \in S$ with $Acc_i \neq \emptyset$ such that there exists a bundle $b \in Acc_i$ with $|M_c| < q_c$ for all $c \in b$. $M$ is called *trade-in-free* if there is no student $i \in S$ with $M_i \neq \emptyset$ such that there exists a bundle $b \in Acc_i$ with $b \succ_i M_i$ and $|M_c^{-i}| < q_c$ for all $c \in b$. Thereby, $M^{-I}$ denotes the resulting matching if we exclude the assignment of all agents $i \in I \subseteq S$ from $M$ (i.e., $M^{-I} = M \setminus \{(i, M_i) \mid i \in I\}$)[5].

**Definition 3.4: Cyclic Coalition (Sng, 2008).** *A* cyclic coalition *for a matching $M$ is a sequence of students $I = (i_0, i_1, \ldots, i_{r-1})$ for $r \geq 2$ with $M_{i_{(j+1)\ mod\ r}} \succeq_{i_j} M_{i_j}$ for each $i_j \in I$.*

That is, each student in $I$ prefers the assignment of his successor in $I$. Note that in this setting $|M_i| \leq 1$ for all students $i$. We can rewrite the definition of a cyclic coalition as follows: A sequence of students $I$ is a cyclic coalition for a matching $M$ if $M' = M^{-I_{l+1}} \cup \left\{ \left(i_j, M_{i_{(j+1)\ mod\ r}}\right) \mid j \leq l \right\}$ with $I_{l+1} = \{i_{j\ mod\ r} \mid j \leq l+1\}$ is feasible for $l = 0, \ldots, r-1$ and $M'$ dominates $M$. That means, a cyclic coalition is a sequence of agents such that we can construct a matching $M'$ Pareto dominating $M$ by iteratively deleting the current assignment of $i_j$ and his successor ($i_{j+1}$), and assigning another object to $i_j$ such that the new object is at least as good as the old one (in this special case the former matching of $i_{j+1}$).

We generalize this idea to bundles. Therefor, we have to define trees and depth of nodes. A *tree* $T = (V, E)$ is a connected, directed graph with no cycles. Every tree has a unique node $r \in V$ with no incoming edge, called the *root*. The *depth* $d_T(v)$ of a node $v \in V$ in $T$ is the number of edges, i.e. the length of the path, between

[5] If $I = \{i\}$ we note $M^{-i}$ instead of $M^{-\{i\}}$ for brevity.

$r$ and $v$. If $v$ is not in the tree, $d_T(v) = \infty$. Let $M[S']$ describe the matching $M$ induced by the agents $i \in S' \subseteq S$ and $S^i = \{j \in S \mid d_T(v_j) \leq d_T(v_i)\}$ be the set of agents, whose nodes are at most as deep as $v_i$ in the tree $T$.

**Definition 3.5: Conflict Tree.** *Given a set of nodes $V$, a set of edges $E \subseteq V \times V$,and two feasible matchings $M, N \in \mathcal{M}$, $T = (V, E, M, N)$ describes a* conflict tree *if:*

*i)* $(V, E)$ *is a tree with* $v_i \in V$ *representing agent* $i \in S$*,*

*ii)* $\forall\ (v_i, v_j) \in E$*:* $N[S^i] \cup M[S \setminus S^i]$ *is infeasible, and*

*iii)* $\forall\, v_i \in V$*:* $N[S^i] \cup M^{-I_j}[S \setminus S^i]$*, with* $I_j = \{j \in S \mid (v_i, v_j) \in E\}$ *is feasible.*

*We say, the agents $j \in I_j$ are* in conflict *with agent $i$.*

The second condition says that there exist students $i, j \in S$, for which $N_i \cap M_j$ includes at least one course $c$ with $|N_c[S^i]| + |M_c[S \setminus S^i]| > q_c$. That is, if we still assign $M_j$ to agent $j$, we would over-allocate $c$. We therefore have to assign an other bundle to $j$. Algorithm 3.1 generates such a conflict tree. The second for-loop ensures that edges are only generated if the conditions two and three are violated, and that no edges to a node $i'$ are generated that already is in conflict with an other agent.

Informally, Algorithm 3.1 generates a conflict tree from a matching by iteratively deleting the assignment of a student and assigning another bundle to him. If the new matching becomes infeasible, we delete the assignments of conflicting students. That is, students who are not in the conflict tree yet and whose currently assigned bundles include over-allocated courses. In the next iteration we reassign these students to other bundles respecting the capacities of the courses. We repeat these steps until there are no further conflicts. Here, the node set $V$ represents the students who might form a coalition, just like the *sequence of students* in a cyclic coalition. The edge set $E$ illustrates the dependencies and the conflict potential of these students in the current matching. $N$ proposes an alternative to $M$ where all students in the tree are reassigned. However, in a conflict tree $N$ does not have to dominate $M$. With help of the conflict tree we can define a tree coalition.

**Algorithm 3.1:** Construction of a conflict tree.

```
Input  : preferences (⪰_i)_{i∈S}, matching M ∈ 𝓜
V_0 = V = i for arbitrary i ∈ S; V_j = ∅ for j > 0
E = ∅; k = 0; N = M^{-i}
while V_k ≠ ∅ do
    forall i ∈ V_k do
        N = N ∪ (i, b) for b ∈ Acc_i ∪ ∅, such that N[V] feasible
        forall i' ∈ S \ (V ∪ V_{k+1}) do
            if ∃c ∈ N_{i'} : |N_c| > q_c then
                V_{k+1} = V_{k+1} ∪ i'
                E = E ∪ (i, i')
                N = N^{-i'}
    V = V ∪ V_{k+1}; k++
Output: nodes V, edges E, matching N
```

**Definition 3.6: Tree Coalition.** *We call* $T = (V, E, M, N)$ *a* tree coalition *for a matching* $M \in \mathcal{M}$ *if* $T$ *is a conflict tree and* $N[S_V]$ *dominates* $M[S_V]$*, with* $S_V = \{i \in S \mid v_i \in V\}$*. A matching* $M \in \mathcal{M}$ *is* tree-coalition-free *if there does not exist any tree coalition for* $M$*.*

Note that for checking this definition one needs to generate all possible conflict trees whereof in general exponentially many exist. Let us first provide an example to give some intuition for the definitions of conflict trees and tree-coalitions.

**Example 3.7: Conflict Trees & Tree Coalitions.** *Suppose there are* 4 *students* $\{1, \dots, 4\}$ *and* 4 *courses* $\{A, \dots, D\}$ *with the following capacity vector:* $(2, 1, 2, 2)$*. The students have the following preferences; all other bundles are unacceptable for the respective students:*

$$1 : \{A, D\} \succ_1 \{C, D\} \succ_1 \{D\} \qquad 2 : \{B, D\} \succ_2 \{C, D\} \succ_2 \{B\}$$
$$3 : \{A, D\} \succ_3 \{A\} \succ_3 \{C\} \qquad 4 : \{A, C, D\} \succ_4 \{A, C\} \succ_4 \{B, D\}$$

*This example contains two parts to illustrate the differences between conflict trees and tree-coalitions.*

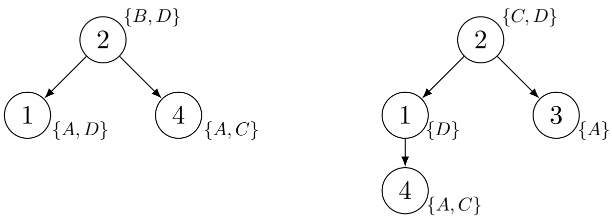


Figure 3.1: Tree coalition (left) & conflict tree (right) for Example 3.7

*i) Consider the matching* $M = \{(1, \{C, D\}), (2, \emptyset), (3, \{C\}), (4, \{B, D\})\}$. *We construct a conflict tree for* $M$ *with initial student* 2 *using Algorithm 3.1 (see left tree in Figure 3.1). That is,*

$$V_0 = \{2\} \text{ and } N = M^{-2} = \{(1, \{C, D\}), (3, \{C\}), (4, \{B, D\})\}.$$

*Now, we choose a new bundle* $\{B, D\}$ *for student* 2 *and receive*

$$N = \{(1, \{C, D\}), (2, \{B, D\}), (3, \{C\}), (4, \{B, D\})\}.$$

*Obviously, the matching induced by student 2,* $N[2] = \{(2, \{B, D\})\}$, *is feasible. However, the whole matching* $N$ *over-allocates the courses* $B$ *and* $D$. *We first consider student* 1 *and delete her assignment. Now,* $B$ *is still over-allocated and we delete the assignment of student* 4. *Hence, student* 2 *is in conflict with the students* 1 *and* 4, *and we therefore add the edges* $(2, 1)$ *and* $(2, 4)$ *to* $E$. *We obtain* $N = N^{-\{1,4\}} = \{(2, \{B, D\}), (3, \{C\})\}$ *and* $V_1 = \{1, 4\}$. *In the next iteration we assign new bundles* $\{A, D\}$ *and* $\{A, C\}$ *to the agents in* $V_1$. *After this iteration the algorithm stops with the left conflict tree shown in Figure 3.1 and* $N = \{(1, \{A, D\}), (2, \{B, D\}), (3, \{C\}), (4, \{A, C\})\}$. *Since* $N_1 \succ_1 M_1$, $N_2 \succ_2 M_2$, *and* $N_4 \succ_4 M_4$, *this conflict tree is also a tree coalition. Additionally we see that a conflict tree not necessarily has to contain all students, i.e.* $|V|$ *might be smaller than* $|S|$.[6]

[6] If one considers in the first iteration agent 4 first, the resulting $N$ would already be feasible and the conflict tree would change to a path $2 \to 4 \to 1$.

*ii) Consider a second matching*

$$M' = \{(1, \{A, D\}), (2, \{B\}), (3, \{C\}), (4, \{A, C, D\})\}.$$

*Student 2 is only assigned to his third preference in $M'$. Hence, he might be interested in getting assigned an other bundle. We can construct the right conflict tree shown in Figure 3.1 by starting with $V_0 = \{2\}$, deleting $(2, \{B\})$ and assigning $\{C, D\}$ to student 2. We receive*

$$N' = M'^{-2} \cup (2, \{C, D\}) = \{(1, \{A, D\}), (2, \{C, D\}), (3, \{C\}), (4, \{A, C, D\})\}.$$

*Courses C and D are now over-allocated. After deleting the assignments of student 1 and 3 the capacities of both courses are respected again, therefore student 2 is in conflict with the students 1 and 3 and we add edges to E respectively. Student 4 is not considered in this iteration, since $N'^{-\{1,3\}}$ is already feasible. Therefore $V_1 = \{1, 3\}$. In the next iteration we add $(1, \{D\})$ and $(3, \{A\})$ to $N'^{-\{1,3\}}$. The resulting matching induced by the already considered students $N'[\{1, 2, 3\}] = \{(1, \{D\}), (2, \{C, D\}), (3, \{A\})\}$ is feasible. However, in $N'$ D is over-allocated and student 4 is in conflict with 1, i.e. $V_2 = \{4\}$. We reassign 4 to $\{A, C\}$ and receive a feasible matching $N' = \{(1, \{D\}), (2, \{C, D\}), (3, \{A\}), (4, \{A, C\})\}$. Since $M'_4 \succ_4 N'_4$, this conflict tree cannot be a tree coalition.*

**Corollary 3.8:** *A matching is maximal and trade-in-free if it is tree-coalition-free.*

*Proof.* Consider a matching that is not maximal, i.e., there has to be at least one unassigned student $i$ who can be assigned to an acceptable bundle $b$. Hence, we can define a conflict tree $T = (\{i\}, \emptyset, M, N = M^{-i} \cup (i, b))$. As $N_i = b \succ_i \emptyset = M_i$, T is a tree coalition. If the matching $M$ is not trade-in-free, we can find an agent $i$ with $b \succ_i M_i \neq \emptyset$ for an acceptable bundle $b \in Acc_i$ such that $N = M^{-i} \cup (i, b)$ is feasible. Again, we can define a tree coalition $T = (\{i\}, \emptyset, M, N)$. □

In a tree coalition $T = (V, E, M, N)$ we only require that $N$ dominates $M$ for the agents in $V$. However, it might be the case that $N$, considered for all agents, does not dominate $M$. With the following corollary we see that we also can find a matching $N'$ that dominates $M$, by combining $M[S \setminus S_V]$ and $N[S_V]$.

**Corollary 3.9:** *If there exists any tree coalition $T = (V, E, M, N)$ for $M$, there also exists a tree coalition $T' = (V, E, M, N')$ where $N'$ dominates $M$.*

*Proof.* If $N$ already dominates $M$, $T' = T$ and the claim is true. Assume $N$ does not dominate $M$. Since $T$ is a conflict tree, $N[S^i] \cup M^{-I_j}[S \setminus S^i]$ is feasible for every $i \in S$ with $v_i \in V$ and $I_j = \{j \in S \mid (v_i, v_j) \in E\}$. Particularly, this condition has to be fulfilled for the agents represented by leafs in $T$ too. Since those nodes do not have any outgoing edge, $I_j = \emptyset$ for these nodes. Let $v_{i^*} = argmax\{d_T(v_i) \mid v_i \in V\}$ the deepest node in $T$. For $i^*$ $S^{i^*} = \{j \in S \mid d_T(v_j) \leq d_T(v_{i^*})\} = S_V$ and therefore $N' = N[S_V] \cup M[S \setminus S_V]$ a feasible matching. Hence, $N'[S \setminus S_V] = M[S \setminus S_V]$ and $N'[S_V] = N[S_V]$ dominates $M[S_V]$. Therefore $T' = (V, E, M, N')$ is a tree coalition where $N'$ dominates $M$. □

With Corollary 3.9 we immediately see that a matching $M$ cannot be Pareto efficient if there exists any tree coalition for $M$. Theorem 3.10 shows that both directions of this statement are true.

**Theorem 3.10:** *A matching $M \in \mathcal{M}$ is Pareto efficient iff it is tree-coalition-free.*

*Proof.* Suppose there exists a tree coalition $T = (V, E, M, N)$ for $M$. Because of Corollary 3.9 we can find a tree coalition $T = (V, E, M, N')$, where $N'$ dominates $M$. That is, $M$ is not Pareto efficient.

Suppose $M$ is not Pareto efficient. That is, it exists an agent $i^* \in S$ and a matching $M' \in \mathcal{M}$ with $M'_{i^*} \succ_{i^*} M_{i^*}$ and $M'_i \succeq_i M_i$ for all $i \in S$. We compare $M$ and $M'$ for $i^*$. Let us consider $N = M^{-i^*} \cup (i^*, M'_{i^*})$, $V = \{v_{i^*}\}$, and $E = \emptyset$.
*Case 1:* If $N$ is feasible, we can define a conflict tree $T = (V, E, M, N)$ for $M$, where $N[S_V]$ dominates $M[S_V]$, i.e., $M$ cannot be tree-coaltion-free.
*Case 2:* Let $N$ be infeasible. As we constructed $N$ from $M$ by changing the assignments of selected agents, and $M$ is feasible, there must be a non empty set

of agents $I \subseteq S \setminus S_V$ and an agent $i' \in S_V$ with $|N_c| > q_c$ for at least one object $c \in (\bigcup_{i \in I} N_i) \cap N_{i'}$. Since $M'_i \succeq_i M_i$ for all $i \in S$ and $M'$ feasible, one can define a feasible matching $N' = N^{-I}[S_V] \cup \{(i, M'_i) \mid i \in I\}$, where all agents in $S_V$ weakly prefer the assignment $N'$ over $M$. We set $N = N'$, $V = V \cup \{v_i \mid i \in I\}$, and $E = E \cup \{(i', i) \mid i \in I\}$, and repeat the argumentation in case 1 and 2.
Since every time $N$ is infeasible we add at least one agent to $S_V$, there has to be a round, where $N$ is feasible (at the latest when $N = M'$) and we can define the respective tree coalition. That is, if $M$ is not Pareto efficient, there exists a tree coalition for $M$. □

Some design desiderata are not defined for the matchings but for the mechanisms calculating them. A *deterministic assignment mechanism* is a function $\chi : \mathcal{P}^{|S|} \rightarrow \mathcal{M}$ that returns a deterministic matching $M \in \mathcal{M}$.

An important property of a (matching) mechanism, is *strategy-proofness.* This means, that there is no incentive for any student not to submit her truthful preferences, no matter which preferences the other students report.

**Definition 3.11: Strategy-Proofness.** *Let* $\succeq \in \mathcal{P}^{|S|}$ *be the (true) preference profile. A deterministic assignment mechanism* $\chi$ *is* strategy-proof *if for every student* $i \in S$ *and* $\succeq'_i \in \mathcal{P}$ *we have* $\chi_i(\succeq) \succeq_i \chi_i(\succeq'_i, \succeq_{-i})$.

Thereby, $\succeq_{-i}$ denotes the preference profile of all agents $i' \in S \setminus \{i\}$. It has been shown that participants in strategy-proof mechanisms such as the Vickrey auction do not necessarily bid truthfully in practice. Therefore, there was a recent discussion about obvious strategy-proofness of extensive form games (Li, 2017). Intuitively, a mechanism is obviously strategy-proof iff the optimality of truth-telling can be deduced without contingent reasoning.

**Definition 3.12: Obviously Strategy-Proofness (Li, 2017).** *A strategy* $\sigma$ *is obviously dominant if, for all other strategies* $\sigma'$*, at any earliest information set where* $\sigma$ *and* $\sigma'$ *diverge, the best possible outcome from* $\sigma'$ *is no better than the worst possible outcome from* $\sigma$*. A mechanism is obviously strategy-proof if it has an equilibrium in obviously dominant strategies.*

### Random Matchings

For randomized mechanisms one has to adapt these design desiderata. Stochastic dominance (SD) is the key concept among all of the following definitions as it provides a natural way to compare random assignments. Let $\Delta$ describe the set of all possible random matchings. With $p_i$ we refer to the assignment of student $i$ in the random matching $p$, and denote with $p_{ib}$ the probability that student $i$ gets allocated bundle $b$. We will omit the subscript $i$ when it is clear which student is meant.

Given two random assignments $p, q \in \Delta$, student $i$ $SD$-prefers $p$ to $q$ if, for every bundle $b$, the probability that $p$ yields a bundle at least as good as $b$ is at least as large as the probability that $q$ yields a bundle at least as good as $b$.

**Definition 3.13: SD-Preference.** *A student $i \in S$* weakly $SD$-prefers *an assignment $p \in \Delta$ over $q \in \Delta$, $p \succeq_i^{SD} q$, if*

$$\sum_{b' \succeq_i b} p_{ib'} \geq \sum_{b' \succeq_i b} q_{ib'}, \forall b \in B$$

*If there additionally exists a bundle $b^* \in B$ such that $\sum_{b' \succeq_i b^*} p_{ib'} > \sum_{b' \succeq_i b^*} q_{ib'}$, student $i$ strongly (or strictly) $SD$-prefers $p$ over $q$, i.e. $p \succ_i^{SD} q$.*

In other words, a student $i$ (weakly) prefers the random assignment $p$ to the random assignment $q$ if $p_i$ (second-order) stochastically dominates $q_i$. Note, that $\succeq^{SD}$ is not a complete relation. That is, there might be assignments $p$ and $q$, which are not comparable with this relation. First-order stochastic dominance[7] holds for all increasing utility functions and implies second-order stochastic dominance, which is defined on increasing concave (risk-averse) utility functions. In other words, risk-averse expected-utility maximizers prefer a second-order stochastically dominant gamble to a dominated one (Müller and Stoyan, 2002).

[7] $p_i$ dominates $q_i$ if $p_{ib} \geq q_{ib}$ for all $b \in B$.

We first generalize *(Pareto) efficiency* to random matchings and lotteries:

**Definition 3.14: Efficiency.** *A random assignment $p \in \Delta$ is called*

*i)* ex post efficient, *if $p$ can be implemented into a lottery over Pareto efficient deterministic assignments.*

*ii)* ordinally efficient, *if there exists no random assignment $q$ stochastically dominating $p$, i.e. $\nexists q \in \Delta : \forall i \in S : q \succeq_i^{SD} p$ and $\exists i \in S : q \succ_i^{SD} p$.*

Ordinal efficiency comes from the Pareto ordering induced by the stochastic dominance relations of individual students. It can be shown that ordinal efficiency implies ex post efficiency (Bogomolnaia and Moulin, 2001).

*Fairness* is another important design goal. A basic notion of fairness for randomized assignments is the *equal treatment of equals*. A stronger property is envy-freeness.

**Definition 3.15: Envy-Freeness.** *A random assignment $p \in \Delta$ is called*

*i)* (strongly) $SD$-envy-free *if $\forall i, j \in S : p_i \succeq_i^{SD} p_j$.*

*ii)* weakly $SD$-envy-free *if $\nexists i, j \in S : p_j \succ_i^{SD} p_i$.*

$SD$-envy-freeness means that student $i$ weakly $SD$-prefers the random matching he got assigned to the random assignments offered to any other student, i.e., a student's allocation stochastically dominates the outcome of every other student. For weak $SD$-envy-freeness it is only demanded that no student's allocation is stochastically dominated by the allocation of another student. We illustrate this difference with the following example:

**Example 3.16:** *Consider two students $1, 2$ each preferring course $A$ over $B$ and a random assignment where $p_1 = (0.5\ 0)$ and $p_2 = (0.4\ 0.2)$. Neither student $1$ does prefer the outcome of student $2$ nor student $2$ does prefer the assignment of student $1$. For student $1$ the probability of getting his most preferred course $(A)$ is higher than the probability for $A$ of student $2$. Though, student $2$ has a lower probability for course $A$, the probability of getting one of the courses $A$ or $B$ is higher than for student $1$. Therefore, the assignment is weakly SD-envy-free.*

*However, the assignment is not strongly SD-envy-free. Neither student 1 nor student 2 prefer their own outcome over the outcome of the other student, since* $0.5 = p_{1A} > p_{2A} = 0.4$ *and* $0.5 = p_{1A} + p_{1B} < p_{2A} + p_{2B} = 0.6$*. That is, the outcomes are not comparable with SD-preference. This can happen as this relation is not complete.*

A *randomized assignment mechanism* is a function $\psi : \mathcal{P}^{|S|} \to \Delta$ that returns a random matching $p \in \Delta$. The mechanism $\psi$ is *ordinally efficient* if it produces ordinally efficient allocations. We call $\psi$ *ex post Pareto efficient* if $p$ can be decomposed as a convex combination of Pareto optimal matchings. $\psi$ is *symmetric*, if for every pair of students $i$ and $j$ with $\succeq_i = \succeq_j$ also $p_i = p_j$. This means that students who have the same preference profile also have the same outcome in expectation. A randomized mechanism is *envy-free* if it always selects an envy-free matching.

