Assigning Course Schedules: About Preference Elicitation, Fairness, and Truthfulness

Sören Merting
Technical University of Munich, soeren.merting@in.tum.de

Martin Bichler
Technical University of Munich, bichler@in.tum.de

Aykut Uzunoglu
University of Augsburg, aykut.uzunoglu@wiwi.uni-augsburg.de

Assigning Course Schedules: About Preference Elicitation, Fairness, and Truthfulness

Completed Research Paper

Martin Bichler
Technical University of Munich
bichler@in.tum.de

Sören Merting
Technical University of Munich
soeren.merting@in.tum.de

Aykut Uzunoglu
University of Augsburg
aykut.uzunoglu@wiwi.uni-augsburg.de

Abstract

Most organizations face distributed scheduling problems where private preferences of individuals matter. Course assignment is a widespread example arising in educational institutions and beyond. Often students have preferences for course schedules over the week. First-Come-First-Served (FCFS) is the most widely used assignment rule in practice, but it is inefficient and unfair. Recent work on randomized matching suggests an alternative with attractive properties – Bundled Probabilistic Serial (BPS). A major challenge in BPS is that the mechanism requires the participants' preferences for exponentially many schedules. We describe a way to elicit preferences reducing the number of required parameters to a manageable set. We report results from field experiments, which allow us to analyze important empirical metrics of the assignments compared to FCFS. These metrics were central for the adoption of BPS at a major university. The overall system design yields an effective approach to solve daunting distributed scheduling tasks in organizations.

Keywords: *Computational Social Science, Course Assignment, Preference Elicitation, Field Study*

Introduction

Scheduling is ubiquitous in organizations and often involves the assignment of people with preferences to scarce resources or tasks. Examples include the allocation of surgeons to hospital operating rooms, workers to shifts or projects over time, or students to different courses on a weekly schedule. There is a huge literature on optimization formulations and decision support for rosting, staffing, or scheduling which is typically analyzing such problems from the point of view of a single decision maker or planner (Burke et al. 2004; Cardoen et al. 2010; Ernst et al. 2004). For most of these problems peoples' private preferences matter. Surgeons have preferences for certain time slots for their operations (Cardoen et al. 2010), workers have preferences for times of the day and free time (Burke et al. 2004), and students have preferences for particular course schedules across the week (Budish et al. 2017). In practice, schedules are often based on first-come first-served procedures (as in course scheduling) or people are simply asked for their preferences that are then considered in staff scheduling and rostering, for example. However, these mechanisms are neither incentive compatible, nor efficient or fair.

Mechanism design investigates economic mechanisms which set incentives for participants to reveal their preferences truthfully. Auctions are known as incentive-compatible and efficient economic mechanisms for resource allocation problems (Klemperer 1999; Milgrom 2004). However, the scheduling problems discussed earlier do not lend themselves to auctions. First, monetary transfers are not allowed or desired in these environments. Second, cardinal utility and interpersonal utility comparison is too much to assume and we can only hope to get ordinal preferences. Third, preferences are private and we want to have mechanisms that incentivize students to reveal these preferences truthfully. Fourth, the allocation of course schedules

is a computationally hard (NP-hard) problem and for realistic problem sizes an exact solution might not be tractable.

The theory of matching under preferences analyzes mechanisms which set incentives for participants to reveal their preferences truthfully without monetary transfers. It is a successful application of computational economics and more broadly computational social sciences. The field has drawn significant academic attention after the Nobel Prize in Market Design in 2012 with important applications in the matching of residents to hospitals, in school choice, or in kidney exchanges (Roth 2015). Until recently, this theory was largely restricted to problems where participants have unit demand (e.g. a single course seat). However, new approaches have been developed which allow for more complex preferences such as preferences for packages of course seats or schedules and other complex constraints (Budish et al. 2013; Kamada and Kojima 2017; Nguyen et al. 2016). Interestingly, randomization turned out to be a powerful tool to achieve mechanisms with good properties in the presence of complex preferences.

This theory provides foundations for new types of distributed information systems respecting private preferences of participants. While these systems can be used for various types of decision support in organizations, we illustrate how these economic design goals come to bear in course assignment. The assignment of students to course schedules can be seen as a wide-spread coordination problem arising in all larger educational institutions and beyond. We introduce new mechanisms to build effective information systems for the assignment of course schedules to students and report results from large-scale field experiments. More importantly, we show that information systems design, in particular the elicitation of user preferences, is of central importance in the implementation of such new forms of distributed information systems. While the topic is very close to others in information systems design such as online auctions, multi-unit auctions, or combinatorial auctions (Adomavicius et al. 2012; Adomavicius and Gupta 2005; Bapna et al. 2007; Bapna et al. 2010; Bapna et al. 2003; Goetzendorff et al. 2015; Scheffel et al. 2011), matching without money has not received much attention yet. Overall, we aim to extend the information systems literature on decision support and design science to distributed resource allocation and coordination problems without money (Banker and Kauffman 2004).