We now generalize strategy-proofness to randomized mechanisms. As for envy-freeness there exists a weak and a strong notion of this concept.

**Definition 3.17: SD-Strategy-Proofness.** *Let* $\psi : \mathcal{P}^{|S|} \to \Delta$ *be a random assignment mechanism and* $\succeq \in \mathcal{P}^{|S|}$ *the (true) preference profile.*

*i)* $\psi$ *is called* (strongly) *SD*-strategy-proof *if for every student* $i \in S$ *with* $\succeq_i' \in \mathcal{P}$ $\psi(\succeq) \succeq_i^{SD} \psi(\succeq_i', \succeq_{-i})$.

*ii)* $\psi$ *is called* weakly *SD*-strategy-proof *if there exists no* $\succeq_i' \in \mathcal{P}$ *for some student* $i \in S$ *such that* $\psi(\succeq_i', \succeq_{-i}) \succ_i^{SD} \psi(\succeq)$.

In other words, an ordinal mechanism is strategy-proof if for any agent, the allocation resulting from misreporting is weakly stochastically dominated by the allocation from truthful reporting, with respect to an agent's true preferences. Weak strategy-proofness means that there may not be any student $i$, who prefers $\psi(\succeq_i', \succeq_{-i})$ over the truthful outcome, but there may be students $i$ who neither prefer $\psi(\succeq_i', \succeq_{-i})$ nor $\psi(\succeq)$. We will omit the prefix *SD* for brevity in the following.

Note that there are also weaker notions of strategy-proofness for randomized mechanisms developed in the field of probabilistic social choice that we do not consider in this thesis. These notions are based on different ways of how to compare lotteries. Interested readers are referred to Brandt (2017).

### 3.2.3 Assignment Mechanisms

A lot is known about assignment problems with single-unit demand. There are basically two classes of mechanisms – *random priority mechanisms* and *random assignment mechanisms*. The *Top-Trading-Cycle* mechanism with random endowments (Shapley and Scarf, 1974) as well as the *Random Serial Dictatorship* (RSD) (Abdulkadiroğlu and Sönmez, 1998) are examples of the first class. These algorithms compute deterministic matchings (not lotteries), but the underlying mechanisms are random. They have typically desired properties like strategy-proofness and ex post Pareto efficiency, but in general they perform poorly in terms of fairness (Budish and Cantillon, 2012). A popular example for random assignment mechanisms is the *Probabilistic Serial* (PS) mechanism presented by Bogomolnaia and Moulin (2001), which produces an envy-free assignment with respect to the reported single-unit demand preferences. It is ordinally efficient, but it is only weakly strategy-proof.

Zhou (1990) showed that no random mechanism for assigning objects to agents can satisfy strong notions of strategy-proofness, ordinal efficiency, and symmetry simultaneously with more than three objects and agents. So, we also cannot achieve these properties in combinatorial assignment problems. In the following we discuss generalizations to these mechanisms.

#### Bundled Random Serial Dictatorship

RSD selects a permutation of the agents uniformly at random and then sequentially allows agents to pick their favorite course among the remaining course seats. Gibbard (1977) showed that random dictatorship is the only anonymous and symmetric, (strongly) strategy-proof, and ex post efficient mechanism when preferences are strict. Pycia and Troyan (2018) prove that RSD is the unique mechanism that is obviously strategy-proof, ex post efficient, and symmetric in mechanisms without transfers.

In other words, in our setting with multiple possible bundles we are necessarily bound to a form of RSD if we want to maintain strategy-proofness, symmetry and efficiency. Therefore, we present a generalisation of the RSD called *Bundled Random Serial Dictatorship* (BRSD) in Algorithm 3.2. BRSD orders the students randomly and assigns the most preferred bundle that still has free capacities at all included courses to each student in this order. Although the bundle preferences take some toll on the runtime it is still very fast (in practice).

**Algorithm 3.2:** Pseudocode of BRSD.

**Input :** preferences $(\succeq_i)_{i \in S}$
order $S$ randomly, $M = \emptyset$
**forall** $i \in S$ **do**
  $b \in Acc_i \cup \emptyset$ – the bundle with highest preference that satisfies:
   $\forall c \in b : |M_c| < q_c$
  $M = M \cup (i, b)$
**Output:** deterministic matching $M$

**Theorem 3.18:** *BRSD is strategy-proof, symmetric and computes a Pareto optimal matching in* $\mathcal{O}(|S| \cdot \alpha \cdot k)$*, where* $\alpha = max\{|Acc_i| \mid i \in S\}$ *and* $k = max\{size(b) \mid b \in Acc_S\}$*.*

*Proof. Strategy-proofness* carries over from RSD (Gibbard, 1977). Briefly, since BRSD searches from the top of a student's preference-list downwards the first bundle that does not contain any full course, reporting a wrong order or omitting preferences would not improve the chance to get a higher ranked bundle.
For *Pareto efficiency* we draw on Theorem 3.10. We have to proof that BRSD produces a tree-coalition-free matching. Let $M$ be the outcome of BRSD. Obviously, $M$ is a feasible matching, because BRSD never over-allocates courses. W.l.o.g. let $I = (i_1, i_2, \ldots, i_n)$ be the chosen random ordering. Consider the first student $i_j \in I$ that is not assigned to his first preference. Let $I_{j-1} = (i_1, i_2, \ldots, i_{j-1})$ the first $j-1$ students. Because of the choice of $j$, all agents in $I_{j-1}$ got their first preference.

This set cannot be empty, as at least the first student gets his first preference. Obviously, no agent from $I_{j-1}$ can be in any tree coalition (since we cannot assign any of these agents to a better alternative). We try to construct a tree coalition from agent $i_j$. However, since BRSD assigned the most preferred bundle to him that still had enough capacity, we cannot assign $i_j$ to a more preferred bundle without getting a conflict with an agent in $I_{j-1}$ or violating any capacity constraint. Hence, there cannot be any tree coalition with agents in $I_j = (i_1, i_2, \dots, i_j)$. Let $i_{j'}$ the next student, who did not get allocated his most preferred alternative. Since all agents in $I_{j'-1}$ are in $I_j$ or they got their first preference, none of them can be in any tree coalition. Analogous to $j$ we can show that no agent in $I_{j'}$ can be in any tree coalition. We repeat these arguments until $I_{j'} = I$ and see, that $M$ has to be tree-coalition-free and therefore BRSD is ex post Pareto efficient.
*Symmetry:* Since every permutation of the agents has the same probability, BRSD is symmetric.
*Runtime:* For every $i \in S$ we have to search the first acceptable bundle $b$ that is feasible. Since $size(b) \leq k$, the algorithm needs to check $\mathcal{O}(k)$ capacity constraints. As every acceptable set's size is less than the size $\alpha$ of the largest set of acceptable bundles, the running time of BRSD is in $\mathcal{O}(|S| \cdot \alpha \cdot k)$. □

Note that the outcome of BRSD can be arbitrary smaller than a maximum matching and unfair. Consider an example, where each bidder has the highest preference for the set of all courses, each having a capacity of one. If this was the case, one bidder would get all courses assigned. While the mechanism is strategy-proof and ex post Pareto efficient, the outcome can be considered unfair.

*First-Come First-Served* (FCFS) can be seen as a serial dictatorship. Students login at a certain registration and then reserve the most preferred bundle of courses that is still available. Although the arrival process is not uniform at random, students have little control over who arrives first. While there is a certain time when the registration starts, hundreds of students log in simultaneously to get course seats and it is random who arrives first. We will simulate FCFS via BRSD and run the algorithm repeatedly to get estimates for performance metrics of FCFS.

There are further - more specialized - generalizations of RSD in the literature of matching and auctions. For example, Hashimoto (2018) presents a generalisation of RSD called General-Random-Priority (GRP) algorithm for multi-unit assignments without money and combinatorial auctions. GRP is shown to be feasible and strategy-proof. For obtaining feasibility, the algorithm withholds a fraction of $\varepsilon$ objects, whereby $\varepsilon$ goes to zero when the number of agents goes to infinity. Cechlárová et al. (2014) introduce a many-to-many capacitated house allocation problem, where agents can obtain multiple objects. However, the agents can only rank the objects, but not different bundles of objects. Every agent has a virtual budget and every object a price. Now, bundles satisfying the budgets of the agents are computed, lexicographically ranked, and a generalisation of RSD is used to receive an allocation.

#### Approximative Competetive Equilibrium from Equal Incomes (A-CEEI)

The randomness in RSD as well as the lack of strong fairness properties might be an issue for some applications. The theorem by Gibbard (1977) provides a clear guideline on what is possible in dominant strategies. Therefore, in order to achieve stronger notions of fairness, such as envy-freeness, one has to give up or relax strategy-proofness or ordinal efficiency. One popular example for non-strategy-proof mechanisms is the *Competitive Equilibrium with Equal Incomes* (CEEI), presented by Hylland and Zeckhauser (1979). These algorithms have desirable properties like envy-freeness and ex post Pareto optimality. Kojima et al. (2010) and Azevedo and Budish (2017) show that CEEI is near strategy-proof if the instance is large enough.

Budish (2011) proposes a mechanism for matching with complementarities focusing on different notions of fairness of allocations with ordinal preferences of the agents. His approach is an approximation to CEEI called A-CEEI that assigns agents with approximately equal income. Here, *income* does not mean monetary transfer or utility but is a virtual currency that helps to assign bundles to agents. The assignment is calculated by setting prices on objects and *selling* the bundles to the agents. Since incomes are guaranteed to differ slightly, market clearing is ensured. In general A-CEEI is not strategy-proof, but for large markets it is approximately strategy-proof. Budish (2011) proposes two new characterisations of fairness and shows that A-CEEI meets these. E.g. A-CEEI bounds envy by a single good. This approach is only approximately ex post efficient and the market is cleared approximately only. The worst-case bound of the clearing error depends neither on the number of agents nor on the object capacities. A market clearing error denotes excess demand. That is, A-CEEI approximates feasibility and might allocate more objects than available.

### Bundled Probabilistic Serial

*Bundled Probabilistic Serial* (BPS) by Nguyen et al. (2016) is a generalization of PS to the combinatorial assignment problem. BPS computes a fractional solution via a generalization of the PS mechanism. Informally, in BPS (see Algorithm 3.3) all agents *eat* their most preferred bundle in the time interval $[0, 1]$ simultaneously with the same speed as long as all included objects are available. As soon as one object is exhausted, every bundle containing this object is deleted and the agents continue eating the next available bundle in their preference list. The duration with which every bundle was eaten by an agent specifies the probability for assigning this bundle to this agent.

**Theorem 3.19: (Nguyen et al., 2016).** *BPS is ordinally efficient, envy-free, weakly strategy-proof, and computes a fractional solution respecting (demand) and (supply).*

We further analyze the outcome of BPS. Even if the size of $Acc_i$ for a student $i \in S$ and the assignment matrix of a random assignment $p \in \Delta$ might be of exponential size, the assignment matrix corresponding to the outcome of BPS is sparse.

**Corollary 3.20:** *The outcome of BPS, $x^*$, has at most $|S| \cdot |C|$ nonzero entries.*

*Proof.* Since students only start eating the next available bundle when the current bundle is not feasible anymore, that is, if at least one included course reaches its

**Algorithm 3.3:** Pseudocode of BPS.

```
Input  : preferences (⪰_i)_{i∈S}
t = 0
x_ib = 0, ∀i ∈ S, b ∈ B
while t < 1 do
    D = ∅
    dem_j = 0, ∀j ∈ C
    forall i ∈ S do choose first valid bundle b ∈⪰_i: D ← b
    forall b ∈ D do
        forall j ∈ b do dem_j++
    δ = min { q_j / dem_j | j ∈ C }
    t+ = δ
    δ* = δ − (t − min {1, t})
    forall i ∈ S do x_ib+ = δ*
    forall j ∈ C do
        q_j− = δ* · dem_j
        if q_j = 0 then ∀b ∈ B : j ∈ b : delete b
Output: allocation x* = (x_ib)_{i∈S,b∈B}
```

capacity, a student can at most change a bundle $|C|$ times in the BPS mechanism. Therefore $x^*$ cannot contain more than $|C|$ nonzero entries for each student □

### 3.2.4 Implementing Random Assignments

Unfortunately, in contrast to the result of PS, the outcome of BPS is not implementable into a lottery of deterministic matchings in general if the maximal bundle size $\ell > 1$. To circumvent this, one can either scale the fractional solution $x^*$ by a factor $\alpha \in (0, 1)$ such that the decomposition becomes possible (Lavi and Swamy, 2011) or one allows for the relaxation of some constraints. Nguyen et al. (2016) present a mechanism to decompose the BPS solution into a lottery over deterministic matchings that over-allocate each course by at most $\ell - 1$ seats, i.e. the (demand) constraints are fulfilled and only the (supply) constraints are relaxed.

#### Description of the Decomposition Mechanism

In the polynomial time *lottery algorithm* (see Algorithm 3.5), we find at most $d + 1$ integral points, the convex hull of which is arbitrarily close to the fractional solution $x^*$ that we get from BPS. The lottery algorithm then returns a lottery over these $d+1$ integral vectors, which is close to $x^*$ in expectation. Variable $d$ describes the dimension of the problem. In this lottery algorithm, we use a subroutine to return an integer point $\bar{x}$ such that $u^\intercal \bar{x} \geq u^\intercal x^*$ for an arbitrary vector $u$. This subroutine is called *iterative rounding algorithm* (IRA) and proceeds as described in Algorithm 3.4.

In Algorithm 3.4 we first fix variables to 0 or 1 if they already have this value in step 1a. If one cannot fix any variable but there are still fractional components left, one can find a (supply) constraint that is always fulfilled if we relax it by $\ell - 1$, even if we round up all remaining fractional components (step 1b). We delete those constraints and reoptimize the reduced problem in step 2. The constraint violation in step 1b of the IRA is depicted in Figure 3.2. Independent from the rounding of the remaining fractional components of the current point $x$, the rounded point

**Algorithm 3.4:** Pseudocode of the iterative rounding algorithm.

**Input :** (fractional) solution $x$, (demand), (supply), additional constraints $\mathfrak{C}$

*1a:* Delete all $x_i = 0$, $x_i = 1$, update the constraints and go to 1b.

*1b:*

If there is no $x_i \in \{1, 0\}$ one can find at least one (supply)-constraint with

$$\sum_{i \in S} \sum_{b \in B} b_j \lceil x_{ib} \rceil \leq q_j + \ell - 1, \quad j \in C \qquad \text{(relaxed supply)}$$

Delete those constraints and go to 2.

*2*

Solve $max \{u^\tau x \mid (demand), (supply), \mathfrak{C}, x \in \mathbb{R}_{\geq 0}\}$

**if** *all* $x_i \in \{0, 1\}$ **then** return $x$

**else** go to 1a

**Output:** integral solution $x$ to (demand), (relaxed supply)

always will stay in the orange circle and therefore wont hurt the relaxed constraints for the courses $j = 2, 3$. Hence, we delete the original constraints for $j = 2, 3$ and allow for a new (fractional) solution in the next optimization step (step 2) that has ideally more integral components than the current solution.

Next, we discuss the lottery algorithm (see Algorithm 3.5). Let $\mathcal{B}(x^*, \delta) = \{x \mid |x^* - x| \leq \delta\} \subseteq \{x \in \mathbb{R}_{\geq 0} \mid (demand)\}$ with $\delta > 0$. The parameter $\delta$ in $\mathcal{B}(x^*, \delta)$ determines a ball around $x^*$ such that the demand constraints $\sum_{b \in B} x_{ib} \leq$

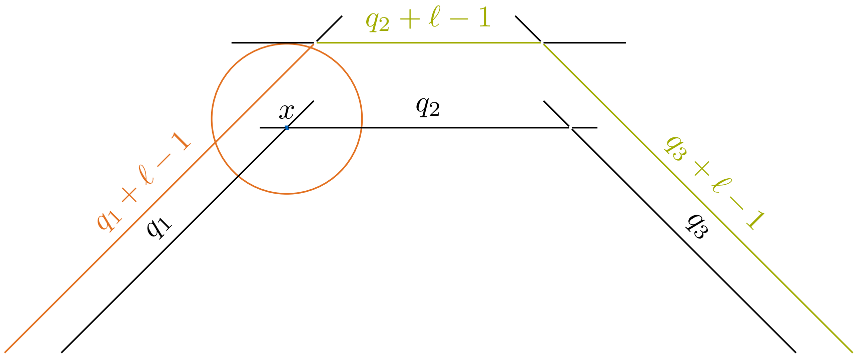


Figure 3.2: Constraint violation in the IRA

1 are not violated. It is always possible to determine such a $\delta$. If there is no slack in the demand constraints one has to scale down the fractional solution $x^*$. Afterwards one has to adjust the allowed error $\varepsilon$ such that after scaling and decomposition the original $\varepsilon$ is still fulfilled. Here, $|x - y|$ describes the Euclidean distance between two vectors $x$ and $y$.

**Algorithm 3.5:** Pseudocode of the lottery algorithm.

**Input :** fractional solution $x^*$

*1:* Set $Z = \{\text{IRA}\,(x^*)\}$, i.e., find an integer solution via IRA.

*2:*

  $y = argmin\,\{|x^* - y| \mid y \in conv\,(Z)\}$

  **if** $|x^* - y| < \varepsilon$ **then** END

*3:* Choose $Z' \subseteq Z$ of size $|Z'| \leq d$ and $y \in conv\,(Z')$ : $z = x^* + \delta \frac{x^* - y}{|x^* - y|}$

*4:* $z' = \text{IRA}(z, (demand), (supply), (x^* - y)^\tau z' \geq (x^* - y)^\tau z)$

*5:* $Z = Z' \cup \{z'\}$ and go back to 2.

**Output:** convex combination of final $y$

In each iteration the algorithm decreases the distance between $y$ and $x^*$ by adding a new integral solution to the solution set $Z$ and terminates when the distance between $y$ and $x^*$ is smaller than $\varepsilon$. That is, we consider $y$ as a good approximation for $x^*$ and return the support of $y$. The algorithm tries to get $x^*$ covered by the convex hull of $Z$ ($conv(Z)$). All solutions in $Z$ that are not part of the support of $y$, calculated in the quadratic optimization problem (QOP) in step 2, are deleted (step 3). Thus, although we add a new integral solution to $Z$ in each iteration, the size of $Z$ never grows above $d + 1$, since as long as $y \neq x^*$, $y$ always has to be on a face of $conv(Z)$. Hence, the support of $y$ consists of at most $d$ solutions. Step 4 ensures that we search in the right direction for new integral solutions. As a side product the QOP also calculates the coefficients $\lambda^{(k)}$ for the convex combination and we have $x^* \approx y = \sum_{k=1}^{|Z|} \lambda^{(k)} x^{(k)}$, for $x^{(k)} \in Z$. Figure 3.3 shows a graphical representation of one iteration of Algorithm 3.5.

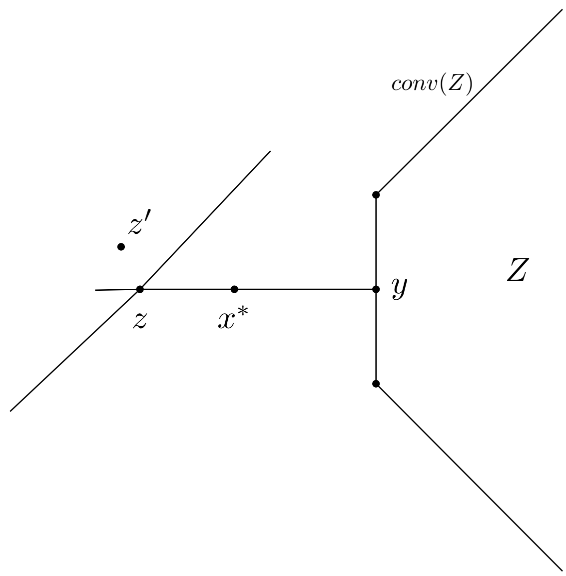


Figure 3.3: Graphical representation of one iteration of the lottery algorithm.

#### Analysis of the Decomposition Mechanism

We now prove the correctness of Algorithm 3.5. The first critical point is the assumption that always a ball around $x^*$ exists, such that the (demand) constraints are fulfilled. If for $x^*$ all (demand) constraints have a positive slack the claim follows directly with $\delta = \min$ slack. Let us assume, there is at least one (demand) constraint that is tight. We scale $x^*$ by a $\delta \in (0, 1)$, i.e. $x^*_{sc} = (1 - \delta) \cdot x^*$, and decompose $x^*_{sc}$ instead of $x^*$. To ensure that the found $y$ is in an $\varepsilon$-environment of $x^*$, we divide $\varepsilon$ into $\varepsilon_{sc} = {}^{(a-1)\varepsilon}/_{a}$ and $\varepsilon_\delta = {}^{\varepsilon}/_{a}$ with $a > 1$, i.e. $\varepsilon = \varepsilon_{sc} + \varepsilon_\delta$ (see Figure 3.4). The parameter $a$ determines the proportion between scaling and allowed error for the decomposition. We now can define $\delta$ depending on $\varepsilon$ and $a$:

$$|x^* - x^*_{sc}| = \delta \cdot |x^*| \leq \frac{\varepsilon}{a} \;\Rightarrow\; \delta \leq \frac{\varepsilon}{a\,|x^*|}$$

Using this $\delta$ we calculate a decomposition for $y$ with $|y - x^*_{sc}| < \varepsilon_{sc}$. Since $|x^* - x^*_{sc}| \leq \varepsilon_\delta$ we have $|y - x^*| < \varepsilon$. Hence, we can summarize:

**Lemma 3.21:** *For every solution $x^*$ respecting (demand) and (supply) it is possible to find a $\delta > 0$ such that $\mathcal{B}(x^*, \delta) \neq \emptyset$ and Algorithm 3.5 returns a convex combination for a $y$ with $|y - x^*| < \varepsilon$.*

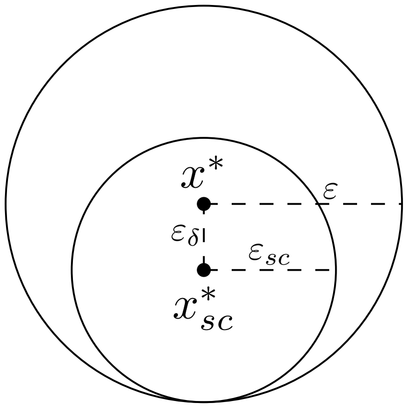


Figure 3.4: Graphical representation of the change of $\varepsilon$.