Course Assignment

Course assignment is a ubiquitous problem at universities and an excellent example of scheduling problems with preferences as they arise in other domains as well. While some universities use matching mechanisms such as the deferred acceptance algorithm (Diebold et al. 2014; Gale and Shapley 1962) or course bidding (Krishna and Ünver 2008; Sönmez and Ünver 2010), in most cases scarce course seats are allocated via first-come first-served (FCFS). Although many course assignment problems are similar to the widely studied school choice problems (Abdulkadiroğlu and Sönmez 2003; Ashlagi and Shi 2016) 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).

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 (TUM) that we will discuss. Since 2014 the TUM is using the deferred acceptance algorithm for two-sided matching problems and random serial dictatorship for one-sided matching problems to assign seminars or practical courses to students. Every semester about 1500 students are being matched centrally (Diebold et al. 2014). For seminars and practical courses students need to get assigned one out of many courses offered per semester. In many situations students' preferences do not only concern a single course. For example, in the first three semesters there are large courses with hundreds of students. 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, a student might want to have two tutorials in the morning and one after lunch on a particular day to reduce her commute time, and she 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 that need to be considered, which makes it a combinatorial assignment problem. Similar problems can be found in many organizations.

A first and seminal approach to address this fundamental problem, the *approximate competitive equilibrium from equal incomes mechanism* (A-CEEI), was published by Budish (2011). In A-CEEI students report their complete preferences over schedules of courses, the mechanism assigns a budget of fake money to each student that she can use to purchase packages (or schedules) of courses. Then an optimization-based mechanism computes approximate competitive equilibrium prices, and the student is allocated her most preferred bundle given the preferences, budgets, and prices. It is well known that serial dictatorships are the only strategy-proof and efficient mechanisms for multi-unit and also combinatorial assignment problems (Ehlers and Klaus 2003; Pápai 2001). A-CEEI is relaxing design goals such as strategy-proofness and envy-freeness to approximate notions, which makes it a remarkable and practical contribution to a fundamentally hard problem. The mechanism has been shown to be approximately strategy-proof, approximately envy-free, and Pareto efficient. 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 price vector and course allocation exists that satisfies all capacity constraints. Second, the problem of computing the allocation problem 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) (Bichler and Goeree 2017; Milgrom 2010). The latter is a general problem in CAP not restricted to A-CEEI, which we will discuss in much more detail below.

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 2001b), 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 packages. In course assignment problems the size of the packages is typically small (e.g., packages with three to four tutor groups) compared to the capacity of the courses or tutor groups (around 30 seats or more). There is no need for prices or budgets, and computationally the mechanism runs in polynomial time, which is important for large instances of the course allocation problem that can frequently be found. This makes BPS an interesting approach to many scheduling problems that appear in practice.

Contributions

We 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 and we provide a practical approach that addresses the combinatorial explosion of possible packages for many applications. Theoretical contributions of assignment mechanisms largely focus on incentive-compatibility, envy-freeness (as a form of fairness), and efficiency as primary design desiderata. Based on a large-scale field experiment, we are able to report properties of matchings such as their size, their average rank, the probability of matching, the profile, and the popularity compared to FCFS. These properties are of central importance for the choice of mechanisms and the design of respective information systems.

Implementing and testing new IS artifacts in organizations is challenging and we are grateful for the possibility to run a large-scale field experiment at the (anonymized) university. 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 1439 students in the summer term 2017 to 67 tutor groups for 4 classes and the assignment of 1778 students in the winter term 2017/2018 to 66 tutor groups for 4 classes using BPS.[1] Based on this data the department has adopted the new mechanism for good.

[1]Not all students submitted a non empty preference list. Therefore, we consider in our evaluation only 1415 students in summer term and 1736 students in winter term.

In such a large application one cannot 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 (as a proxy for FCFS) on equal footing.

FCFS only collects limited information about the preferences of participants, a single package only. Mechanisms for the combinatorial assignment problem allow participants to specify preferences for all possible packages. We argue that this provides a much better comparison of the average performance of FCFS compared to just a single instance of FCFS.