Next, we have to ensure, that Algorithm 3.4 always terminates. Lemma 3.22 helps us proofing this.

**Lemma 3.22:** *If $\{x \in \mathbb{R}_{\geq 0} \mid (demand), (supply), \mathfrak{C}\} \neq \emptyset$, Algorithm 3.4 calculates a solution $x$ that has at most $2 \cdot |\mathfrak{C}|$ fractional components.*

*Proof.* Suppose that $|\mathfrak{C}| = \gamma$. The proof of Algorithm 3.4 with $\gamma = 0$ in Nguyen et al. (2016) uses a counting argument of linear programming that says that there are as many linearly independent and binding constraints as there are non-zero decision variables. We build on this proof. Assume the current $x$ has still $n$ non-zero components, and Algorithm 3.4 can neither run step 1a nor step 1b. That is, all $n$ remaining components have to be fractional and there cannot be any (supply) constraint in the updated program anymore. This follows directly from Nguyen et al. (2016). Assume there are still (supply) constraints in the reduced problem. If we ignore the constraints $\mathfrak{C}$, we could delete (some of) these constraints. Since we have a maximization problem, and the (supply) constraints are $\leq$-constraints with non-negative coefficients, adding further constraints would not increase the left hand side of these constraints. That is, we also could delete these (supply) constraints in the problem with $\mathfrak{C}$ – that would be a contradiction to the assumption, that we cannot enter step 1b. Hence, the reduced problem only consists of constraints in $\mathfrak{C}$ and (demand), and has $n$ linear independent binding constraints.

We now consider the worst case of this setting. Since (demand) constraints do not share any variables and have a binary right hand side, constraints of this type having only one variable cannot exist in the current state. Otherwise, these variables would have to be binary as well, since the constraints are binding. But this would be a contradiction to the assumption that we cannot enter step 1a. Hence we consider the case where all $\gamma$ constraints from $\mathfrak{C}$ are binding. Additionally we have $n - \gamma$ binding (demand) constraints, which have disjoint sets of variables. Hence, we have to distribute $n$ variables to $n - \gamma$ constraints. Therefore, at most $\gamma$ constraints can consist of more than one variable. That is, in the worst case we have to be in a state, where only $\gamma$ demand constraints and the $\gamma$ $\mathfrak{C}$-constraints are linear independent and binding. Hence, the current $x$ has at most $n = 2\gamma$ fractional components. □

We use Algorithm 3.4 with one additionally constraint. Hence, it can happen that the algorithm runs in a state, where 2 fractional components are left, but neither step 1a nor step 1b can be used. That is, one demand constraint with 1 on the right hand side and the additional constraint is left. However, now one can round one of the two variables to 1 and the other to 0 (or both to 0), such that the additional constraint is fulfilled, since we use Algorithm 3.4 only to find an integral solution and this solution does not need to be *optimal* in any sense.

Using the correctness proof for Algorithm 3.4 and Algorithm 3.5 from Nguyen et al. (2016) together with Lemma 3.22 and Lemma 3.21 we can conclude:

**Theorem 3.23: Near Feasible Decomposition.** *If $x^*$ is an envy-free and efficient solution to (demand) and (supply), Algorithm 3.5 returns in polynomial time a lottery over integral solutions respecting (demand) and (relaxed supply) that is envy-free and efficient in expectation, and asymptotically strategy-proof.*

Finally, we can prove that we can decompose the BPS-solution with desired properties in polynomial time.

**Theorem 3.24: BPS-Lottery.** *The solution of BPS can be implemented into a lottery that is envy-free and ordinal efficient in expectation, asymptotically strategy-proof, and that over-allocates each object $c \in C$ by at most $\ell - 1$. This lottery can be constructed in polynomial time.*

*Proof.* The first part follows with Theorem 3.19 (BPS computes an efficient and envy-free solution to (demand) and (supply)) directly from Theorem 3.23. BPS always returns a sparse solution, i.e. a solution with at most $|S| \cdot |C|$ nonzero entries (Corollary 3.20). Furthermore, no variable $x^*_{ib} = 0$ can be of value 1 in any solution included in the lottery, otherwise the probability of allocating $b$ to $i$ would be nonzero in expectation. However, this cannot be the case, since the expected outcome of the lottery has to equal the fractional solution. Therefore, it is sufficient to reduce $x^*$ to its nonzero components, and useing this reduced (polynomial size) $x^*$ as input for Algorithm 3.5. Hence, because of Theorem 3.23, the solution can be decomposed in polynomial time. □

## 3.3 Preference Elicitation

This section focuses on the preference elicitation, which is important given the exponential set of possible bundles students might be interested in. We first introduce the environment and the problem for students, before we discuss different approaches to elicit preferences.

### 3.3.1 Background on the Application

The Department of Informatics has been using stable matching mechanisms for the assignment of students to courses since 2014 (Diebold and Bichler, 2017; Diebold et al., 2014). The system provides a web-based user interface and every semester almost 1500 students are being matched to practical courses or seminars via the deferred acceptance algorithm for two-sided matching or random serial dictatorship for one-sided matching problems.

In the context of the study reported in this chapter, the web-based software was extended with BPS, the lottery mechanism for decomposing fractional solutions, and BRSD. 1439 Students in computer science and information systems in their second semester participated in the matching during the summer term 2017 and they could choose tutorial groups from several courses including linear algebra, algorithms, software engineering, and operations research. A computer science student could have preferences for up to 5760 ($= 10 \cdot 24 \cdot 24$) bundles[8] and an information systems student could have preferences for up to 5184 ($= 9 \cdot 24 \cdot 24$) bundles.[9] During the winter term 2017/2018 and 2018/2019, 1778 (resp. 1683) computer science and information systems students in their third semester participated in the matching and could choose bundles of tutor groups out of four classes. A computer science student could have more than 700,000 different bundles.[10]

### 3.3.2 Automated Ranking of Bundles

A naive approach would be to let the students rank bundles on their own by choosing the time-slots they want to have in their preference list. This would take a lot of time and lead to a substantial missing bids problem. We developed an algorithm that allows to rank-order all possible bundles based on a few parameters that students need to specify. For this, we can leverage prior knowledge about timely preferences of students for schedules of tutorials and lectures.

Students' preferences mainly concern their commute and the possibility to free large contiguous blocks of time (e.g., a day or a half-day) that they can plan for other activites (e.g., a part-time job). In larger cities such as Munich, the time that students spend for commuting is significant. Also long waiting times

[8]Consisting of the courses: linear algebra, algorithms, software engineering.
[9]Consisting of the courses: operations research, algorithms, software engineering.
[10]The computer science students need tutorials from all four classes ($< 22 \cdot 25 \cdot 26 \cdot 52$).

between courses are perceived as a waste of time as it is often difficult to work productively in several one- or two-hour breaks without appropriate office facilities available. For example, if a student had a tutorial on linear algebra in the morning, a lunch break, and then the tutorials for algorithms and software engineering in the afternoon of the same day with the minimal time for breaks specified, this would be considered ideal. The desired length for breaks between tutorials and for the lunch break are considered parameters with default values in the preference elicitation.

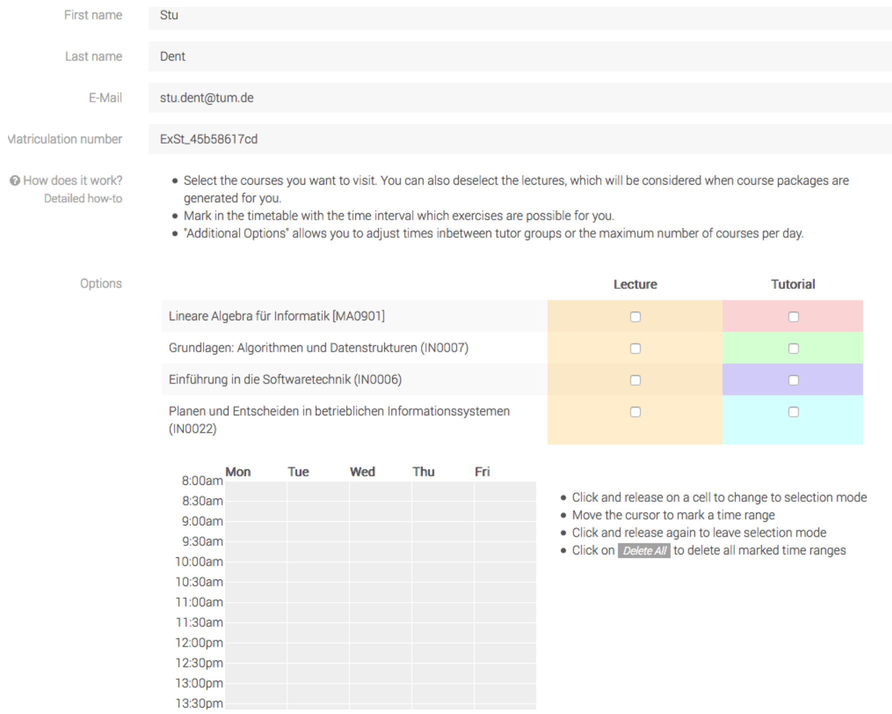


Figure 3.5: User Interface to Select Courses

Figure 3.5 shows the initial page where students can select the courses of interest. On this page students choose the lectures and tutorials they are interested in. The selected lectures will be considered in the bundle generation as constraints, i.e. if a time-slot of a tutorial overlaps with the time of a selected lecture, it will no longer be considered in order to allow students to participate in the lecture. In a second step, the students mark available time ranges in a *weekly schedule* (see Figure 3.6). The day is partitioned into weekdays and time blocks of 30 minutes from 8:00 AM to 8:30 PM. If a tutorial is selected, all time-slots of this tutorial will be highlighted with a specific color. Thus, students learn when the tutorials and lectures of interest take place.

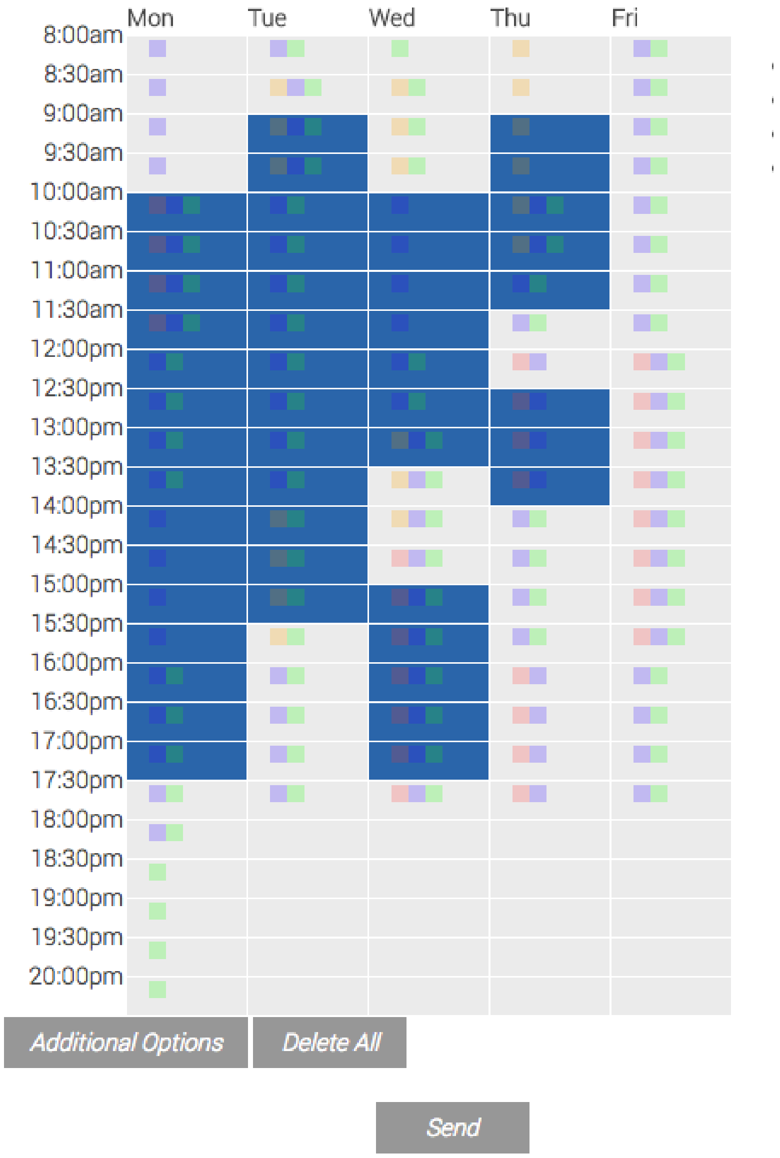


Figure 3.6: The Week Schedule

Students can set a minimal amount of time for a lunch break and a minimal amount of time in-between two events (default value is 15 minutes). We also allow students to provide weights $\{1, \ldots, 5\}$ for the different days. That is, the students can express preferences over the days.

The preferences elicited on this screen are input for an algorithm that uses prior knowledge about student preferences to rank-order all possible bundles. The algorithm first generates bundles that satisfy all constraints and then ranks them. Finding the bundles that do not violate constraints of the students (e.g., lectures to be attended) can be cast as a *constraint satisfaction problem*. After the feasible bundles are generated, we rank these bundles. For this we assign a score to each bundle that considers

- how many days a student needs to come to the university per week in total,
- the preference ordering over the days,
- the total time a student has to stay at the university each day, and
- the length of the lunch breaks between courses.

The score for a bundle $b$ of courses across the week is the sum of the daily score ($score\,(b, day)$) for all weekdays $d$. The daily score is computed as

$$score\,(b, day) = \left(\frac{w\,(b, day)}{sp\,(b, day)} \cdot f\,(sp\,(b, day)) + br\,(b, day)\right) \cdot prio\,(day) \qquad (3.3)$$

This score is scaled between 0 and 27.5 at a maximum and it considers how well the day is utilized with courses. Ideally, a student would have all his tutorials on a single day (his most preferred day) with a 1-hour lunch break and a minimal time for breaks in between courses. This would yield 27.5 points.

The time spent at the university per day $sp\,(b, day)$ is considered relative to the time a student attends courses on that day ($w\,(b, day)$). These courses include tutorials and lectures. The ratio is used to weigh the score for a day ($f\,(sp\,(b, day))$). Hence, a day where students do not spend more time in breaks than the minimum number of minutes specified maximizes the score. The function $f\,(\cdot)$ assigns 1 point for up to 2 hours spent at the university on a day ($sp(b, day) \leq 2$), 2 points for up to 4 hours, 3 points for up to 6 hours, 4 points for up to 8 hours, but only 2 points for days where a student is between 8 and 10 hours at the university. Longer schedules are not permitted.

A second component in the daily score ($score(b, day)$) is the lunch break. A 1-hour break was considered best. The scoring function $br(\cdot)$ would assign 0 points for lunch breaks less than 30 minutes, 1 point for 30-45 minutes, 1.5 points for 45-60 minutes, 2 points for 60-75 minutes, and again a low number of points for longer breaks. Students could also exclude schedules with a break less than a certain time, say 30 minutes.

The daily scores are then multiplied by the priority of the day [1..5]. If students do not have to visit the university at day $d$, they get a fixed score of 30 for this day. The overall score of a bundle $b$ is the sum of the $score(b, d)$ for all weekdays. As a result of this scoring rule, the more days the student can stay at home, the higher is the score of this bundle. As a simplified example, if a student had to come to the university on three different days to attend one course only, this bundle would get a score of less than 25, while if he could attend all courses on a single day with minimal breaks, this will get an overall score of more than 80 (for these three days).

In other words, the scoring rule will place bundles that use a minimal number of days (ideally the most preferred days) with only a few breaks but a one hour lunch break on top of the preference list. This would minimize the commute time and maximize the contiguous time a student can devote to learning or work. If the breaks between courses grow larger or courses take place on different or more days, this decreases the score. Ties are not impossible but almost never occur such that the algorithm typically generates a strict ranking of the feasible bundles.

Even if it is a fair assumption that students have quite homogeneous preference structures, there might be some special cases we cannot cover with such a scoring rule. Therefore we give the students the possibility to adjust the outcome of this scoring procedure. On the ranking page, we display the 30 top rated pre-ranked bundles and the students can adapt this ranking manually, go back to the previous screen and adapt the input parameters, or just accept the ranking with a single click (see Figure 3.7). Note that $\approx 90\%$ of the students received one of their top ten ranked bundles and only a few students received a bundle with a rank less than 30. So, if a student inspects and confirms the ranking of the first 10-30 bundles, this covers the most important quantile of the overall ranking list. We generated a ranking over 200 bundles for each student in advance based on the pre-specified parameters and further preferences only if necessary.[11]


Figure 3.7: Page with top-ranked bundles

So far, we described the user interface for the winter term 2017/18. The user interface in the summer term 2017 required students to explicitly drag and drop

[11]Since the processor load of our server was low with this setting, we increased the number of generated bundles per student to 400 in WS 2018/2019 and gave the students the opportunity to rank the top 40 bundles.

the pre-ranked bundles on a screen. This was considered tedious such that in the winter term the generated ranking was suggested to students right away without any drag-and-drop activies required and could be confirmed without much effort. The main web page and the main steps students have to take are summarized in Figure 3.8.

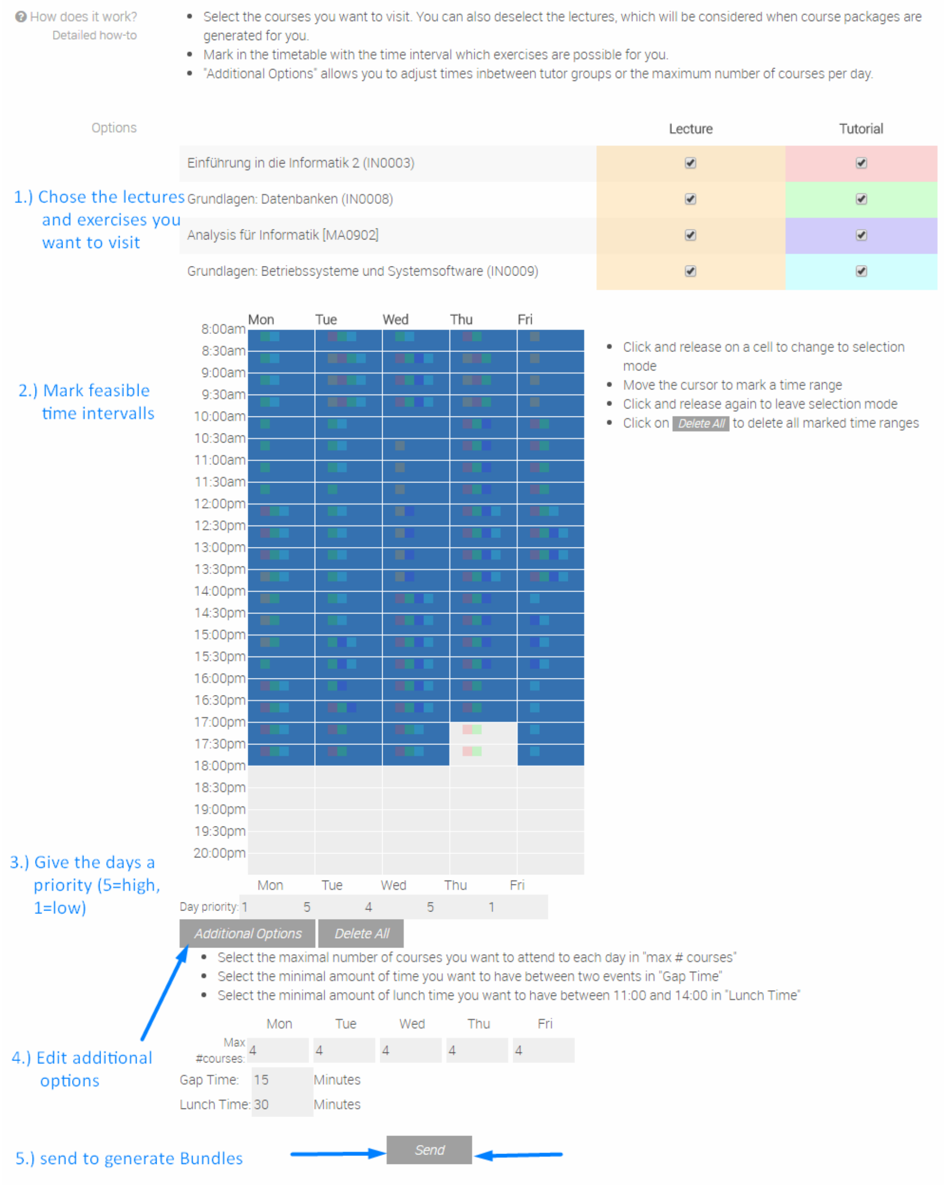


Figure 3.8: Process to rank-order bundles

### 3.3.3 Challenges of Course-Level Scoring

Ranking an exponential set of bundles is a general issue in combinatorial assignment problems, and one might ask if alternative methods are available. Budish et al. (2017) describes the preference elicitation used at the Wharton School of Business. Students could report cardinal item values on a scale of 1 to 100 for any course they were interested in taking. In addition, they could report adjustments for pairs of courses, which assigned an additional value to schedules that had both course sections together. With this information, courses were scored and transformed into an ordinal ranking over feasible schedules. The authors argue that they felt that "adding more ways to express non-additive preferences would make the language too complicated." Wharton also provided a decision support tool listing the 10 most-preferred bundles, which allowed students to inspect top-ranked schedules and modify the cardinal values.

Two problems make this method challenging to apply. First, the ranking is sensitive to minor changes in the weights, which is a well-known issue in multi-criteria decision making with additive value functions. Evaluation is characterized by a substantial degree of random error, and the amount of error tends to increase as the decision maker attempts to consider an increasing number of attributes (or courses in our case) (Bowman, 1963; Fischer, 1972). Difficulties in the calibration of scores for each course can lead to substantial differences in the resulting ranking.

Second, and more importantly, significant non-linearities arise due to the timely preferences of students in the assignment of tutorials, making it impossible to describe the preferences via a course-level utility function as proposed by Budish et al. (2017). Even if three tutorials get a high number of points, this does not mean that their combination is preferable by a student as these tutorials might be on different days or have long breaks inbetween.

To investigate if the method proposed by Budish et al. (2017) is applicable for our allocation problem, we translate the ranking of bundles into a set of inequalities with weights ($w$) as variables. Following revealed preference theory (Mas-Colell

| | student 1 | student 2 | student 3 | student 4 |
|---|---|---|---|---|
| min lunch time | 45 min | | 0 min | |
| time ranges | 8am to 6pm | | 10am to 6pm | |
| feasible days (score) | Mo(2), Tu(4), We(5), Th(4), Fr(1) | Tu(5), We(5), Th(2) | Mo(5), Tu(3), We(5), Th(3) | Mo(5), Tu(5), We(5), Th(2), Fr(1) |
| # of bundles | 8503 | 4120 | 4425 | 12370 |
| $err\ (\varepsilon)$ | $\approx 0.005$ | $\approx 0.004$ | $\approx 0.004$ | $\approx 0.005$ |

Table 3.1: Parameters of the (REV) on the data for the winter term.

et al., 1995), we use these inequalities to understand whether there is any set of weights that would allow to describe the ranking using a utility function $\sum_{i \in C} b_i w_i + \sum_{\substack{i,j \in C \\ j>i}} b_i b_j w_{ij}$. The function $r(b)$ describes the rank of a bundle, while $b$ is a binary parameter vector with each component $b_i \in \{0, 1\}$ showing whether a course $i \in C$ is part of a bundle or not. The objective function minimizes the sum of error variables $\varepsilon$ in (REV). If there is any set of weights that could reflect the ranking of bundles in our experiments without these error variables, the resulting optimal objective value would be zero. For every violation of a constraint one has to increase the respective error variable to a positive value.

$$
\begin{aligned}
\min_{\text{s.t.}} \quad & err\,(\varepsilon) = \sum_{b \in B} \varepsilon_b + \sum_{\substack{i,j \in C \\ j>i}} \varepsilon_{ij} && && \text{(REV)} \\
\sum_{i \in C} b_i w_i + \sum_{\substack{i,j \in C \\ j>i}} b_i b_j w_{ij} + \varepsilon_b & \geq \sum_{i \in C} b'_i w_i + \sum_{\substack{i,j \in C \\ j>i}} b'_i b'_j w_{ij} && && \forall b, b' : r\,(b') = r\,(b) + 1 \\
w_i + w_j + w_{ij} + \varepsilon_{ij} & \geq 0 && && \forall i, j \in C, j > i \\
w_i & \in [0, 1] && && \forall i \in C \\
w_{ij} & \geq -2 && && \forall i, j \in C, j > i \\
\varepsilon_b, \varepsilon_{ij} & \geq 0 && && \forall i, j \in C, j > i, b \in B
\end{aligned}
$$

We solved the linear program for a large number of student preferences in our environment, and none of the problems were feasible. Table 3.1 shows the parameters, the number of generated bundles as well as the respective objective function value of (REV) for the different sample preferences.

We had preferences ranking 4000 to 12000 bundles for the courses of the winter term. None of these settings could be solved with objective value zero, that is, the generated preference lists are not representable with a linear model with adjustment-terms used by Budish et al. (2017). Even if it was possible to find such a vector of course-level weights, it would probably be very difficult to parametrize by students. The way Budish et al. (2017) elicit preferences might be sufficient for settings, where students only are interested in a very small subset of groups of the courses. However, assuming that students are able to adjust weights for up to 50 groups per course is utopian.

Eliciting preferences for hundreds of bundles is a challenging problem, but the quality of any mechanism for combinatorial allocation problems depends crucially on this input. There will be differences in the type of decision support one can provide in various types of applications. However, it is typically important that the timely preferences for students are captured.

## 3.4 Empirical Analysis

In Section 3.2.2 we have summarized first-order design desiderata for assignment problems: strategy-proofness, fairness, and efficiency. Now we introduce second-order design goals and respective metrics allowing us to compare the assignments of BPS and FCFS empirically. Then we provide numeric results and summarize the outcomes of a survey we conducted after the matching.

### 3.4.1 Metrics

Apart from efficiency, fairness, and strategy-proofness, *popularity* was raised as a design goal. An assignment is called popular if there is no other assignment that is preferred by a majority of the agents. Popular deterministic assignments might not always exist, but popular random assignments exist and can be computed in polynomial time (Kavitha et al., 2011). However, Brandt et al. (2017) prove that popularity is incompatible with very weak notions of strategy-proofness and envy-freeness, but it is interesting to understand the popularity of BPS vs. BRSD.

In our empirical evaluation we analyze whether BPS or FCFS are more popular. To measure popularity we first define the function $\phi_i(b, b') : B \times B \to \{\pm 1, 0\}$ associated with the preference relations:

$$\phi_i(b, b') = \begin{cases} +1 & \text{if } b \succ_i b' \\ -1 & \text{if } b' \succ_i b \\ 0 & \text{else} \end{cases} \tag{3.4}$$

**Definition 3.25: Popularity.** *A random assignment $p \in \Delta$ is* more popular *than an assignment $q$, denoted $p \blacktriangleright q$, if $\rho(p, q) > 0$ with*

$$\rho(p, q) = \sum_{i \in S} \sum_{b, b' \in B} p_{ib} \cdot q_{ib'} \cdot \phi_i(b, b') \tag{3.5}$$

*A random assignment $p$ is* popular*, if $\nexists q \in \Delta : q \blacktriangleright p$.*

Apart from popularity, the size and the average or median rank are of interest. The *size* of a matching describes the number of matched agents. The *average rank* is only meaningful in combination with the size of the matching, because a smaller matching could easily have a smaller average rank. We report the average rank, because it has been used as a metric to gauge the difference in welfare of matching algorithms in Budish et al. (2017) and Abdulkadiroğlu et al. (2009), two of the few experimental papers on matching mechanisms.

The *profile* contains more information as it compares how many students are (fractionally) assigned to their first rank, how many to their second rank, and so on. The profile of two matchings is not straightforward to compare. We want to compare multiple profiles based on a single metric, and decided to use a metric similar to the *Area under the Curve of a Receiver Operating Characteristic* in signal processing (Hanley and McNeil, 1982), which was already used by Diebold and Bichler (2017). The *Area Under the Profile Curve Ratio* (AUPCR) is the ratio of the Area Under the Profile Curve (AUPC) and the total area (TA) and is scaled between 0 and 100%, where the AUPC describes the integral below the profile curve. The

AUPCR up to a specific rank $r$ is equal to the probability that a matching mechanism will match a randomly chosen student at least to his $r$-th preference.

**Definition 3.26: AUPCR (Diebold and Bichler, 2017).** *Let $C$ be the possible courses with $c \in C$ and $Q$ be the sum of all capacities, regarding the students $i \in S$ the AUPCR is defined as follows:*

$$\begin{aligned} TA(M) &= |C| \cdot min\{|S|, Q\} \\ AUPC(M) &= \sum_{r=1}^{|C|} |\{(i,c) \in M \mid rank(i,c) \le r\}| \\ AUPCR(M) &= \frac{AUPC(M)}{TA(M)} \end{aligned}$$

For the allocation of bundles we have to rewrite the definition of the AUPCR.

**Lemma 3.27: AUPCR.** *With $R$ denoting the number of possible ranks and $b \in B$, the AUPCR can be rewritten as:*

$$AUPCR(M) = \frac{1}{R} \sum_{r=1}^{R} \frac{|\{(i,b) \in M \mid rank(i,b) \le r\}|}{|S|}$$

*Proof.* Since students are interested in seats for more than one course, the sum of capacities of all selectable courses (tutor groups) is significantly higher than the number of participating students, therefore $min\{Q, |S|\} = |S|$. For matching problems with single unit demand, the number of possible ranks equals the number of courses, i.e. $|C| = R$. That is, we can rewrite $TA(M) = R \cdot |S|$. Since the students do not rank single courses but bundles of courses, we have to replace $c \in C$ by $b \in B$. We use this to get our conclusion:

$$\begin{aligned} AUPCR(M) &= \frac{AUPC(M)}{TA(M)} = \frac{\sum_{r=1}^{R} |\{(i,b) \in M \mid rank(i,b) \le r\}|}{R \cdot |S|} \\ &= \frac{1}{R} \sum_{r=1}^{R} \frac{|\{(i,b) \in M \mid rank(i,b) \le r\}|}{|S|} \end{aligned}$$

□

### 3.4.2 Empirical Results

The first application from the summer term 2017 comprised 1415 students and 67 courses (see Table 3.3). Overall, we had a list of 5847 different bundles for the summer term. We simulated FCFS via BRSD on the preferences collected for the BPS. BPS is weakly strategy-proof and in such a large application it is fair to assume that students do not have sufficient information about the preferences of others. In the survey, we will see that a small proportion of the students reported that they deviated from truthful bidding and did not report some of their preferred time-slots. However, taking the preferences for bundles of tutor groups elicited for the BPS allows for a comparison with BRSD. To compare the result of BPS and BRSD we actually would have to run the BRSD for all permutations of the students. Note that computing probabilities of alternatives in RSD explicitly is $\#P$-complete (Aziz et al., 2013). We ran BRSD 1000 to 1,000,000 times with the same preferences but random permutations of the order of students and derived estimates for the different metrics. Since these results are very close, one can assume, that 1Mio runs of BRSD generate a good approximation to the (real) induced random matching.

#### Popularity

For the data from the summer and the winter term, BPS is more *popular* than BRSD(1000000). 636 students prefer BPS to FCFS, while 96 students prefer FCFS to BPS. 683 students are indifferent (see Table 3.2). A positive popularity score as described in Definition 3.25 means that BPS is more popular than the BRSD outcome and the score for BPS is between 1.78 and 3.41 for the three instances (compared to BRSD(1000000)). For the data from the winter term 754 students prefer BPS to FCFS, while 120 students prefer FCFS to BPS. 862 students are indifferent (see Table 3.2). Table 3.2 summarizes popularity and stochastic dominance for the three experiments. The syntax for the *$SD$-preference* is the number of students preferring (BPS|BRSD(x)). It shows that BPS is preferable to BRSD according to $SD$-preference.

| Metric | BRSD(1000000) |
|---|---|
| popularity summer 17 | 2.73635 |
| popularity winter 17/18 | 3.41499 |
| popularity winter 18/19 | 1.78421 |
| $SD$-prefer summer 17 | (636\|96) |
| $SD$-prefer winter 17/18 | (754\|120) |
| $SD$-prefer winter 18/19 | (767\|87) |

Table 3.2: Popularity and stochastic dominance of BPS vs. BRSD

### Rank and Size

Table 3.3 reports that in terms of average rank, average size, and the probability of being matched to an acceptable bundle, BPS achieves higher scores in the summer term. Only the AUPCR for BRSD(1000000) is slightly better than for BPS. The computation times were negligible for BRSD (0.007 seconds per run). BPS required 0.12 seconds computation time with additional 6 minutes for the lottery algorithm in the summer term. This shows that BPS is a practical technique even for large assignment problems.

| Metric | BPS | BRSD(1000) | BRSD(1000000) |
|---|---|---|---|
| exp. rank | 2.20163 | 2.20867 | 2.20835 |
| exp. size | 1086.58 | 1085.84 | 1085.79 |
| prob. match | 0.767901 | 0.76738 | 0.767345 |
| AUPCR | 0.747419 | 0.74679 | 0.750782 |
| weak envy | 0 | 380 | 381 |
| strong envy | 0 | 981 | 1064 |

Table 3.3: Summary statistics for the summer term 2017.

In the BPS outcome 72.7% of the students receive an assignment ranked in their top ten while in BRSD 72.6% receive such an outcome (see Table 3.4 for BPS and 3.5 for BRSD with 1 mio. permutations of the students). Table 3.4 reports the probability of being matched to a particular rank and the AUPC in percentage for BPS, and Table 3.5 shows the rank profile for BRSD.

| Rank | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Prob match(%) | 54.174 | 5.691 | 4.542 | 2.025 | 1.506 | 0.935 | 1.167 | 0.940 | 1.141 | 0.613 |
| AUPC in (%) | 54.174 | 59.865 | 64.407 | 66.432 | 67.938 | 68.874 | 70.041 | 70.981 | 72.122 | 72.735 |

Table 3.4: Rank profiles for BPS in summer term 2017.

| Rank | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Prob match(%) | 53.973 | 5.725 | 4.538 | 2.053 | 1.529 | 0.931 | 1.181 | 0.948 | 1.150 | 0.610 |
| AUPC in (%) | 53.973 | 59.697 | 64.236 | 66.289 | 67.818 | 68.748 | 69.929 | 70.877 | 72.027 | 72.637 |

Table 3.5: Rank profile BRSD(1000000) in summer term 2017.

The second application in the winter term 17/18 included 1736 students and 66 courses. Overall, we had a list of 20,845 different bundles for the winter term. Again, BPS achieved better results than BRSD in all metrics (see Table Table 3.6). In the BPS outcome 89% of the students receive an assignment ranked in their top ten while in BRSD 88.9% receive such an outcome (see Table 3.7 for BPS and Table 3.8 for BRSD with 1 mio. permutations of the students). The computation times were again very low. BPS required 0.382 seconds, but the lottery algorithm around 30 minutes due to the higher number of bundles generated in the winter term.

The third application in the winter 18/19 term included 1683 students and 68 courses. Overall, we had a list of 27,677 different bundles for the winter term. Again, BPS achieved better results than BRSD in all metrics, apart from AUPCR (see Table 3.9). In the BPS outcome 87.27% of the students receive an assignment ranked in their top ten while in BRSD 87.21% receive such an outcome (see Table 3.10 for BPS and Table 3.11 for BRSD with 1 mio. permutations of the students). The computation times were again very low. BPS required 0.42 seconds and the lottery algorithm around 30 minutes.

### Envy

Our experiments in the summer and the winter term confirm the theoretical result that BPS is (strongly) *envy-free*. BRSD is neither weakly nor strongly envy-free. In the summer term, 1064 students do not fulfill the envy-freeness condition (see

| Metric | BPS | BRSD(1000) | BRSD(1000000) |
|---|---|---|---|
| exp rank | 1.97372 | 1.9784 | 1.97873 |
| exp size | 1603.01 | 1601.03 | 1600.84 |
| prob match | 0.923394 | 0.922253 | 0.922142 |
| AUPCR | 0.889512 | 0.888184 | 0.888058 |
| weak envy | 0 | 427 | 451 |
| strong envy | 0 | 1050 | 1202 |

Table 3.6: Summary statistics for the winter term 2017/2018.

| Rank | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Prob match(%) | 73.596 | 7.083 | 3.392 | 1.660 | 1.041 | 0.698 | 0.465 | 0.447 | 0.366 | 0.299 |
| AUPC in (%) | 73.596 | 80.678 | 84.070 | 85.730 | 86.772 | 87.470 | 87.935 | 88.381 | 88.747 | 89.047 |

Table 3.7: Rank profiles for BPS in winter term 2017/2018.

| Rank | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Prob match(%) | 73.452 | 7.046 | 3.382 | 1.673 | 1.040 | 0.704 | 0.486 | 0.443 | 0.358 | 0.307 |
| AUPC in (%) | 73.452 | 80.497 | 83.879 | 85.553 | 86.593 | 87.297 | 87.783 | 88.226 | 88.584 | 88.891 |

Table 3.8: Rank profile BRSD(1000000) in winter term 2017/2018

| Metric | BPS | BRSD(1000) | BRSD(1000000) |
|---|---|---|---|
| exp rank | 2.55189 | 2.56632 | 2.56692 |
| exp size | 1549.07 | 1548.19 | 1548.67 |
| prob match | 0.92042 | 0.919899 | 0.920181 |
| AUPCR | 0.880628 | 0.879742 | 0.895603 |
| weak envy | 0 | 395 | 491 |
| strong envy | 0 | 1034 | 1243 |

Table 3.9: Summary statistics for the winter term 2018/2019.

| Rank | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Prob match(%) | 67.837 | 7.938 | 4.235 | 2.279 | 1.768 | 0.814 | 0.757 | 0.705 | 0.395 | 0.546 |
| AUPC in (%) | 67.837 | 75.775 | 80.010 | 82.289 | 84.057 | 84.871 | 85.628 | 86.334 | 86.728 | 87.274 |

Table 3.10: Rank profiles for BPS in winter term 2018/2019.

| Rank | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Prob match(%) | 67.723 | 7.972 | 4.248 | 2.280 | 1.768 | 0.820 | 0.755 | 0.693 | 0.403 | 0.546 |
| AUPC in (%) | 67.723 | 75.695 | 79.944 | 82.224 | 83.992 | 84.812 | 85.567 | 86.261 | 86.664 | 87.210 |

Table 3.11: Rank profile BRSD(1000000) in winter term 2018/2019

Definition 3.15), from which 381 students do not even fulfill the weak envy-freeness condition (see BRSD(1000000) in Table 3.3). Similarly, for the winter term 17/18 1202 students do not $SD$-prefer their outcome over the outcomes of every other student, and 451 of those students even prefer an outcome of another student (see BRSD(1000000) in Table 3.6). The results for the winter term 18/19 confirm this observation (see Table 3.9).

### 3.4.3 The Lottery of the Summer Term Instance

We already have discussed that we still have to decompose the solution of BPS into a lottery over integral solutions, to choose a deterministic allocation. This subsection presents exemplary with the Data from summer term 2017, how such a lottery is structured, and how significant the problem of overallocation is in practice.

Figure 3.9 shows the lottery resulting from decomposing the BPS solution described in Table 3.3 into a lottery over approximative feasible integral solutions via Algorithm 3.5. We see that most solutions are close to the fractional solution in terms of number of allocated students (remember: the size of the BPS solution is 1086.58).

One interesting question is how the solutions with a bigger size differ from the matchings with a lower number of allocated students. We computed the average ranks of the deterministic solutions and compared them with the size of the particular matchings.

Figure 3.10 shows the distribution of the different matchings in the lottery with respect to size and average rank. In the first fourth of the x-axis (the size of the matching) the variance is high for the average rank but the pattern gets clearer for a size higher than 1150 – there is a trade-off between size of the matching and the average rank of the allocated students.

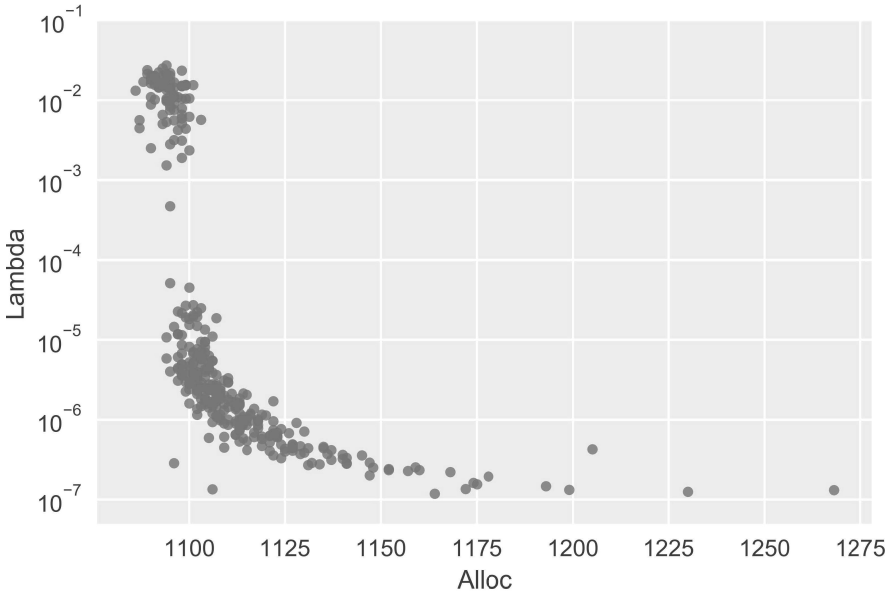


Figure 3.9: The lottery: Probabilities ($\lambda$) and size of the different deterministic matchings returned by Algorithm 3.5.

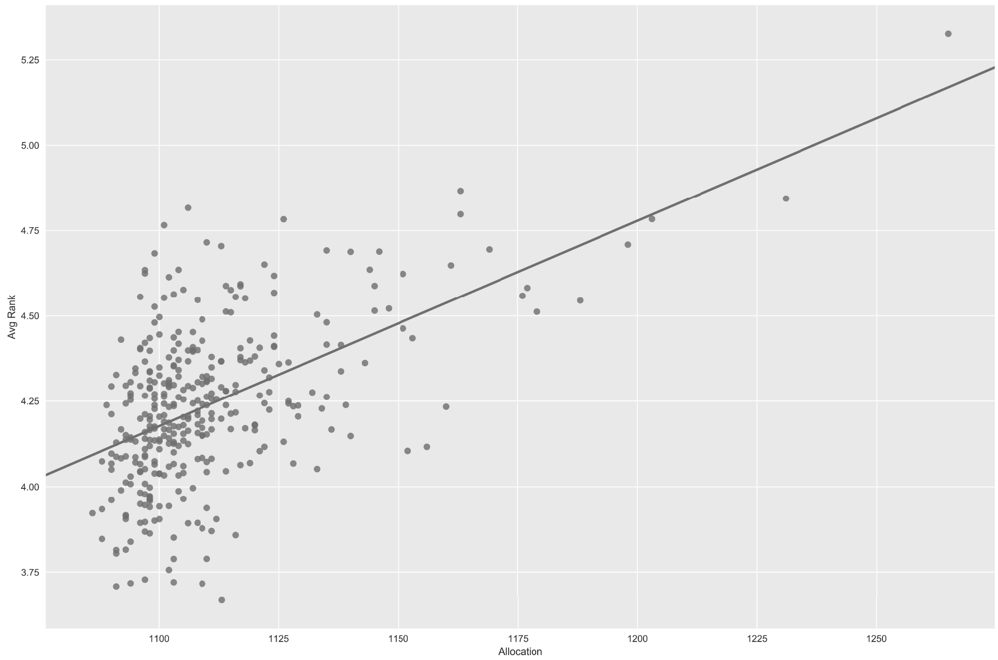


Figure 3.10: Average rank vs. size of the matchings in the lottery.

In Section 3.2.3 and Section 3.2.4 we discussed that the capacity constraints of the courses might be violated. In the reported instance, $\ell = 4$. Hence, the worst case violation of these constraints is 3.