However, BPS requires a ranking of exponentially many schedules, which is a prerequisite for the economic properties described. Preference elicitation and user interface design have long been a topic in IS research (Lee and Benbasat 2011; Santos and Bariff 1988). 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 packages. 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 packages 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 packages with at most four tutor groups. This has to be traded off with the simplicity of FCFS. Overall, we show that randomized matching mechanisms together with appropriate decision support tools are a powerful new IS design recipe for daunting coordination problems in organizations. The work falls into the broader category of IS design science (Hevner and Chatterjee 2010; Peffers et al. 2007). We design an artifact to solve a relevant problem in organizations, develop a technology-based solution, and evaluate the quality and efficacy in a field experiment.

Problem

Let us now define the combinatorial assignment problem (CAP) in the context of course assignment applications, desirable properties, and randomized mechanisms.

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*. We use the term assignment or 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 complete and transitive preference relation $\succeq_i \in \mathcal{P}$ over subsets (or bundles) $b \in B$ of elements of C. A preference profile $\succeq = (\succeq_1, \ldots, \succeq_n) \in \mathcal{P}^{|S|}$ is an n-tuple of preference relations. For most of the paper we assume strict preferences.

We can model feasible assignments with linear constraints. 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. The supply constraints make sure that the capacity of the courses are not exceeded, and the demand constraints determine that each student can win at most one bundle.

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

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

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

Courses in our application are actually tutor groups and each tutor group belongs to one of ℓ 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 $size(b) \leq \ell \ll m$. The web interface takes care that students only submit valid bundles, which have at most one tutor group for each of the ℓ classes and a size less than or equal to ℓ.

A *deterministic combinatorial assignment* (deterministic matching) is a mapping $M \subseteq S \times B$ of students S and bundles B of courses C. $\mathcal{M}$ describes the set of all deterministic matchings. A matching is *feasible* if it is a feasible integer solution to the constraints demand and supply. *Random combinatorial assignments* (random matchings) are related to fractional assignments with $0 \leq x_{ib} \leq 1$ and random assignment mechanisms can be used to fractionally allocate bundles of course seats to students. A *lottery* L is a probability distribution over feasible deterministic matchings.

Nguyen et al. (2016) show that a lottery of bundles induces probability shares over these bundles that satisfy demand and supply constraints. Thus a lottery coincides with a fractional solution to both constraints. However, a fractional solution respecting demand and supply does not need to have a lottery over deterministic assignments.

For (non-combinatorial) assignment problems with single-unit demands 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. However, the Birkhoff-von-Neumann theorem fails when bundles of course seats need to be assigned. Nguyen et al. (2016) generalize this result 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 $\ell - 1$ course seats.

Design Desiderata

Efficiency, envy-freeness, and strategy-proofness are design desiderata of first-order importance typically considered in the theoretical literature on deterministic assignment problems. For randomized mechanisms one has to reconsider these design desiderata and we will briefly introduce relevant definitions in this section.

Stochastic dominance (SD) is the key concept among all of these definitions as it provides a natural way to compare random assignments (Bogomolnaia and Moulin 2001b). Let Δ 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 1 (SD-Preference)**.** A student $i \in S$ *SD-prefers* an assignment $p \in \Delta$ over $q \in \Delta$, $p \succeq_i^{SD} q$, if

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

In other words, a student i prefers the random assignment p to the random assignment q if p_i 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 (p_i dominates q_i if $p_{ib} \geq q_{ib}$ for all $b \in B$) 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).

One desirable property of matchings is *(Pareto) efficiency* such that no student can be made better off without making any other student worse off. A deterministic matching M is *efficient* with respect to the students if there is no other feasible matching M' such that $M'(i) \succeq_i M(i)$ for all students $i \in S$ and $M'(i) \succ_i M(i)$ for some $i \in S$. One can generalize this to random matchings and lotteries:

Definition 2 (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 2001b).

Fairness is another important design goal. A basic notion of fairness for randomized assignments is the *equal treatment of equals*, i.e., students with identical preferences receive identical (symmetric) random allocations. Envy-freeness is stronger.

Definition 3 (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 she is faced with to the random assignment 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. SD-envy-freeness implies equal treatment of equals.

An assignment mechanism is an algorithm, which computes a matching M for given preferences of students. More formally, A *deterministic assignment mechanism* is a function $\chi : \mathcal{P}^{|S|} \to \mathcal{M}$ that returns a feasible matching $M \in \mathcal{M}$ of students to courses for every preference profile of the students.

A *randomized assignment mechanism* is a function $\psi : \mathcal{P}^{|S|} \to \Delta$ that returns a random matching $p \in \Delta$. The mechanism $\psi(\succeq) = p$ is *ordinally efficient* if it produces ordinally efficient allocations. We call ψ *ex post Pareto efficient*, if p can be decomposed as a convex combination of Pareto optimal matchings. ψ 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 that 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.