For the computation, we run the lottery with $\varepsilon = 2.0$. Let $e_{x^{(k)},L}$ be the number of goods that experienced a supply violation by $L$ units in the (integral) solution $x^{(k)}$, $\lambda^{(k)}$ the probability of matching $x^{(k)}$ and

$$E_L(Z) = \sum_{x^{(k)} \in Z} \lambda^{(k)} \cdot e_{x^{(k)},L} \tag{3.6}$$

shows how often an overallocation of exact $L$ seats occurs in the set of the lottery $Z$ on average. With the settings mentioned above we receive:

$$E_1(X) = 5.36 \qquad E_2(X) = 0.64 \qquad E_3(X) = 0.07$$

We see that an overallocation by 3 seats rarely happens. Even a violation of 2 seats occurs on average only in 0.64 of the 67 courses. An overallocation of 1 seat occurs on average in 5.36 courses. In the actual application this overallocation did not even require special procedures and course organizers could typically accommodate one or two more students without problems.

The violations also barely change for results with $\varepsilon = 1.0$:

$$E_1(X) = 5.23 \qquad E_2(X) = 0.9 \qquad E_3(X) = 0.06$$

We informed the course organizers before the matching, that small violations of the capacities are possible and no one had a problem with that. If the capacities of some courses were tight, one could solve the problem, by defining a smaller capacity for those courses. Theoretically one had to reduce the capacities by $\ell - 1$ (3 in our case). However, our empirical results suggest, that a reduction of one seat should be sufficient to ensure a feasible allocation with a high probability. However, the reduction of the capacities comes of cost of a lower efficiency of the matching in general.

### 3.4.4 Survey Results

After the students were assigned to the tutor groups and the courses started, we conducted a survey among the students using a 5-point Likert scale (1 = strongly agree, 2 = agree, 5 = strongly disagree). 169 students out of 1736 students participated in the survey in the winter term and we report their responses in Table 3.12. Note that the students were exposed to FCFS in other semesters and now participated in BPS, which allowed them to compare both mechanisms.

Students were not forced to participate and we made clear that the feedback was used for research purposes only. The responses indicate that the majority of the students responding found the system easy to use and that they could express their preferences well. More than 50% agreed (2) or strongly agreed (1) to questions 1 to 6. A majority also considers the system as fair (question 7), but almost 22% of the respondents also disagreed to this statement. Note that students might have had an understanding of fairness that is different from envy-freeness or equal treatment of equals. For example, some students felt that in FCFS they could improve their assignment by making sure that they are among the first to register. This was perceived as fair as the additional effort would lead to higher chances of getting their best allocation, compared to those students who were not as involved.

| | **Question** | **1** | **2** | **3** | **4** | **5** |
|---|---|---|---|---|---|---|
| 1 | I had no problems to select my time ranges in the weekly schedule | 34.9 | 34.9 | 11.8 | 9.5 | 8.9 |
| 2 | The ranking of the generated sets of time-slots was easy | 26.6 | 26.6 | 18.9 | 14.8 | 13.0 |
| 3 | The instructions on the matching system were sufficient | 25.4 | 37.3 | 18.3 | 10.1 | 8.9 |
| 4 | The generated sets of tutorial groups met my expectations | 37.9 | 27.8 | 10.1 | 9.5 | 14.8 |
| 5 | I was able to express my preferences on sets of tutor groups well | 42.6 | 24.9 | 13.6 | 7.7 | 11.2 |
| 6 | I consider the way bundles are allocated through the matching system as fair | 32.5 | 27.2 | 18.3 | 5.9 | 16.0 |
| 7 | I am satisfied with the matching outcome | 45.0 | 17.0 | 9.5 | 6.5 | 21.9 |
| 8 | I felt like I had control over my schedule | 29.0 | 18.9 | 13.0 | 17.2 | 21.9 |
| 9 | I was expressing my preferences truthfully | 72.4 | 13.4 | 4.2 | 3.6 | 6.5 |
| 10 | I was strategically hiding some of my most preferred time-slots | 5.3 | 4.7 | 8.3 | 13.6 | 68.0 |
| 11 | I was strategically hiding some of my least preferred time-slots | 16.0 | 12.4 | 16.0 | 12.4 | 43.2 |

Table 3.12: Survey results, values in %

62.1% of the respondents were satisfied with the outcome (agreed or strongly agreed), while 28.4% were not. It is unclear how those students who did not respond perceived the outcome, but there is a tendency that students who are unhappy with the outcome rather respond than students who got a high ranked bundle. Hence, the sample of students who respond might be slightly biased towards unsatisfaction. The ranking and profile information reported earlier provides alternative information about satisfaction of students with the outcome.

85.8% of the respondents reported that they were expressing their preferences truthfully in BPS (agreed or strongly agreed), while around 10.1% did not (disagreed or strongly disagreed). 10% were also indicating that they were hiding some of their most preferred time-slots, while even 28.4% agreed or strongly agreed to the statement that they were hiding some of their least preferred time-slots.

Still, the fact that a significant part of the students indicate that they did not report preferences truthfully is a tangible difference to FCFS. In FCFS, students only provide their single best bundle at the point in time when they log in. This is simple, intuitive, and obviously strategy-proof. This property has to be traded off against the envy-freeness in BPS.

In a final question students were asked whether they prefer FCFS or BPS: 106 students (62,7%) preferred BPS, while 63 (37.3%) preferred FCFS. To understand the concerns of those students who preferred FCFS, it is useful to look at the written comments. Some students who provided comments were unhappy with the outcome, others were unhappy about the effort to rank-order their bundles.

### 3.4.5 Discussion of Differences

The results from our field experiments and the survey reveal a number of interesting insights. Overall, BPS dominates BRSD on all metrics from our empirical evaluation in all field studies. It has a better average rank, a higher average size and a higher probability of matching, and it does not exhibit envy. However, the differences in average rank, average size, and the profile are small, which is interesting given the fact that only a small number of preferences per student are considered via BRSD (FCFS).

There are a number of reasons that help explain the close performance of BPS and BRSD in these metrics. First, Che and Kojima (2010) find that random serial dictatorship and probabilistic serial become equivalent when the market becomes large, i.e. the random assignments in these mechanisms converge to each other as the number of copies of each object type grows, and the inefficiency of RSD becomes small. Our empirical results suggest that differences might also be small in large combinatorial assignment markets with limited complementarities.

Second, ordinal preferences do not allow to express the intensity of preferences. Suppose there are two students who both prefer course $c_1$ to $c_2$, each having one course seat only. No matter who gets course $c_1$, the average rank and size of the matching as well as the profile will be the same even though one student might desperately want to attend $c_1$, while the second student only has a mild preference for $c_1$. Without cardinal information about the intensity of a preference the differences in aggregate metrics can be small.

Third, an earlier comparison of FCFS with a deferred acceptance algorithm by Diebold et al. (2014) also showed that FCFS yields surprisingly good results. While the average rank of FCFS was worse, the size of the matching resulting from FCFS was significantly larger compared to that from the deferred acceptance algorithm. For the combinatorial assignment problem, BPS actually had a larger average size than FCFS in all studies. For applications of matching in practice it is important to understand these trade-offs.

## 3.5 Combinatorial Assignment Problems with Cardinal Utilities

In this section we introduce an additional application that can be modeled as combinatorial assignment problem. However, here the agents are able to express cardinal utilities over different bundles. We first motivate and model the problem, and show how the approach of Nguyen et al. (2016) can be applied in this setting.

Afterwards we discuss potential efficiency losses if the capacities of the objects are tight.

### 3.5.1 Efficient Coordination in Retail Logistics without Money

According to a survey among more than 500 transportation companies in Germany, 18% of them have an average waiting time of more than two hours and 51% have an average waiting time between one to two hours at each warehouse (Bundesverband Güterkraftverkehr Logistik und Entsorgung e.V., 2013). Such waiting times are a significant problem for carriers and warehouse operators. A recent study among 778 truck drivers by the German Federal Office for Transportation reports that the waiting times even increased in the past 5 years (Bundesamt für Güterverkehr, 2018). In another study, the German Federal Office for Transportation describes the uncoordinated arrivals of trucks as the main reasons for waiting times (Bundesamt für Güterverkehr, 2011). Adding capacity with additional loading docks at warehouse sites requires substantial investments and can also be infeasible in urban areas.

Overall, the lack of coordination causes substantial inefficiencies in retail transportation logistics. The carriers decentrally solve vehicle routing problems and compute optimal routes, but they do this in an uncoordinated manner. The warehouses face capacity planning problems for their loading docks because of the random carrier arrival. If all information about supplier preferences for different routes and warehouse capacities was available, then a central clearing house (potentially organized via a booking platform) could select routes and allocate time-slots to carriers such that waiting times are minimized.

Some retailers use First-Come First-Served (FCFS) time-slot management systems and charge a fixed price for each slot. However, the adoption is low as margins for carriers are low and they are not willing to pay for reservations. Moreover, the simple FCFS mechanism collects only little information about carrier preferences (a single bundle of time-slots on a route) such that one cannot expect an efficient allocation of the available capacities at the warehouses. However, in most cases there is not even an FCFS mechanism in place that would alleviate the long waiting times that arise (Bundesverband Güterkraftverkehr Logistik und Entsorgung e.V., 2013). Karaenke et al. (2018) analyze auction mechanisms, but the potentially high payments of carriers for the reservations again constitute a barrier to adoption in the field. So, the question we ask is, whether efficient coordination can also be facilitated without payments by the carriers.

We consider a single-period coordination problem with warehouses $K$, carriers $I$ (agents), and intra-day time-slots $T$. The locations of warehouses and carriers are given within the transportation network with known (average) travel distances and travel times. The service capacity of warehouses (loading docks) is modeled as a multidimensional knapsack problem. In each time-slot $t \in T$ each warehouse $k \in K$ has a capacity of $c_k = \left(c_{k1}, \ldots, c_{k|T|}\right)$. That is, warehouse $k$ can service up to $c_{kt}$ trucks in time-slot $t$. We call a pair $o = (k, t)$ of a warehouse and a time-slot an *object* with capacity $q_o$, and define $O$ as the set of all objects.

Carriers have to deliver freight to a warehouse, pick it up there, or both. We assume that each carrier has a truck with sufficient capacity to fulfill the orders. The truck starts at the depot and returns to the depot again after (un)loading his freight at the retailers' warehouses. Within the reserved time-slots a carrier can (un)load his freight.

The carriers $i \in I$ have valuations (cardinal preferences) $v_{ib}$ for bundles $b \in B \subseteq \{0,1\}^{|O|}$ of objects represented as vectors where $b_o = 1$ if object $o$ is in the bundle. A bundle $b$ encodes the sequence of visited warehouses and the respective time-slots. Carriers are allowed to submit preferences for as many bundles they want, i.e., they can express preferences for alternative routes and corresponding time-slots.

Similar to Section 3.2 we can model the winner determination problem (WDP) of the coordinator as an IP. The only difference is that we now have cardinal utilities and therefore also can formulate a cardinal objective function.

$$\begin{aligned}
\underset{\text{s.t.}}{\text{Max}} \quad & \sum_{i \in I, b \in B} v_{ib} x_{ib} && && \text{(WDP)}\\
& \sum_{i \in I, b \in B} b_o x_{ib} \leq q_o && \forall o \in O && \text{(supply)}\\
& \sum_{b \in B} x_{ib} \leq 1 && \forall i \in I && \text{(demand)}\\
& x_{ib} \in \{0,1\} && \forall i \in I, b \in B && \text{(binary)}
\end{aligned}$$

The objective is to maximize the sum of valuations of the accepted bundles in (WDP), i.e., to maximize the social welfare. The (supply) constraint ensures that the warehouse capacities are not exceeded for allocated bundles for each time-slot. Constraint (demand) ensures that each carrier wins at most one bundle. Carriers who won one of their tours have reservations for the respective time-slots at the loading docks, while losing carriers have to queue for service with lower priority. This is a weighted set packing problem, which is known to be $\mathcal{NP}$-hard.

According to Theorem 3.23 one only needs to find an envy-free fractional solution to (WDP) to construct a mechanism with desireable properties. A fractional solution to (WDP) can easily be found by solving the LP-relaxation of this IP. However, this solution might not be envy-free in general. Therefore one has to add additional constraints to ensure envy-freeness. An agent $i \in I$ envies another agent $j \in I$ if $i$ prefers the assignment of $j$ over his own assignment. We formalize this as a linear inequality: agent $i$ envies agent $j$ iff

$$\sum_{b \in B} v_{ib} x_{ib} < \sum_{b \in B} v_{ib} x_{jb}. \qquad \text{(envy)}$$

With this we can introduce the (no envy)-constraint for every pair of agents $i, j \in I,\ i \neq j$:

$$\sum_{b \in B} v_{ib} \left(x_{ib} - x_{jb}\right) \geq 0 \quad \forall i, j \in I. \qquad \text{(no envy)}$$

With these additional constraints, one can define a mechanism to solve the coordination problem with cardinal utilities: The first step is to solve the problem

$$x^* = argmax \left\{ \sum_{i,b} v_{ib} x_{ib} \mid (supply), (demand), (no\ envy), x_{ib} \in \mathbb{R}_{\geq 0} \right\}.$$

Because of the (demand)-constraints every variable $x_{ib}$ is in $[0, 1]$ and the sum over all variables referring to the same agent is not greater than 1. Hence, we can interpret the single variables as probabilities and the fractional solution $x^*$ as a random matching. Now we can use Algorithm 3.5 to create a lottery over integral solutions respecting (demand) and the relaxed version of (supply). This mechanism is referred to MAXCU (maximizing cardinal utilities) introduced by Nguyen et al. (2016). Let $k = max \left\{size(b) \mid b \in Acc_I\right\}$ be the size of the biggest acceptable bundle. With Theorem 3.23 we get directly:

**Theorem 3.28: MAXCU.** *MAXCU returns a lottery that is envy-free and efficient in expectation, asymptotically strategy-proof, and that over-allocates each object $o \in O$ by at most $k - 1$.*

### 3.5.2 Capacity Reduction

Even if overallocation of more than one unit rarely happens, there might be applications where even small violations of the capacity constraints are not permitted. For those applications Nguyen et al. (2016) propose to reduce the capacities respectively while calculating the fractional solution. That is, we decompose the smaller $x^*$ and therefore ensure that even if the worst case violation happens, all capacities are respected in all integer solutions. However, a capacity reduction affects the efficiency. Therefore we analyze a multiplicative and an additive capacity adjustment for MAXCU in this section.

We consider the allocation problem (WDP). Let $k = max\left\{size(b) \mid b \in Acc_I\right\}$, the size of the biggest acceptable bundle, be at most as high as the minimal capacity, i.e. $k \leq q_{min}$. That is, we consider problems with limited complementarities.

For the *additive adjustment* we subtract $k-1$ from each capacity $q_o$. Since $k \leq q_{min}$ the LP-relaxation is still solvable. For the *multiplicative adjustment* we need to multiply a factor $\overline{k_o}$ to every $q_o$ such that $q_o \cdot \overline{k_o} + k - 1 \leq q_o$ for every object. That is, $\overline{k_o} \leq {}^{q_o-k+1}/_{q_o} = 1 - {}^{k-1}/_{q_o}$ for every object $o \in O$. To ease the analysis we use an uniform factor for the whole IP, i.e.

$$\overline{k} = min\left\{\overline{k_o} \mid o \in O\right\} = 1 - \frac{k-1}{q_{min}}. \tag{3.8}$$

To estimate the potential efficiency loss due to the capacity reduction, we use duality theory. Let $OPT_D$ be the optimal dual objective value without any adjustments and $OPT_{WDP}$ the optimal primal objective value. Since changes on the right hand side of the primal program only affect the coefficients of the dual objective function, but not the dual feasible space, we can estimate the new optimal dual objective value:

$$OPT_D \geq \overline{OPT_D} = \sum_{i \in I} y_i + \sum_{o \in O} q_o \cdot \overline{k} \cdot y_o \geq \overline{k}\left(\sum_{i \in I} y_i + \sum_{o \in O} q_o \cdot y_o\right) = \overline{k} \cdot OPT_D \tag{3.9}$$

Every feasible solution of the dual describes an upper bound for the primal objective and because of strong duality the optimal solutions of the dual and primal are equal. Hence, we can conclude for the objective value with multiplicative adjustments ($\overline{OPT_{WDP}}$) using (3.9):

$$\overline{k} \cdot OPT_D = \overline{k} \cdot OPT_{WDP} \leq \overline{OPT_{WDP}} \leq OPT_{WDP} = OPT_D \tag{3.10}$$

Finally we compare the additive and the multiplicative adjustment. Let $\widehat{OPT_D} = \sum_{i \in I} y_i + \sum_{o \in O}(q_o - k + 1) \cdot y_o$ be the optimal dual objective value of the problem with additive adjustments. $\widehat{OPT_D}$ and $\overline{OPT_D}$ only differ in the coefficients of the *supply*-variables.

$$\begin{aligned}
\widehat{OPT_D} &= \sum_{i\in I} y_i + \sum_{o\in O} (q_o - k + 1)\cdot y_o = \sum_{i\in I} y_i + \sum_{o\in O} (q_o - (k-1))\cdot y_o \\
&= \sum_{i\in I} y_i + \sum_{o\in O} \left(1 - \frac{k-1}{q_o}\right)\cdot q_o \cdot y_o = \sum_{i\in I} y_i + \sum_{o\in O} \overline{k_o}\cdot q_o\cdot y_o \\
&\geq \sum_{i\in I} y_i + \sum_{o\in O} \overline{k}\cdot q_o\cdot y_o = \overline{OPT_D}
\end{aligned} \tag{3.11}$$

That is, the problem with individual multiplicative adjustments $\overline{k_o}$ for every $o \in O$ leads to the same objective value as with additive adjustments. However, the problem with uniform multiplicative adjustments leads to a lower objective value. With (3.10) we get:

$$\overline{k}\cdot OPT_{WDP} \leq \overline{OPT_{WDP}} \leq \widehat{OPT_{WDP}} \leq OPT_{WDP} \tag{3.12}$$

Hence, if even small capacity violations are not permitted and the minimal capacities are at least as high as the maximal bundle size, a possible approach is to subtract the worst case violation, $k-1$, from each capacity constraint. We leave open an empirical analysis of the efficiency loss and how one can improve the approach by subtracting a smaller amount, such that the capacity constraints are respected (only) with high probability.

# 4 Truthfulness in Advertising? Approximation Mechanisms for Knapsack Bidders [12]

In auction theory, bidders are traditionally modeled as payoff-maximizing individuals via a quasilinear utility function (Krishna, 2009; Milgrom, 2004). If bidders have such a quasilinear utility function the Vickrey-Clarke-Groves (VCG) mechanism is the unique mechanism to implement maximum welfare in dominant strategies (Green and Laffont, 1979). This specific assumption on the utility function is important, because there exist no dominant-strategy incentive-compatible mechanisms for general valuations (Gibbard, 1973; Satterthwaite, 1975). However, there are markets where pure payoff-maximization is not the right assumption. For example, a large number of papers on digital advertising auctions suggest that automated bidders in such auctions rather maximize value subject to a budget constraint as in a knapsack problem (Feldman et al., 2008; Berg et al., 2010; Zhou et al., 2008; Lee et al., 2013; Chen et al., 2011; Zhang et al., 2014), which differs from payoff maximization assumed in a quasilinear utility function.

In such display ad auctions individual user impressions on a website are auctioned off. Typically, the advertising firm provides an intermediary, a demand-side platform (DSP), with a budget for the overall campaign and a value or willingness-to-pay for individual impressions. This value might be the profit from selling a product, and a DSP is not allowed to bid beyond, because it would incur a loss even

[12] This chapter is based on Bichler and Merting (2018). All sections are mainly identical to the respective parts of Bichler and Merting (2018).

if the product got sold. The DSP provides autonomous agents bidding on behalf of the advertiser. The advertising firm (or client of the DSP) typically considers the overall budget $c_i$ as sunk cost devoted to a campaign. In other words, the task of the DSP is to invest the campaign budget such that the advertiser $i$'s sum of valuations of won items is maximized and the total payment for those items is not higher than the budget constraint. We refer to such bidders as *knapsack bidders* or bidders having a *knapsack utility function* and we will introduce them more formally in Definition 4.1. In essence, quasilinear utility is a suitable model when money continues to retain its value into the future. But if money has no value after the current time period, then it makes sense to maximize value (rather than surplus), subject to a budget constraint. Therefore, in business contexts where a budget is an *authorized maximum* that a business unit may charge back to the firm, but the firm has not given that sum of money to the business unit, *knapsack utility* is an appropriate model. In this chapter, we focus on this knapsack utility model and ask whether there are truthful mechanisms maximizing welfare if bidders have such utility functions.

Knapsack utility functions are not only adequate for online markets where supply or demand arrive dynamically over time, but also for conventional offline markets where supply and demand are all present at the time when the market is cleared. For example, bidders (advertisers) on a *TV advertising market* have similar characteristics, but ad slots in the program of a TV station are typically sold for certain periods of time such as every week, rather than dynamically whenever a new impression arrives (Goetzendorff et al., 2015; Nisan et al., 2009). Media agencies bidding in such auctions can also be modeled as knapsack bidders. Maximizing value subject to a budget constraint is actually wide-spread and also assumed in classic micro-economic demand theory (Mas-Colell et al., 1995, p. 50). So, our analysis might be relevant for other domains as well, but we motivate the model assumptions primarily from the literature on advertising markets.

Mechanism design theory typically imposes incentive-compatibility and individual rationality as constraints such that the resulting mechanism provides incentives to bid truthfully and participants do not make a loss (Borgers et al., 2015). To implement a social choice function means to define a mechanism where truthful bidding is an equilibrium. Ideally, the equilibrium solution concept is prior-free, i.e., an agent does not need assumptions about the type distribution. Dominant strategy equilibria for deterministic mechanisms, and truthfulness-in-expectation (TIE) for randomized mechanisms are two such solution concepts used in this chapter.

Unfortunately, welfare maximization with quasilinear bidders appears to be an exceptional environment where the social choice function (i.e., welfare maximization) can be implemented in dominant strategies via the VCG mechanism. Also, randomized mechanisms for general quasilinear preferences which are truthful-in-expectation typically draw on the VCG payment rule (Lavi and Swamy, 2011). However, the VCG payment rule would not incentivize truthful bidding when bidders have a knapsack utility function as such bidders consider their budget as sunk cost and do not value the discount.

A few papers study mechanism design with non-quasilinear utility models. Kazumura and Serizawa (2016) shows that there is no multi-object auction mechanism for heterogenous goods that is dominant strategy incentive compatible and Pareto-efficient, even if only one bidder has multi-unit demands. Similarly, Baisa (2017) shows that if bidders have multi-dimensional types, there is no mechanism that satisfies (1) individual rationality, (2) dominant strategy incentive compatibility, (3) ex-post Pareto efficiency, and (4) weak budget balance for homogenous goods. The utility functions studied in this new line of mechanism design literature are general. For example, Baisa (2017) only assumes that a bidder's demand for the good increases as her wealth increases for a constant price level and some level of risk aversion.

Utility functions with more specific assumptions might allow for more positive results. Fadaei and Bichler (2017b) introduce a model of *value bidders* who maximize the value of packages (bundles) of items $J$ for which they are given financial limits $v_i(S)$ reflecting their valuation that they must not overbid. The utility function of a value bidder is $u_i(S) = v_i(S)$ if $p_i(S) \leq v_i(S)$, and $u_i(S) = -\infty$ otherwise, with $S \subseteq J$. The authors maximized the total bidder valuations and show that with a truthful and deterministic mechanism for value bidders only an $n$-approximation can be achieved, where $n$ is the number of bidders. For randomized mechanisms they describe a complex mechanism with a $\mathcal{O}(\sqrt{m})$ approximation ratio, where $m$ is the number of items.

*Knapsack bidders* are an important special case of value bidders where bidders only have additive valuations up to an overall budget constraint they are given. They want to maximize the total value for the budget they invest. The main difference from value bidders to knapsack bidders is that the value for bundles of objects is additive, and therefore the results from Fadaei and Bichler (2017b) extend. This assumption allows for more positive results, as we will see later.

Apart from item-level valuations there are a few assumptions where advertising markets in the field differ from the value bidders described in Fadaei and Bichler (2017b): (1) bidders can only bid on objects individually, (2) markets are large with many items and many bidders, and (3) prices are determined by a second-price rule and considered as exogenous random variables by the bidders. Already Roberts and Postlewaite (1976) showed that in large markets the ability of an individual player to influence the market is minimal, so bidders should behave as price-taking agents. This is well reflected in the literature on bidding in display ad auctions (see for example Lee et al. (2013); Chen et al. (2011); Zhang et al. (2014)). Actually, prices for an impression are very volatile across the day and across different types of impressions and therefore very hard to predict in these markets (Ghosh et al., 2009; Cui et al., 2011). We will use the term *advertising model* to refer to large markets with many knapsack bidders and many objects, where bidders consider prices as exogenous and behave as price-takers.

Even in a model where bidders cannot influence prices, truthful mechanisms are difficult to construct as we will show. Bidders have a choice as to which objects they bid on and which objects they do not. This is actually a key strategic decision of DSPs in display ad auctions with millions of impressions per day. Targeting strategies have assumed center stage (Levin and Milgrom, 2010; McAfee, 2011), and they describe this strategic choice of impressions by a bidder (Bergemann and Bonatti, 2011). Cream skimming strategies refer to buying up the best impressions promising the highest value for a particular advertiser, while lemons avoidance refers to strategies avoiding the worst impressions (Abraham et al., 2013). In other words, bidders only bid on high-valued impressions, but they pretend to have no value for low-valued impressions or items, as they promise a lower return on investment. If all bidders only revealed their preferences for a small set of high-valued impressions, this might lead to a few good matches, but would also have a negative impact on efficiency and seller revenues overall (Levin and Milgrom, 2010). Many impressions would remain unsold. Our advertising model is motivated by these real-world observations.

## 4.1 Contributions

In what follows, we want to study if there are truthful mechanisms with a good approximation ratio for knapsack bidders in the advertising model. We differentiate between *offline markets* and *online markets*. In offline markets the auctioneer first collects the bids of all bidders for all items before computing an allocation (and prices). Indeed, bidders are present all the time in online markets too, however, the items show up over time and have to be allocated instantly. That is, at the moment of allocation, the auctioneer has only the information about the bids for prior and current items, but cannot say anything about items appearing in the future.

We focus on offline markets, because if there are no truthful mechanisms with good approximation ratios for this environment, then we also cannot find respective online mechanisms. Truthful online mechanisms are significantly more challenging, as we will discuss in the conclusions. In the *offline model* we actually get a positive result. We leverage insights from matching theory and use randomization in the allocation rule to incentivize truthful bidding. Interestingly, for knapsack bidders there is a randomized 4-approximate mechanism, which is much better than the $n$-approximation for the general case of value bidders, who have a budget for each bundle and cannot overbid.

## 4.2 Related Literature

It is well-known that with general valuations, any non-dictatorial mechanism with at least three possible outcomes is not strategy-proof (Gibbard, 1973; Satterthwaite, 1975). Gibbard (1977) showed that every strategy-proof mechanism is a lottery over deterministic mechanisms each of which either has not more than two alternatives or is dictatorial. Even with more specific assumptions on the utility functions, truthful mechanisms appear to be restricted to sequential dictatorships. For example, a number of papers analyze specific assignment problems and bidders with responsive preferences, and show that only sequential dictatorships are strategy-proof and Pareto-optimal (Svensson, 1999; Papai, 2001; Ehlers and Klaus, 2003; Hatfield, 2009).

Procaccia and Tennenholtz (2009) introduced the technique of welfare approximation as a means to derive truthful approximation mechanisms without money. There have been positive results for environments with limited private information. For example, a few related papers analyze truthful mechanisms without money for a strategic variant of the generalized assignment problem (Dughmi and Ghosh, 2010; Chen et al., 2013; Fadaei and Bichler, 2017a). The assumptions in these models are different in a number of details leading to differences in the algorithms, but they also use approaches from matching and approximation as we do in our model. Singer (2010) describes the case of procurement auctions in which sellers have private costs, and the auctioneer aims to maximize a utility function on subsets of items, under the constraint that the sum of the payments provided by the mechanism does not exceed a given budget. This is also different to our model where bidders have budget constraints.

There have been a number of recent papers about bidders with quasilinear utility functions and a private budget constraint, which is different to the knapsack utility model considered in this chapter, because knapsack bidders do not value residual budget. For example, Dobzinski et al. (2012) showed that truthful and Pareto-optimal mechanisms without positive transfers are impossible with private budget constraints. Other authors have analyzed approximation mechanisms when bidders have quasilinear utility functions with a budget constraint (see for example Ashlagi et al. (2010); Dütting et al. (2015); Talman and Yang (2014)). A knapsack utility function is different from a quasilinear utility function with a budget constraint, and this difference has ample consequences for auction design.

## 4.3 The Model

Let us now introduce knapsack bidders formally. First, we draw connections to the value bidders as they have been analyzed in the literature and show that the welfare of truthful approximation mechanisms can be very low in general. Then we make additional assumptions motivated by advertising markets and show that each of these assumptions impacts the approximation ratios of truthful mechanisms. Although these assumptions are adequate for display ad auctions, we show that only an offline model allows for good approximation ratios.

### 4.3.1 Knapsack Bidders and Value Bidders

We study a market where one seller 0 has to auction off a set of $m$ heterogeneous items (or objects) $J$ to $n$ bidders (or agents) $I$. Each bidder $i \in I$ has a cardinal willingness to pay or value $v_{ij} \in \mathbb{R}_{\geq 0}$ for any item $j \in J$. There is one copy of each item. We describe the vector of valuations $v_{ij}$ of a bidder $i$ as $v_i \in \mathbb{R}^m_{\geq 0}$. The values $v_{ij}$ are normalized with 0 for the empty set. We denote $\mathcal{V} \subseteq \mathbb{R}^{n \times m}_{\geq 0}$ as the set of all possible valuation matrices. In addition to the willingness to pay or value for one item, bidders face an overall budget constraint $c_i$. In contrast to mechanism design with quasilinear utility functions, where utility is defined as valuation minus price of a bundle, we assume a knapsack utility function.

**Definition 4.1: Knapsack Utility.** *Given an allocation matrix* $X \in \mathcal{X} \subseteq \{0,1\}^{n \times m}$*, a price matrix* $P \in \mathcal{P} \subseteq \mathbb{R}^{n \times m}_{\geq 0}$*, and a budget vector* $c \in \mathbb{R}^n_{\geq 0}$*, the utility function*

$$u_i(X, P) = \begin{cases} v_i^\intercal x_i & \text{if } p_i^\intercal x_i \leq c_i \wedge \forall j \in J : p_{ij} x_{ij} \leq v_{ij} \\ -\infty & \text{else} \end{cases}$$

*is called* knapsack utility function*. Bidders with a respective utility function are called* knapsack bidders*.*

In this definition $x_i$ is a binary vector describing the allocation for bidder $i$ where $x_{ij} = 1$ when bidder $i$ wins item $j$. If we assume linear and anonymous prices, then $p_i$ is the vector of payments[13] $p_{ij} = p_j$ for all items $j \in J$. Similar to the literature on quasilinear bidders with budget-constraints (Dobzinski et al., 2008), we assume the budget $c_i$ as exogenously given and focus on the strategies of knapsack bidders. As in the related literature on display ad auctions, bidder $i$ must not bid beyond his value $v_{ij}$ for an item (the item value can be seen as an item-level budget limit). Suppose, an advertiser is selling cameras, which yield a profit of 10$. The expected profit considering the conversion rate of an impression can now be considered as the value $v_{ij}$ of the bidder. An advertiser does not want the DSP to bid more than $v_{ij}$ as this would lead to a loss when selling the product to the end consumer.

Let us now define relevant terms from mechanism design for markets with knapsack bidders. Based on the revelation principle, we focus only on direct revelation mechanisms.

**Definition 4.2: Direct Revelation Mechanism.** *A (direct revelation)* mechanism *comprises an allocation function* $f : \mathcal{V} \rightarrow \mathcal{X}$ *and a vector of payment functions* $p_1, \ldots, p_n$*, where* $p_i : \mathcal{V} \rightarrow \mathbb{R}_{\geq 0}$ *defines the payments of bidder* $i$*.*

We aim for incentive-compatible and prior-free mechanisms, i.e., truthful bidding is an equilibrium even without the availability of prior value distributions.

[13]For now, the payments can be computed by any payment rule, no matter if it is a fixed, first or second price rule (or something else).

**Definition 4.3: Truthfulness.** *Given any matrix of true valuations $V \in \mathcal{V}$ and any bidder $i \in I$ with (not necessary true) valuations $v_i' \in \mathcal{V}_i$ we denote $x = f(v_i, V_{-i})$ and $x' = f(v_i', V_{-i})$ the allocations if $i$ reports $v_i$ and $v_i'$ respectively with $p$ and $p'$ being the payments in the two allocations. A mechanism $(f, P)$ is called* truthful *if $u_i(x, p) \geq u_i(x', p')$.*

We are only interested in mechanisms that satisfy *individual rationality*, i.e., participation in the mechanism never makes the agent worse off. The second-price rule typically used in display ad auctions satisfies this assumption. We talk about *strategy-proofness* if we have a deterministic mechanism and truthful bidding is a dominant strategy. For randomized mechanisms, we mainly focus on *truthfulness-in-expectation* (TIE), which will be defined in Section 4.5.

We study truthful and *welfare maximizing* mechanisms for knapsack bidders. In other words, we want to maximize the sum of utilities of all market participants, i.e., the bidders and the seller. We assume a neutral auctioneer who provides the exchange and aims for welfare maximization for all participants. The buyers are the strategic bidders in this exchange, and we assume that the seller has a value of zero for the items. But the knapsack utility function has a significant impact on the social welfare function. A utilitarian social welfare function has the form $SW = \sum_{i \in I, j \in J} u_{ij}$ (Mas-Colell et al., 1995, p. 827). In our model with knapsack bidders, this translates into $SW = \sum_{j \in J, i \in I} (v_{ij} + p_j) x_{ij}$. In contrast, in the standard quasilinear utility model the prices of the items cancel out.[14] Example 4.4 shows the impact of this change.

| | A | B | $c_i$ |
|---|---|---|---|
| Bidder 1 | 6 | 4 | 4 |
| Bidder 2 | 4 | 3 | 4 |
| Bidder 3 | 3 | 0 | 4 |

Table 4.1: Bidder values and and budgets.

[14]With quasilinear utility in case of assignment ($x_{ij} = 1$) we have for bidders $u_{ij}(X, P) = v_{ij} - p_j$ and for the seller $u_{sj}(X, P) = p_j$. Welfare is calculated as follows: $SW = \sum_{i \in I, j \in J} u_{ij}(X, P) + \sum_{j \in J} u_{sj}(X, P) = \sum_{i \in I, j \in j} (v_{ij} - p_j + p_j) x_{ij} = \sum_{ij} v_{ij} x_{ij}$

**Example 4.4:** *Consider an auction with three bidders having values and budgets like in Table 4.1. A welfare maximizing mechanism that does ignore the sellers revenue within the allocation would allocate item A to bidder 1 with price of* 4 *and item B to bidder 2 with price* 0*. This leads to a welfare of* 9 *for the bidders and a revenue of* 4 *for the seller, i.e., an overall welfare of* 13*. If we consider the sellers revenue already for the allocation, we would assign item B to bidder 1 and A to bidder 2, each at a price of* 3*. Now the bidders have a welfare of* 8 *and the seller a revenue of* 6*, which leads to an overall welfare of* 14*. Hence, it is important for the goal of welfare maximization to consider the sellers' revenue as well as the bidders' values.*

We can interprete knapsack bidders as a special class of value bidders. Unfortunately, we cannot hope for good approximation ratios if value bidders are allowed to submit bids on bundles of objects. The extensive proof for the $n$ approximation for general markets in Fadaei and Bichler (2016) yields that a truthful mechanism can only elicit a single bundle value from each bidder. To avoid arbitrarily low approximation ratios, the auctioneer needs to elicit the valuation for the bundle of all objects from each bidder. The proofs in their model also hold for knapsack bidders who are allowed to bid on bundles of items and pay-as-bid. To achieve better results, we have to make further assumptions about the bidder model.

### 4.3.2 The Advertising Model

Several assumptions from the model with value bidders do not carry over to advertising auctions in the field, and the results with value bidders might be too pessimistic. We now introduce the *advertising model*, which adds three assumptions:

i) no bidding on bundles is possible, and

ii) the market is large with many many bidders ($n \to \infty$), and

iii) (linear) prices $p_j$ for each item are considered exogenous by bidders.

All three assumptions are met in display ad auctions where bidders cannot bid on bundles of impressions and the auctioneer uses a second-price payment rule per impression. The idea that market size ease incentive problems goes back a long time with some of the earliest contributions being Roberts and Postlewaite (1976). Jackson and Manelli (1997) show conditions under which, as the size of the market increases, all the market-clearing prices and allocations based on reported bids approximate the competitive equilibria of the market with true valuations. Therefore, assumption 3 follows from the literature on large markets with many bidders. Also, the literature on bidding strategies referenced in the introduction considers prices as an exogenous variable (Chen et al., 2011; Zhang et al., 2014).

Let us outline the allocation problem of the auctioneer in an offline auction. This *allocation problem* (AP) can be described as a binary program where bidder $i$ with a budget constraint $c_i$ has a value $v_{ij}$ for each item $j$ with an anonymous and linear price $p_j$.

$$
\begin{aligned}
\underset{\text{s.t.}}{\text{Max}} \; & \sum_{i \in I} \sum_{j \in J} (v_{ij} + p_j)\, x_{ij} && && (\mathbf{AP}) \\
& \sum_{i \in I} x_{ij} \leq 1 && \forall j \in J && \text{(Supply)} \\
& \sum_{j \in J} p_j x_{ij} \leq c_i && \forall i \in I && \text{(Budget)} \\
& x_{ij} \in \{0, 1\} && \forall i \in I, j \in J && \text{(Binary)}
\end{aligned}
$$

The optimal solution to AP maximizes welfare in a market if the auctioneer had access to the true valuations and budget constraints of the bidders. We refer to AP-LP as the LP relaxation of AP. A feasible assignment may allocate a subset of items $S$ to bidder $i$ such that $\sum_{j \in S} p_j \leq c_i$ (Budget). A feasible assignment may assign each item at most once (Supply). Note that this is a slightly modified version of the generalized assignment problem (the parameter $p_j$ in the objective function), which has an integrality gap of 2 (Shmoys and Tardos, 1993). As indicated, we consider the price as an exogenous variable for our strategic analysis and study whether truthful and prior-free mechanisms are possible in this environment. In an online market the price $p_j$ could be determined via a second-price rule as in display ad auctions. In an offline market, the auctioneer might simply determine a fixed price per type of object based on the bid distribution.

The key decision of a knapsack bidder in the (offline and online) advertising model is whether to bid on an item or not at all and to decide what budget he submits to the auctioneer. Indeed, we consider the budget as exogenous given for the bidder (e.g. by a client or a superior), however, it is still private information that the auctioneer does not know in advance.

## 4.4 Deterministic Approximation Mechanisms

The purpose of this section is to highlight the difficulties in designing deterministic approximation mechanisms with a good approximation ratio. The discussion is useful to introduce our main results in Section 4.5.

Example 4.5 is sufficient to show that a truthful and welfare maximizing mechanism is not possible independent of the payment rule that we use in the presence of knapsack bidders.

**Example 4.5:** *Consider a market with two bidders and two items. Table 4.2 describes bidder valuations and their budget constraints. The allocation maximizing value is to assign item B to bidder 1 and item A to bidder 2 with a total bidder value of* $1 + 3\varepsilon$. *However, bidder 1 can increase his utility to* $1$, *by bidding on A only and pretending that his value for B is zero. This way the auctioneer allocates A to bidder 1 and B to bidder 2 with a total value of* $1 + 2\varepsilon$. *Thus, there cannot be a deterministic and welfare maximizing mechanism that is also truthful.*

Since there exist for deterministic, truthful, and welfare maximizing mechanisms, we relax the the goal of maximizing welfare and study deterministic approximation mechanisms. A mechanism returns (at most) an $\alpha$-approximation of the optimal if its value is always greater than or equal to $1/\alpha$ times the optimal value ($\alpha \geq 1$).

| | **Valuations** | | |
|---|---|---|---|
| Items | A | B | $c_i$ |
| Bidder 1 | 1 | $4\varepsilon$ | 1 |
| Bidder 2 | $1-\varepsilon$ | $2\varepsilon$ | 1 |

Table 4.2: Bidder valuations, where bidder 1 can shade his value for $B$ to win $A$.

We first give a strategyproof mechanism in Algorithm 4.1 and analyze afterwards how well it performs in terms of welfare.

This serial dictatorship (SD) mechanism sorts the bidders by decreasing value of $min\left\{c_i, \sum_j v_{ij}\right\}$. Usually this is a sorting with respect to the submitted budgets $c_i$, since in general there will be $\sum_j v_{ij} \geq c_i$. But if $c_i > \sum_j v_{ij}$, the budget constraint actually is not necessary for this bidder and we therefore consider for the sorting a *scaled down* budget $\bar{c_i} = \sum_j v_{ij}$ instead. Next the algorithm iteratively chooses bidders with respect to this sorting and computes an assignment out of the remaining items to this bidder that maximizes his sum of valuations subject to the budget constraint. Finally the allocated items will be removed and we choose the next bidder until there is no one left.

**Algorithm 4.1:** SD algorithm for knapsack bidders

**Input :** knapsack bidders $I$ with private budgets $c_i$ and values $V$ for items $J$, prices $p$

$\mathcal{L}$ is a list of bidders sorted by decreasing value of $min\left\{c_i, \sum_j v_{ij}\right\}$.
$s_j = 1, \forall j \in J$
**forall** $i \in \mathcal{L}$ **do**
  $x_i^* = argmax\left\{v_i^\intercal x_i \mid \sum_j p_j x_{ij} \leq c_i,\, x_{ij} \leq s_j,\, x_{ij} \in \{0,1\}\right\}$
  **for** $j$ *with* $x_{ij}^* = 1$ **do** $s_j = 0$

**Output:** assignment $X$

**Lemma 4.6:** *Algorithm 4.1 is a strategy-proof mechanism for knapsack bidders.*

*Proof.* Let us first observe that the only way a knapsack bidder could lie is to submit a wrong budget or to decide not to bid on some items. Because of assumption

2 and 3, bidders can neither influence the market with the value of their bid nor can they know in advance the market price of a special item. That is, submitting a bid $0 < b_{ij} < v_{ij}$ for some item $j \in J$ cannot have any positive effect for bidder $i \in I$. Furthermore a knapsack bidder cannot claim a higher budget $c_i$ and he can also not overbid since this could lead to a utility of $-\infty$. Shading the budget $c_i$ or hiding valuations for items to zero, only decreases their ranking in the algorithm and cannot improve their allocation. Hence, the mechanism is truthful. □

Unfortunatly, Algorithm 4.1 can lead to low welfare in the worst case.

**Lemma 4.7:** *Algorithm 4.1 achieves an approximation ratio not better than* $m = |J|$.

*Proof.* Consider a market with $m+1$ bidders and $m$ items with $p_j \leq 1$ for all $j \in J$. Table 4.3 describes bidder valuations and their budget constraints. In this market, a deterministic and value-maximizing auction would allocate item $j_i$ to bidder $i$. The welfare is $m^2 - \sum_{i=1}^{m} \varepsilon_i$. However, Algorithm 4.1 assigns first the most valued feasible set of items to bidder 0, since his sum of valuations as well as his budget is greater than those of the other bidders. That is, Algorithm 4.1 allocates all items to bidder 0 with a total utility of $m$, which implies an approximation ratio $\geq m$. □

| | **Valuations** | | | | |
|---|---|---|---|---|---|
| Items | $j_1$ | $j_2$ | $\cdots$ | $j_m$ | $c_i$ |
| Bidder 0 | 1 | 1 | $\cdots$ | 1 | $m$ |
| Bidder 1 | $m - \varepsilon_1$ | 0 | $\cdots$ | 0 | $m - \varepsilon_1$ |
| Bidder 2 | 0 | $m - \varepsilon_2$ | $\cdots$ | 0 | $m - \varepsilon_2$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\vdots$ |
| Bidder m | 0 | 0 | $\cdots$ | $m - \varepsilon_m$ | $m - \varepsilon_m$ |

Table 4.3: Bidder valuations.

**Theorem 4.8:** *Algorithm 4.1 is strategy-proof and it achieves an approximation ratio within* $\Theta(m)$.

*Proof.* Lemma 4.6 shows that Algorithm 4.1 is strategy-proof. Based on Lemma 4.7, we know that the approximation ratio for Algorithm 4.1 is within $\Omega(m)$. We only have to show that the approximation ratio is within $\mathcal{O}(m)$.

We relax the algorithm in the *forall*-loop to allow for fractional solutions $x_{ij} \in [0,1]$ and construct a feasible dual solution with a value at most $(m+1)$ times the value obtained by the relaxed algorithm. By calling the weak duality theorem together with the integrality gap of 2 (Shmoys and Tardos, 1993), the claim follows.