An important property of a 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 4 (Strategy-Proofness)**.** Let $\succeq \in \mathcal{P}^{|S|}$ be the (true) preference profile. A deterministic assignment mechanism χ 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. Hence, 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. For randomized mechanisms we need to adapt the definitions.

Definition 5 (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) ψ 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) ψ is called *weakly SD-strategy-proof* if there exists no $\succeq_i' \in \mathcal{P}$ for some student $i \in S$ such that $\psi(\succeq_i', \succeq_{-i}) \succeq_i^{SD} \psi(\succeq)$.

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 that strictly prefers $\psi(\succeq_i', \succeq_{-i})$ over the truthful outcome, but there may be students i that neither prefer $\psi(\succeq_i', \succeq_{-i})$ nor $\psi(\succeq)$. This can happen as the $\succeq^{SD}$-relation is not complete. We will omit the prefix SD for brevity in the following.

Assignment Mechanisms

A lot is known about assignment problems with single-unit demand. 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 ones. Gibbard (1977) showed that RSD is the only anonymous and symmetric, strongly SD-strategy-proof, and ex post efficient assignment rule when preferences are strict. Pycia and Troyan (2016) prove that RSD is the unique mechanism that is obviously strategy-proof, efficient, and symmetric in mechanisms without transfers.

However, RSD is not always ordinally efficient, only ex post efficient (Bogomolnaia and Moulin 2001a). Zhou (1990) actually 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 hope for these properties in combinatorial assignment problems. RSD can also be applied to the combinatorial assignment problem. The Bundled Random Serial Dictatorship (BRSD) orders the students randomly and assigns the most preferred bundle which is still available to each student in this order. Although the package preferences take some toll on the runtime it is still very fast.

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 almost random who arrives first. We will simulate FCFS via BRSD and run the algorithm repeatedly to get estimates for performance metrics of FCFS.

Probabilistic Serial (PS) (Bogomolnaia and Moulin 2001a) produces an envy-free assignment with respect to the reported unit-demand preferences, and it is ordinally efficient, but it is only weakly SD-strategy-proof. 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. The BPS mechanism is also ordinally efficient, envy-free, and weakly strategy-proof if preferences are strict.

Informally, in BPS 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.[2]

Artifact

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

[2]Due to space constraints we omit a formal description of BPS and the decomposition algorithm. Interested Readers are referred to Nguyen et al. (2016).

Background on the Application

The Technical University of Munich has been using stable matching mechanisms for the assignment of students to courses since 2014. The system provides a web-based user interface and every semester almost 1500 students are being matched to lab 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 paper, the web-based software for BPS and for BRSD was implemented. 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[3] and an information systems student could have preferences for up to 5184 $(= 9 \cdot 24 \cdot 24)$ bundles.[4] During the winter term 2017/2018, 1778 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.[5]

Automated Ranking of Packages

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 packages 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, the time that students spend for commuting is significant. Also long waiting times between courses are perceived as a waste of time as it is often hard for them 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 1 shows the initial page where a student can select the courses of interest. First, 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, then it will no longer be considered in order to allow students to participate in the lecture. In a second step, the student marks available time ranges in the *weekly schedule*. 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. A student 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 packages. 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,