Assume $X$ is the outcome of the relaxed Algorithm 4.1. Using $X$ we can construct a feasible solution to the dual of AP-LP.

$$\begin{aligned} \underset{\text{s.t.}}{\text{Min}} \ & \sum_{j=1}^{m} \rho_j + \sum_{i=1}^{n} c_i d_i && \textbf{(AP-LPD)} \\ & \rho_j + p_j d_i \geq v_{ij} + p_j, \ \forall i \in I, j \in J \\ & \rho, d \geq 0 \end{aligned}$$

Observe that in Algorithm 4.1 the bidders are sorted in decreasing order of $min\left\{c_i, \sum_j v_{ij}\right\}$. Wlog. let $i_1, i_2, i_3, \ldots, i_n$ be the order of bidders computed by Algorithm 4.1. Furthermore, the algorithm assigns the current bidder available (fractions of) items $j$ in decreasing order of the density $v_{ij}/p_j$ to maximize his utility. Note, that the price $p_j$ for item $j$ is the same for all bidders $i \in I$ and therefore the ordering with respect to this density is the same as $(p_j+v_{ij})/p_j = v_{ij}/p_j + 1$. Initially, let $\rho = \vec{0}$ and $d = \vec{0}$. If item $j$ gets exhausted by assigning $j$ to bidder $i_k$, set $\rho_j = p_j + max\left\{v_{i_l j} \mid l \geq k\right\}$, the highest valuation for items $j$ over all bidders with a lower rank than $i_k$ plus the price of item $j$. Furthermore, for all bidders $i$ with exhausted budget, set $d_i = 1 + min\left\{v_{ij}/p_j \mid \forall j \in J : x_{ij} > 0\right\}$, that is, one plus the density of the last item assigned to $i$. This satisfies the dual constraints in AP-LPD.

In particular, if the budget of bidder $i$ is exhausted, then for each item $j$ (fractionally) assigned to bidder $i$, either $j$ gets exhausted with this assignment or does not. If $j$ is exhausted, we have $\rho_j \geq v_{ij} + p_j$, and therefore the constraint holds. If $j$ is not exhausted, we have either that $j$ is the last assigned item and therefore

$d_i = {}^{v_{ij}}/_{p_j} + 1$, or $j$ is not assigned to $i$ and therefore $d_i \geq {}^{v_{ij}}/_{p_j} + 1$; and the constraint holds as well. If bidder $i$ has residual budget, every item $j$ that is assigned to it is exhausted by this assignment. That is, we have $\rho_j \geq v_{ij} + p_j$ and the constraint thus holds. For every item $j$ that is not assigned to the bidder but for that the bidder has a positive valuation, we have $\rho_j \geq v_{ij} + p_j$, since the item is exhausted due to the assignment to a bidder $i_k$ with higher rank than $i$, and we have $v_{ij} \leq max\left\{v_{i_l j} \mid l \geq k\right\} = \rho_j - p_j$. In sum, we have constructed a feasible dual solution using $x$, the allocation resulting from the relaxed Algorithm 4.1.

We now bound the value of the dual solution with respect to the primal solution. First, we observe that

$$\sum_{i,j}(v_{ij} + p_j)x_{ij} \geq \sum_{i,j} d_i p_j x_{ij} = \sum_i d_i \sum_j p_j x_{ij},$$

because $d_i$ is only non-zero if the budget of bidder $i$ is exhausted,

$$d_i p_j = \left(1 + min\left\{\frac{v_{ij'}}{p_{j'}} \mid \forall j' \in J : x_{ij'} > 0\right\}\right) p_j \leq \left(\frac{v_{ij}}{p_j} + 1\right) p_j = v_{ij} + p_j$$

for items with $x_{ij} > 0$, and $(p_j + v_{ij})\, x_{ij} = d_i p_j x_{ij} = 0$ for items $j$ with $x_{ij} = 0$. Second, we show that $m \sum_{i,j} (p_j + v_{ij})\, x_{ij} \geq \sum_j \rho_j \sum_i x_{ij}$. Now we sum up both inequalities and obtain

$$\begin{aligned}(m+1) \textstyle\sum_{i,j} (v_{ij} + p_j)\, x_{ij} &\geq \textstyle\sum_j \rho_j \sum_i x_{ij} + \sum_i d_i \sum_j p_j x_{ij} \\ &= \textstyle\sum_j \rho_j + \sum_i d_i c_i\end{aligned}$$

Notice, only for items $j$ that get exhausted $(\sum_i x_{ij} = 1)$ we have $\rho_j > 0$ and only for full bidders $\left(\sum_j p_j x_{ij} = c_i\right)$ we have $d_i > 0$. The final term is the value of the dual, the desired conclusion.

It remains to show that $m \sum_{i,j} (v_{ij} + p_j)\, x_{ij} \geq \sum_j \rho_j \sum_i x_{ij}$. Since $v_{ij} \leq c_i$ for all $i \in I$ and $j \in J$, we can observe that $(\rho_j - p_j)\, x_{i_k j} = max\left\{v_{i_l j} \mid l \geq k\right\} x_{i_k j} \leq c_{i_k}$.

This follows from the ordering of the bidders in Algorithm 4.1. For all bidders $i_l$ with a lower rank than bidder $i_k$ we have

$$min\left\{c_{i_l}, \sum_j v_{i_l j}\right\} \leq min\left\{c_{i_k}, \sum_j v_{i_k j}\right\} \leq c_{i_k}.$$

That means either $v_{i_l j} \leq c_{i_l} \leq c_{i_k}$ or $v_{i_l j} \leq \sum_j v_{i_l j} \leq c_{i_k}$ for all items $j$. That is, the assignment of one item to bidder $i_k$ with dual price $\rho_j - p_j$ is always primal feasible as well.[15] The worst case is that bidder $i_k$ gets all $m$ items, but all these items $j$ have a high (dual) price $\rho_j - p_j$ such that $i_k$ can buy at most one of them without violating his budget constraint. Since for all those items $j$ of bidder $i$ $\rho_j - p_j \leq c_i$, we have

$$\begin{aligned} \textstyle\sum_{i,j} v_{ij} x_{ij} &\geq 1/m \textstyle\sum_j (\rho_j - p_j) \sum_i x_{ij} \\ &= 1/m \textstyle\sum_j \rho_j \sum_i x_{ij} - 1/m \sum_j p_j \sum_i x_{ij} \end{aligned}$$

and therefore

$$\begin{aligned} \textstyle\sum_{i,j} (v_{ij} + p_j)\, x_{ij} &\geq \textstyle\sum_{i,j} (v_{ij} + p_j/m)\, x_{ij} \\ &= \textstyle\sum_{i,j} v_{ij} x_{ij} + 1/m \sum_j p_j \sum_i x_{ij} \\ &\geq 1/m \textstyle\sum_j \rho_j \sum_i x_{ij}, \end{aligned}$$

the desired conclusion. □

Algorithm 4.1 is a deterministic and truthful mechanism that yields low welfare. It is straightforward to show that the ratio is tight for a market with two knapsack bidders and two items only.

**Proposition 4.9:** *No strategy-proof and deterministic mechanism for knapsack bidders exists with an approximation ratio better than 2 of the optimal welfare.*

[15] Since prices $p_j$ do not depend on the allocation and are the same for all bidders, a loss in social welfare can only occur due to a sub-optimal assignment of the items to bidders if all items are assigned. Hence, the dual prices $\rho_j - p_j$ can be interpreted as opportunity costs for the bidders.

*Proof.* Consider a market with two bidders and two items. Table 4.4 describes bidder valuations and their budget constraints. In this market, a deterministic and value-maximizing auction would allocate item 2 to bidder 1 and item 1 to bidder 2. The welfare is $2 - 2\varepsilon$. Bidder 1 can shade his bid for item 2 to zero, such that only item 1 would be allocated to him. That is, any strategy-proof mechanism has to assign item 1 to bidder 1 before trying to assign item 2, leading to an approximation ratio of 2. This is true for any price $p_j \leq 1 - \varepsilon$. □

| | **Valuations** | | |
|---|---|---|---|
| Items | 1 | 2 | $c_i$ |
| Bidder 1 | 1 | $1-\varepsilon$ | 1 |
| Bidder 2 | $1-\varepsilon$ | 0 | $1-\varepsilon$ |

Table 4.4: Bidder valuations.

We leave it for future research to find a more precise lower bound on the approximation ratio for truthful and general markets with arbitrary numbers of bidders and items. This might be challenging to show as the general result and the analysis for value bidders suggests (Fadaei and Bichler, 2016). However, a simple example shows that also large markets with knapsack bidders are susceptible to similar types of manipulation as in the proof to Proposition 1 and this can lead to significant welfare losses. Note that the example is a stylized form of cream skimming, which was described as a wide-spread bidding strategy in display ad auctions in the introduction.

**Example 4.10:** *Consider bidder 1 who has a value of \$1 for 100 impressions of type A and a value of \$0.9 for 100 impressions of type B. His budget is \$50. Bidders 2 and 3 have a value of \$0.5 for type A valuations and a budget of \$50. The welfare maximizing allocation would be to allocate A impressions to either bidder 2 or 3, and the B impressions to bidder 1. Welfare would be \$190, the total of \$90 from bidder 1 plus \$50 from bidder 2 plus \$50 for the auctioneer, which is the payment of bidder 2 or 3, if we assume a second-price rule. However, if bidder 1 does not bid on B impressions, she will be awarded the A impressions and the welfare will only be \$150, i.e., \$100 for bidder 1 and \$50 for the auctioneer, which* *is the payment of bidder 1 with a second-price rule. Bidder 1 would increase his utility from \$90 for the B impressions to \$100 for the A impressions.*

## 4.5 Randomized Approximation Mechanisms

We now resort to randomization as a means to improve the approximation guarantees, but first introduce some definitions. Let $\mathcal{A}$ denote a randomized algorithm, which takes instance $(V, c)$ of AP and computes $X \in \mathcal{X}$, an assignment of items to bidders. The assignment is deterministic (each bidder receives a set of items), but algorithm $\mathcal{A}$ is randomized, i.e., $\mathcal{A}$ returns a solution that is randomly chosen according to a probability distribution over feasible assignments. Hence, $\mathcal{A}(V, c) \subseteq \mathcal{X}$ denotes a set of possible feasible solutions. Truthfulness-in-expectation is a solution concept for randomized approximation mechanisms.

**Definition 4.11: Truthfulness-In-Expectation.** *A randomized algorithm $\mathcal{A}$ is said to be* truthful-in-expectation (TIE) *if for $X \in \mathcal{A}(V, c)$ it holds:*

*i) (feasibility)* $\forall j \in J : \mathbb{P}\left(\sum_{i \in I} x_{ij} \leq 1\right) = 1$ *and*
$\forall i \in I : \mathbb{P}\left(\sum_{j \in J} p_{ij} x_{ij} \leq c_i \wedge \forall j \in J : p_{ij} x_{ij} \leq v_{ij}\right) = 1$

*ii) (truthfulness) for any $i$ and $v_i' \in \mathbb{R}^m_{\geq 0}$, it is*
$\mathbb{E}\left[\sum_{j \in J} v_{ij} x_{ij}\right] \geq \mathbb{E}\left[\sum_{j \in J} v_{ij} x'_{ij}\right]$, *where $X' \in \mathcal{A}(v_i' \cup V_{-i}, c)$.*

Note, the expected value of the bidders in both cases are computed with respect to their true valuations. Random Serial Dictatorship (RSD) is a well-known example of a randomized mechanism, but it is easy to see that the welfare of RSD can be arbitrary low. For example, RSD might first assign all items to a bidder with low valuations and sufficient budget, although other bidders have a much higher valuation. Therefore, our objective is to propose a randomized algorithm $\mathcal{A}$ for AP, which is truthful and always returns a feasible assignment whose value approximates the optimal total value as well as possible.

Our technique is a relax-and-round approach with oblivious rounding as it has been used recently in the literature on approximation mechanisms (Lavi and Swamy, 2011; Dughmi and Ghosh, 2010; Fadaei and Bichler, 2017a). We design a *fractionally truthful* approximation algorithm that returns a feasible solution to AP-LP. A fractionally truthful algorithm allocates fractional assignments to bidders $i$, and no bidder can improve its fractional value by an untruthful report. In particular, a fractionally truthful algorithm $\mathcal{A}^F$ takes $(V, c)$ and returns deterministically $X \in \mathcal{X}$, a feasible solution to AP-LP with the following property. For each bidder $i$, if the bidder reports $v'_i \in \mathbb{R}^m_{\geq 0}$, it holds $\sum_{j\in J} v_{ij}x_{ij} \geq \sum_{j\in J} v_{ij}x'_{ij}$, where $X' = \mathcal{A}^F(v'_i \cup V_{-i}, c)$[16]. Next, we round the fractional solution using a special rounding technique, which makes sure that each bidder obtains a fixed fraction of its fractional value in expectation. The *randomized meta-rounding* (Carr and Vempala, 2000) is capable of maintaining this fixed fraction.

To use the randomized meta-rounding, we have to scale down the fractional solution by factor 2, which is the integrality gap of the AP-LP (Shmoys and Tardos, 1993). Assuming $X^*_F = \mathcal{A}^F(V, c)$, the randomized meta-rounding represents $X^*_F/2$ as a convex combination of polynomially-many feasible integer solutions. Looking at the provided convex combination as a probability distribution over integer solutions, we sample a randomized solution $X \in \mathcal{X}$, which is always feasible, and its expected value is $1/2$ of the fractional value of $X^*_F$. This is confirmed by Theorem 4.12, which has been proved in related work by Dughmi and Ghosh (2010).

**Theorem 4.12: (Dughmi and Ghosh, 2010).** *If there exists a fractionally truthful $\alpha$-approximation algorithm for AP, then there exists a TIE $(2\alpha)$-approximation solution for AP.*

Even though, their model is different, we can draw on this theorem. We propose Algorithm 4.2, an adaptation of the Deferred Acceptance Algorithm (DAA) by Gale and Shapley (1962). In Algorithm 4.2 all bidders with a positive budget submit a bid for the item with the highest density $v_{ij}/p_j$ from their preference list (*Pref*) and delete this item from this list. Afterwards, all items $j \in J$ sort their list of not rejected bids $(B_j)$ according to the value of the bids, and accept all (fractional) bids until their supply $(s_j)$ of 1 is exhausted. If the current item $j$

[16] We can write = here since $\mathcal{A}^F$ is deterministic and therefore it returns always the same solution for a fixed instance $\left(\left|\mathcal{A}^F(V, c)\right| = 1\right)$

has sufficient residual supply to store the whole amount of the current bid $b_{ij}$, we assign (temporary) $j$ to $i$ but only that fraction that the bidder is still able to pay (first *if*-condition).

If the supply of $j$ is still not zero, but not high enough to store the whole bid, we assign $i$ only the available fraction of $j$ ($x_{ij} = s_j$). If the supply of $j$ is exhausted, we reject the bid finally (second *if*-condition), since the threshold for assignment weakly increases each iteration. If no bidder can submit a new bid, the algorithm terminates and returns the assignment matrix $X$.

**Algorithm 4.2:** Fractional DAA for Knapsack Bidders

**Input :** knapsack bidders $I$ with private budgets $c_i$ and values $V$ for items $J$, prices $p$

$x_{ij} = 0,\ \forall i \in I, j \in J$
$Pref_i = \{j \mid v_{ij} > 0, j \in J\},\ \forall i \in I$
$B_j = \emptyset,\ \forall j \in J$
**while** $\forall i \in I, c_i > 0 : Pref_i \neq \emptyset$ **do**
  **forall** $i \in I$ *with* $c_i > 0$ **do**
    $j = argmax\{v_{ij}/p_j \mid j \in Pref_i\}$
    $b_{ij} = v_{ij}$
    $Pref_i = Pref_i \setminus j,\ B_j = B_j \cup b_{ij}$
  **forall** $j \in J$ **do**
    sort $B_j$ according to $b_{ij}$, $s_j = 1$
    **forall** $b_{ij} \in B_j$ **do**
      $c_i = c_i + p_j x_{ij},\ x_{ij} = 0$
      **if** $(s_j - min\{1, c_i/p_j\} \geq 0)$ **then**
        $x_{ij} = min\{1, c_i/p_j\}$
        $s_j = s_j - min\{1, c_i/p_j\}$
        $c_i = c_i - min\{p_j, c_i\}$
      **else**
        **if** $(s_j = 0)$ **then** $B_j = B_j \setminus b_{ij}$
        $x_{ij} = s_j$
        $c_i = c_i - p_j s_j,\ s_j = 0$

**Output:** assignment $X$

**Lemma 4.13:** *Algorithm 4.2 is truthful.*

*Proof.* As already discussed in the proof of Lemma 4.6, knapsack bidders only bid zero or their true value. Since bidders propose to items in decreasing order of the density $v_{ij}/p_j$, hiding values can only lead to an assignment to items with a lower density, and therefore decrease the utility of a bidder. Therefore, knapsack bidders have no incentives to hide valuations. Next, we discuss the budget $c_i$. A knapsack bidder must not spend more than $c_i$. If a bidder reports a lower $\hat{c}_i < c_i$ there are two possible cases. If $\hat{c}_i$ is not exhausted during the process, it does not have an impact on the allocation computed by the algorithm. If $\hat{c}_i$ becomes binding, it can only restrict the assignment of additional items to a bidder, which would otherwise have been possible, and therefore decrease the total sum of valuations of a bidder. That is, Algorithm 4.2 is truthful for knapsack bidders. □

**Lemma 4.14:** *Algorithm 4.2 returns a 2-approximation solution to* AP-LP.

*Proof.* Based on the proof of Proposition 4.9, we know that an approximation ratio better than 2 is impossible. We only have to show that it is not worse than 2. For this, we construct a feasible dual solution with a value at most twice the value obtained by the algorithm, then by calling the weak duality theorem, the claim follows. Assume $X$ is the outcome of Algorithm 4.2. Using $X$ we can construct a feasible solution to the dual of AP-LP.

Observe that in Algorithm 4.2 a bidder $i$ proposes items in decreasing order of the density $v_{ij}/p_j$, and item $j$ examines bids in decreasing order of $v_{ij}$. Since $p_j$ is the same for all bidders $i$, the ordering of the bids at item $j$ according to $v_{ij}/p_j$ is the same as to $v_{ij}$ and $v_{ij} + p_j$. Initially, let $\rho = \vec{0}$ and $d = \vec{0}$. If item $j$ gets exhausted, set $\rho_j = v_{i_k j} + p_j$, where $i_k = argmin\,\{v_{ij} \mid \forall i \in I : x_{ij} > 0\}$. Furthermore, for all bidder $i$ with exhausted budget, set $d_i = v_{ij_k}/p_{j_k} + 1$, where $j_k = argmin\,\{v_{ij}/p_j \mid \forall j \in J : x_{ij} > 0\}$ is the last item assigned to bidder $i$. This satisfies the dual constraints in AP-LPD.

In particular, if the budget of bidder $i$ is exhausted, then for each $j$ of bidder $i$, either $j$ gets exhausted with this assignment or does not. If $j$ is exhausted, it is $\rho_j = v_{ij} + p_j$, and therefore the constraint holds. If $j$ is not exhausted, it is $d_i \geq v_{ij}/p_j + 1$ and thus the constraint holds as well. If bidder $i$ has residual budget, every item $j$ which is assigned to it is exhausted by this assignment. Hence, $\rho_j = v_{ij} + p_j$ and the constraint thus holds. For every item $j$ that is not assigned to the bidder but for that the bidder has a positive valuation, it is $\rho_j \geq v_{ij} + p_j$, since the item is exhausted due to another assignment $v_{i'j} \geq v_{ij}$, which is equivalent to $v_{i'j} + p_j \geq v_{ij} + p_j$. In sum, we have constructed a feasible dual solution using $x$, the allocation resulting from Algorithm 4.2.

Now, we bound the value of the dual solution with respect to the primal solution. First, we observe that

$$\sum_{i,j} (v_{ij} + p_j)\, x_{ij} \geq \sum_j \rho_j \sum_i x_{ij},$$

since $\rho_j$ is non-zero only if $j$ is fully assigned. Second,

$$\sum_{i,j} (v_{ij} + p_j)\, x_{ij} \geq \sum_i d_i \sum_j (p_j x_{ij}),$$

because $d_i$ is only non-zero if the budget of bidder $i$ is exhausted, and then it holds

$$d_i p_j \leq \left( \frac{v_{ij}}{p_j} + 1 \right) p_j = v_{ij} + p_j$$

for all $j$ with $x_{ij} > 0$. Now we sum up both inequalities and obtain

$$\begin{aligned} 2 \textstyle\sum_{i,j} (v_{ij} + p_j)\, x_{ij} &\geq \textstyle\sum_j \rho_j \sum_i x_{ij} + \sum_i d_i \sum_j p_j x_{ij} \\ &= \textstyle\sum_j \rho_j + \sum_i d_i c_i \end{aligned}$$

Notice, only for items $j$, which get exhausted ($\sum_i x_{ij} = 1$) we have $\rho_j > 0$ and only for full bidders ($\sum_j p_j x_{ij} = c_i$) we have $d_i > 0$. The final term is the value of the dual, the desired conclusion. □

Finally, we use Theorem 4.12 together with Lemma 4.13 and Lemma 4.14 and obtain our main result.

**Theorem 4.15:** *There exists a* TIE *4-approximation mechanism for the* AP *with strategic bidders.*

# 5 Discussion and Conclusion

This thesis studies allocation problems where the standard assumption that bidders have a quasilinear utility function does not hold. Such problems occur when bidders are budget restricted and they have no benefit from budget not spent (like e.g. in advertising markets), or the bidders are not willing to pay for an allocation (like e.g. in reservation systems), or they are not even able do express valuations for the different objects (e.g. in course assignment). We have analyzed two applications. First, a course allocation problem, where students are allowed to express ordinal preferences over whole schedules. Second, an online advertising problem, where bidders are interested in buying multiple slots for display ads respecting their budget constraints.

Overall, this thesis highlights how randomization can be used to circumvent problems and impossibilities when aiming for efficient and incentive compatible mechanisms, and it shows how to deal with problems, where standard models usually used in the literature are not satisfying the requirements.

Next, we summarize the main findings for these two applications and discuss limitations and possibilities for future research in these areas.

## 5.1 Course Assignment with Preferences over Schedules

We modeled the problem of assigning schedules to students as a matching problem with preferences over bundles. We generalized efficiency criteria and other design desiderata to the problem with bundles, and presented and analyzed different randomized mechanisms to solve the matching problem.

We reported of three large field studies and showed that BPS performs well on a number of additional criteria including average rank, average size, probability of a matching, and the overall profile of ranks assuming a complete, truthful, and strict ranking of all bundles. The matching based on BPS is also more popular than BRSD based on the preferences submitted for BPS. The level of envy in BRSD is significant, even though the size of the bundles that can be submitted is limited to the number of classes (three to four groups per bundle).

The assignment of tutor groups is specific as preferences are mainly about times of the week. The preferred time-slots in a week may differ among students. However, the way how tutor groups should be ordered within these time-slots (e.g. time for breaks) can be described with a few parameters such that it was possible to generate bundles according to a score. The feedback of students was that this automated ranking met their preferences well and we argue that this is a good way to address the missing bids problem in similar applications. In other applications, generating good bundles might not be as straightforward and this will have an impact on efficiency. Compact and domain-specific bid languages have been discussed in the auction literature (Bichler et al., 2011), and they could also be a possibility to allow mechanisms without transfers circumvent the missing bids problem.

This thesis highlights basic trade-offs in market design without money: FCFS can be seen as BRSD, which is ex-post efficient, and obviously strategy-proof and treats students equally. It is also transparent and simple to implement, and to understand for students. BPS is a randomized mechanism that is only weakly strategy-proof, but envy-free and ordinally efficient, which is stronger than ex-post efficiency assuming strict preferences. Note that these properties hinge on the availability of strict preferences over all, exponentially many, bundles.

Even if the missing bids problem can be addressed, two important problems remain: First, in contrast to FCFS, the BPS mechanism is not obviously strategy-proof and a part of the students in the survey already indicated that they either hid their most preferred or least preferred time-slots strategically.[17] Second, the assumption of strict preferences is strong in the presence of exponentially many bundles. Unfortunately, extending PS or BPS to preferences with ties is not without loss. On the one hand, Katta and Sethuraman (2006) extended PS to preferences with indifferences and showed that it is not possible for any mechanism to find an envy-free, ordinally efficient assignment that satisfies even weak strategy-proofness as in the strict preference domain. On the other hand, with indifferences and random tie breaking efficiency cannot be guaranteed. Our preference elicitation technique generated a strict and complete ranking of course bundles based on a few input parameters and is one way to address these issues.

The key difference between BPS and FCFS is the absence of envy. The level of envy in FCFS is significant. Note that it might be even more pronounced if students were allowed to pick larger bundles. Envy-freeness or stability has been raised as one of the arguments why the Gale-Shapley mechanism for simple assignment problems where agents have unit-demand is so successful in practice (Roth, 2002). If the market outcome is not guaranteed to be stable, there might exist agents who have the incentive to circumvent the match. We argue that this property is also important for the assignment of course schedules. If envy-freeness matters, the BPS mechanism has a number of attractive economic properties and is computationally tractable.

[17] Remember that our empirical comparisons are based on the preferences reported in BPS. A part of these preferences might not have reflected the true preferences of participants, and the comparison might be biased towards BPS.

## 5.2 Non-Quasilinear Bidder Models in Advertising Markets

Auctions in advertising markets are often based on the standard assumption of quasilinear utility functions. However, as the literature about bidding strategies suggests, quasilinearity does not appear to be an adequate model for bidding agents in these markets. Rather, agents try to maximize the total sum of valuations subject to an overall budget that is devoted to a campaign. Mechanism design tries to devise auctions where truthful bidding is an equilibrium. This is typically hard to achieve in environments where bidders do not have quasilinear utility functions and their types are multi-dimensional. We have explored possibilities for truthful approximation mechanisms with knapsack utility functions in large advertising markets. This thesis highlights the types of assumptions, which allow for truthful mechanisms in an offline market and presents a 4-approximate mechanism. The mechanism is randomized and complex, which suggests that truthfulness is hard to achieve in practical environments. The results provide motivation to analyze non-truthful mechanisms for advertising markets.

We focused on welfare maximization, not on revenue maximization. In the field display ad exchanges are competing and if one seller extracts revenue and the other aims for welfare maximization, bidders will move to the second one. Most display-ad auctions use a second-price principle and argue that this would be truthful and welfare maximizing (if bidders have quasilinear utilities for individual impressions and no budget constraints). In any case, welfare maximization (efficiency) is an important first step, but revenue maximization is an interesting avenue to explore in this environment for the future.

It is important to study the approximation ratios of truthful online mechanisms as, for example, display ads need to be sold one at a time, and allocation and pricing need to be decided dynamically. Online mechanisms are often analyzed in the adversarial setting with items arriving in the worst possible order. The approximation ratios achieved in Section 4.5 for the offline environment can be seen as lower bounds on what can be achieved with truthful online mechanisms. Online mechanism design is, however, much less well understood even with quasilinear utility functions, and truthful mechanisms were designed only recently. Typically, the literature assumes unit-demand bidders and only a few papers deal with multi-unit demand assuming homogeneous goods and quasilinear utility functions. Devanur et al. (2018) analyzed online mechanisms for quasilinear bidders who have preferences for multiple heterogeneous items arriving over time. They show that there is no deterministic truthful and individually rational mechanism that gets any finite approximation factor to the optimal social welfare, even if payments can be computed after all items arrived. We assume that the design of truthful online mechanisms will be as challenging for knapsack bidders. There are many possible online environments one can explore and details in the assumptions matter. We leave the analysis of truthful online mechanisms for knapsack bidders for future research.

# References

Abdulkadiroğlu, A., Pathak, P., Roth, A., 2009. Strategy-proofness versus efficiency in matching with indifferences: Redesigning the NYC high school match. The American Economic Review 88 (5), 1954–1978.

Abdulkadiroğlu, A., Pathak, P., Roth, A. E., Sonmez, T., 2006. Changing the Boston school choice mechanism. Tech. rep., National Bureau of Economic Research.

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

Abraham, I., Athey, S., Babaioff, M., Grubb, M., 2013. Peaches, lemons, and cookies: designing auction markets with dispersed information. In: EC. pp. 7–8.

Ashlagi, I., Braverman, M., Hassidim, A., 2014. Stability in large matching markets with complementarities. Operations Research 62 (4), 713–732.

Ashlagi, I., Braverman, M., Hassidim, A., Lavi, R., Tennenholtz, M., 2010. Position auctions with budgets: Existence and uniqueness. The BE Journal of Theoretical Economics 10 (1).

Azevedo, E. M., Budish, E., 2017. Strategy-proofness in the large. Tech. rep., National Bureau of Economic Research.

Aziz, H., Brandt, F., Brill, M., 2013. The computational complexity of random serial dictatorship. Economics Letters 121 (3), 341–345.

Baisa, B., 2017. Auction design without quasilinear preferences. Theoretical Economics 12 (1), 53–78.

Berg, J., Greenwald, A., Naroditskiy, V., Sodomka, E., 2010. A knapsack-based approach to bidding in ad auctions. In: ECAI. Vol. 215. pp. 1013–1014.

Bergemann, D., Bonatti, A., 2011. Targeting in advertising markets: implications for offline versus online media. The RAND Journal of Economics 42 (3), 417–443.

Bichler, M., Goeree, J., Mayer, S., Shabalin, P., 2014. Simple auctions for complex sales: Bid languages and spectrum auction design. Telecommunications Policy 38, 613–622.

Bichler, M., Merting, S., 2018. Truthfulness in advertising? approximation mechanisms for knapsack bidders. European Journal of Operational Research 270 (2), 775 – 783.

Bichler, M., Merting, S., Uzunoglu, A., 2018. Assigning course schedules: About preference elicitation, fairness, and truthfulness. CoRR abs/1812.02630.
URL http://arxiv.org/abs/1812.02630

Bichler, M., Merting, S., Uzunoglu, A., 2019. Matching with bundle preferences: Tradeoff between fairness and truthfulness. In: Ludwig, T., Pipek, V. (Eds.), Proceedings of the 14th International Conference on Wirtschaftsinformatik. Universität Siegen, pp. 378 – 392.

Bichler, M., Schneider, S., Guler, K., Sayal, M., 2011. Compact bidding languages and supplier selection for markets with economies of scale and scope. European Journal on Operational Research 214, 67–77.

Birkhoff, G., 1946. Three observations on linear algebra. Univ. Nac. Tucumán. Revista A 5, 147–151.

Bogomolnaia, A., Moulin, H., 2001. A new solution to the random assignment problem. J. Economic Theory 100 (2), 295–328.

Borgers, T., Krahmer, D., Strausz, R., 2015. An introduction to the theory of mechanism design. Oxford University Press, USA.

Bowman, E. H., 1963. Consistency and optimality in managerial decision making. Management Science 9 (2), 310–321.

Brandt, F., 2017. Rolling the dice: Recent results in probabilistic social choice. Trends in Computational Social Choice.

Brandt, F., Hofbauer, J., Suderland, M., 2017. Majority graphs of assignment problems and properties of popular random assignments. In: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems. International Foundation for Autonomous Agents and Multiagent Systems, pp. 335–343.

Budish, E., 2011. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. Journal of Political Economy 119 (6), 1061–1103.

Budish, E., Cachon, G. P., Kessler, J. B., Othman, A., 2017. Course match: A large-scale implementation of approximate competitive equilibrium from equal incomes for combinatorial allocation. Operations Research 65 (2), 314–336.

Budish, E., Cantillon, E., 2012. The multi-unit assignment problem: Theory and evidence from course allocation at harvard. American Economic Review 102 (5), 2237–71.

Budish, E., Kessler, J., 2017. Can agents "report their types"? an experiment that changed the course allocation mechanism at wharton. Chicago Booth Working Paper.

Bundesamt für Güterverkehr, 3 2011. Sonderbericht zur Situation an der Laderampe.

Bundesamt für Güterverkehr, 2 2018. Abläufe an den Laderampen verbessern.

Bundesverband Güterkraftverkehr Logistik und Entsorgung e.V., 10 2013. Jahresbericht 2012/2013.

Carr, R., Vempala, S., 2000. Randomized metarounding. In: Proceedings of the thirty-second annual ACM symposium on Theory of computing. ACM, pp. 58–62.

Cechlárová, K., Eirinakis, P., Fleiner, T., Magos, D., Mourtos, I., Potpinková, E., 2014. Pareto optimality in many-to-many matching problems. Discrete Optimization 14, 160–169.

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

Chen, N., Gravin, N., Lu, P., 2013. Truthful generalized assignments via stable matching. Mathematics of Operations Research 39 (3), 722–736.

Chen, Y., Berkhin, P., Anderson, B., Devanur, N. R., 2011. Real-time bidding algorithms for performance-based display ad allocation. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 1307–1315.

Clarke, E., 1971. Multipart pricing of public goods. Public Choice 11, 17–33.

Cui, Y., Zhang, R., Li, W., Mao, J., 2011. Bid landscape forecasting in online ad exchange marketplace. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 265–273.

Day, R., Milgrom, P., 2008. Core-selecting package auctions. International Journal of Game Theory 36, 393–407.

Devanur, N. R., Sivan, B., Syrgkanis, V., 2018. Truthful multi-parameter auctions with online supply: an impossible combination. In: Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms. SIAM, pp. 753–769.

Diebold, F., Aziz, H., Bichler, M., Matthes, F., Schneider, A., 2014. Course allocation via stable matching. Business & Information Systems Engineering 6 (2), 97–110.

Diebold, F., Bichler, M., 2017. Matching with indifferences: A comparison of algorithms in the context of course allocation. European Journal of Operational Research 260 (1), 268–282.

Dobzinski, S., Lavi, R., Nisan, N., 2008. Multi-unit auctions with budget limits. In: Foundations of Computer Science. Philadelphia, PA, pp. 260–269.

Dobzinski, S., Lavi, R., Nisan, N., 2012. Multi-unit auctions with budget limits. Games and Economic Behavior 74 (2), 486 – 503.

Dughmi, S., Ghosh, A., 2010. Truthful assignment without money. In: Proceedings of the 11th ACM conference on Electronic commerce. ACM, pp. 325–334.

Dütting, P., Henzinger, M., Starnberger, M., 2015. Auctions for heterogeneous items and budget limits. ACM Transactions on Economics and Computation 4 (1), 4.

Economic Sciences Prize Committee of the Swedish Academy of Sciences, October 2012. Scientific background on the Sveriges Riksbank prize in economic sciences in memory of Alfred Nobel 2012: Stable allocations and the practice of market design. Nobel Media AB.

Ehlers, L., Klaus, B., 2003. Coalitional strategy-proof and resource-monotonic solutions for multiple assignment problems. Social Choice and Welfare 21 (2), 265–280.

Fadaei, S., Bichler, M., 2016. Truthfulness and approximation with value-maximizing bidders. In: Symposium on Algorithmic Game Theory. Springer, pp. 566–576.

Fadaei, S., Bichler, M., 2017a. Generalized assignment problem: Truthful mechanism design without money. In: Operations Research Letters. Vol. 45. Elsevier, pp. 72–76.

Fadaei, S., Bichler, M., 2017b. Truthfulness with value-maximizing bidders: On the limits of approximation in combinatorial auctions. European Journal of Operational Research 14 (1), 1–11.

Feldman, J., Muthukrishnan, S., Nikolova, E., Pal, M., 2008. A truthful mechanism for offline ad slot scheduling. In: Monien, B., Schroeder, U. (Eds.), Algorithmic Game Theory. Vol. 4997 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 182–193.

Fischer, G. W., 1972. Four methods for assessing multi-attribute utilities: An experimental validation. Tech. rep., Michigan Univ Ann Arbor Engeneering Psychology Lab.

Gabow, H. N., 1974. Implementation of algorithms for maximum matching on nonbipartite graphs.

Gale, D., Shapley, L. S., 1962. College admissions and the stability of marriage. American Mathematical Monthly 69 (1), 9–15.

Gale, D., Sotomayor, M., 1985. Ms. machiavelli and the stable matching problem. The American Mathematical Monthly 92 (4), 261–268.

Ghosh, A., Rubinstein, B. I., Vassilvitskii, S., Zinkevich, M., 2009. Adaptive bidding for display advertising. In: Proceedings of the 18th international conference on World wide web. ACM, pp. 251–260.

Gibbard, A., 1973. Manipulation of voting schemes: a general result. Econometrica 41, 587–601.

Gibbard, A., 1977. Manipulation of schemes that mix voting with chance. Econometrica: Journal of the Econometric Society, 665–681.

Goetzendorff, A., Bichler, M., Shabalin, P., Day, R. W., 2015. Compact bid languages and core pricing in large multi-item auctions. Management Science 61 (7), 1684–1703.

Green, J., Laffont, J., 1979. Incentives in public decision making. North Holland, Amsterdaml.

Groves, T., 1973. Incentives in teams. Econometrica: Journal of the Econometric Society, 617–631.

Gusfield, D., Irving, R. W., 1989. The stable marriage problem: structure and algorithms. MIT press.

Hanley, J. A., McNeil, B. J., 1982. The meaning and use of the area under a receiver operating characteristic (roc) curve. Radiology 143 (1), 29–36.

Hashimoto, T., 2018. The generalized random priority mechanism with budgets. Journal of Economic Theory 177, 708–733.

Hatfield, J., 2009. Strategy-proof, efficient, and nonbossy quota allocations. Social Choice and Welfare 33 (3), 505–515.

Hylland, A., Zeckhauser, R., 1979. The efficient allocation of individuals to positions. Journal of Political economy 87 (2), 293–314.

Jackson, M. O., Manelli, A. M., 1997. Approximately competitive equilibria in large finite economies. Journal of Economic Theory 77 (2), 354–376.

Karaenke, P., Bichler, M., Minner, S., 2018. Coordination is hard: Electronic auction mechanisms for increased efficiency in transportation logistics. Management Science forthcoming.

Katta, A.-K., Sethuraman, J., 2006. A solution to the random assignment problem on the full preference domain. Journal of Economic theory 131 (1), 231–250.

Kavitha, T., Mestre, J., Nasre, M., 2011. Popular mixed matchings. Theoretical Computer Science 412 (24), 2679–2690.

Kazumura, T., Serizawa, S., 2016. Efficiency and strategy-proofness in object assignment problems with multi-demand preferences. Social Choice and Welfare 47 (3), 633–663.

Kojima, F., Manea, M., et al., 2010. Incentives in the probabilistic serial mechanism. Journal of Economic Theory 145 (1), 106.

Krishna, A., Ünver, M. U., 2008. Research note—improving the efficiency of course bidding at business schools: Field and laboratory studies. Marketing Science 27 (2), 262–282.

Krishna, V. (Ed.), 2009. Auction Theory. Elsevier Science, San Diego, CA, USA.

Lavi, R., Swamy, C., 2011. Truthful and near-optimal mechanism design via linear programming. Journal of the ACM (JACM) 58 (6), 25.

Lee, K.-C., Jalali, A., Dasdan, A., 2013. Real time bid optimization with smooth budget delivery in online advertising. In: Proceedings of the Seventh International Workshop on Data Mining for Online Advertising. ACM, p. 1.

Lee, Y. E., Benbasat, I., 2011. Research note—the influence of trade-off difficulty caused by preference elicitation methods on user acceptance of recommendation agents across loss and gain conditions. Information Systems Research 22 (4), 867–884.

Levin, J., Milgrom, P., 2010. Online advertising: Heterogeneity and conflation in market design. The American Economic Review 100 (2), 603–607.

Li, S., 2017. Obviously strategy-proof mechanisms. American Economic Review 107 (11), 3257–87.

Manlove, D. F., 2013. Algorithmics of Matching Under Preferences. Vol. 2 of Series on Theoretical Computer Science. World Scientific.

Mas-Colell, A., Whinston, M. D., Green, J. R., et al., 1995. Microeconomic theory. Vol. 1. Oxford university press New York.

McAfee, R. P., 2011. The design of advertising exchanges. Rev Ind Organ 39, 169–185.

Merting, S., Karaenke, P., Bichler, M., 2016. Strategy-proof assignment of bundles with ordinal preferences: An application in retail logistics. In: Nissen, V., Stelzer, D., Straßburger, S., Fischer, D. (Eds.), Proceedings of the Multikonferenz Wirtschaftsinformatik (MKWI) 2016: Band I. Universitätsverlag Ilmenau, pp. 205–216.

Milgrom, P., 2004. Putting auction theory to work. Cambridge University Press.

Milgrom, P., 2010. Simplified mechanisms with applications to sponsored search and package auctions. Games and Economic Behavior 70 (1), 62–70.

Müller, A., Stoyan, D., 2002. Comparison methods for stochastic models and risks. Vol. 389. Wiley New York.

Nguyen, T., Peivandi, A., Vohra, R., 2016. Assignment problems with complementarities. Journal of Economic Theory 165, 209 – 241.

Nisan, N., Bayer, J., Chandra, D., Franji, T., Gardner, R., Matias, Y., Rhodes, N., Seltzer, M., Tom, D., Varian, H., et al., 2009. Google's auction for tv ads. Automata, Languages and Programming, 309–327.

Othman, A., Papadimitriou, C., Rubinstein, A., 2016. The complexity of fairness through equilibrium. ACM Transactions on Economics and Computation (TEAC) 4 (4), 20.

Papai, S., 2001. Strategyproof and nonbossy multiple assignments. Journal of Public Economic Theory 3, 257–271.

Procaccia, A. D., Tennenholtz, M., 2009. Approximate mechanism design without money. In: Proceedings of the 10th ACM conference on Electronic commerce. ACM, pp. 177–186.

Pycia, M., Troyan, P., 2018. Obvious dominance and random priority. Working Paper.

Roberts, D. J., Postlewaite, A., 1976. The incentives for price-taking behavior in large exchange economies. Econometrica: Journal of the Econometric Society, 115–127.

Roth, A. E., November 1982. The economics of matching: Stability and incentives. Mathematics of Operations Research 7 (4), 617–628.

Roth, A. E., 1984. The evolution of the labor market for medical interns and residents: a case study in game theory. Journal of political Economy 92 (6), 991–1016.

Roth, A. E., 1986. On the allocation of residents to rural hospitals: a general property of two-sided matching markets. Econometrica: Journal of the Econometric Society, 425–427.

Roth, A. E., 1990. New physicians: a natural experiment in market organization. Science 250 (4987), 1524–1528.

Roth, A. E., 1991. A natural experiment in the organization of entry-level labor markets: regional markets for new physicians and surgeons in the united kingdom. The American economic review, 415–440.

Roth, A. E., 2002. The economist as engineer: Game theory, experimentation, and computation as tools for design economics. Econometrica 40 (4), 1341–1378.

Roth, A. E., Xing, X., 1994. Jumping the gun: Imperfections and institutions related to the timing of market transactions. The American Economic Review, 992–1044.

Santos, B. L. D., Bariff, M. L., 1988. A study of user interface aids for model-oriented decision support systems. Management Science 34 (4), 461–468.

Satterthwaite, M. A., 1975. Strategy-proofness and arrow's conditions. existence and correspondence theorems for voting procedures and social welfare functions. Journal of Economic Theory 10.

Shapley, L. S., Scarf, H., 1974. On cores and indivisibility. Journal of Mathematical Economics 1, 23–37.

Shmoys, D. B., Tardos, É., 1993. An approximation algorithm for the generalized assignment problem. Mathematical Programming 62 (1-3), 461–474.

Singer, Y., 2010. Budget feasible mechanisms. In: Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on. IEEE, pp. 765–774.

Sng, C. T. S., 2008. Efficient algorithms for bipartite matching problems with preferences. Ph.D. thesis, University of Glasgow.

Sönmez, T., Ünver, M. U., 2010. Course bidding at business schools. International Economic Review 51 (1), 99–123.

Svensson, L.-G., 1999. Strategy-proof allocation of indivisible goods. Social Choice and Welfare 16 (4), 557–567.

Talman, A., Yang, Z., 2014. An Efficient Multi-Item Dynamic Auction with Budget Constrained Bidders. Vol. 44.

Vickrey, W., 1961. Counterspeculation, auctions, and competitive sealed tenders. The Journal of finance 16 (1), 8–37.

Von Neumann, J., 1953. A certain zero-sum two-person game equivalent to the optimal assignment problem. Contributions to the Theory of Games 2, 5–12.

Zhang, W., Yuan, S., Wang, J., 2014. Optimal real-time bidding for display advertising. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '14. ACM, New York, NY, USA, pp. 1077–1086.

Zhou, L., October 1990. On a conjecture by gale about one-sided matching problems. Journal of Economic Theory 52 (1), 123–135.

Zhou, Y., Chakrabarty, D., Lukose, R., 2008. Budget constrained bidding in keyword auctions and online knapsack problems. In: Internet and Network Economics. Springer, pp. 566–576.