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

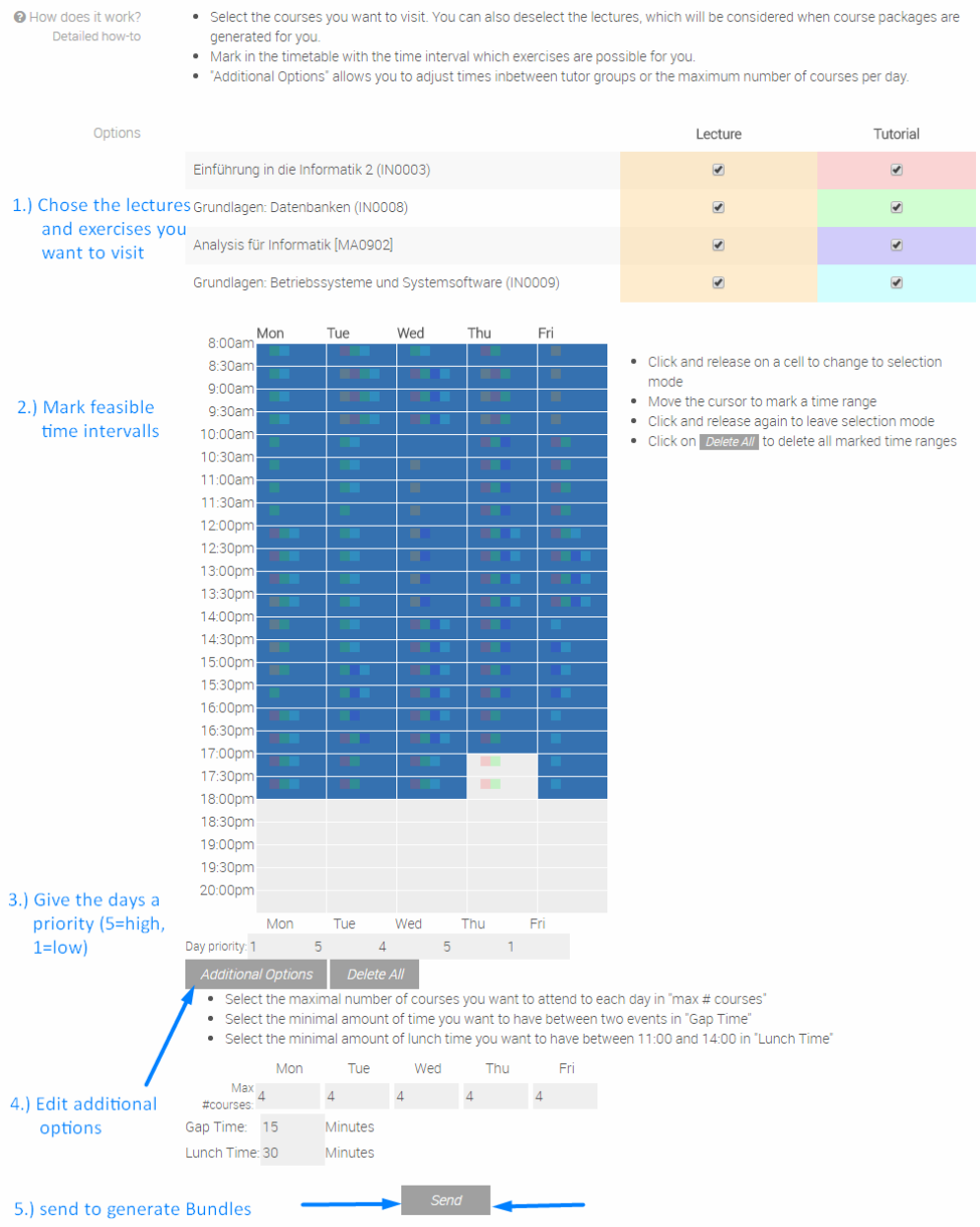

Figure 1. Process to rank-order packages

- 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.

These criteria and their implementation are developed in cooperation with the students representatives of the computer science department of the TUM. The implementation might be special, however, as we will see in the survey evaluation, the students found that their preferences were well expressed by this method.

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

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

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 her tutorials on a single day, her most preferred day, have a 1-hour lunch break and a minimal time for breaks inbetween 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 $(f\,(sp\,(b,day)))$. This way 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 $(score\,(b,d)))$ 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 $\{1,\ldots,5\}$. If the student does not have to visit the university at day d, he gets a fixed score of 30 for day d. 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 packages.

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. Note that $\approx 90\%$ of the students received one of their top ten ranked packages and only a few students received a package with a rank less than 30. So, if a student inspects and confirms the ranking of the first 10-30 packages, 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.

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 the pre-ranked packages 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.

Evaluation

We already have summarized first-order design goals 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.

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: $\phi_i(b, b') = \left\{ \begin{array}{c} +1 \text{ if } b \succ_i b' \\ -1 \text{ if } b' \succ_i b \\ 0 \text{ else} \end{array} \right\}$.

Definition 6 (Popularity)**.** A random assignment $p \in \Delta$ is *more popular* than an assignment q, denoted $p \blacktriangleright q$, if $pop(p, q) > 0$ with $pop(p, q) = \sum_{i \in S} \sum_{b, b' \in B} p_{ib} \cdot q_{ib'} \cdot \phi_i(b, b')$. 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 simply 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 were (fractionally) assigned to their first choice, how many to their second choice, 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 also used in Diebold and Bichler (2017). The *Area Under the Profile Curve Ratio* (AUPCR) for matching problems with bundles is defined as follows:

Definition 7 (AUPCR)**.** With R denoting the number of possible ranks and $b \in B$, the AUPCR is:

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

Empirical Results

The first application from the summer term 2017 comprised 1415 students and 67 courses (see Table 1). 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[6].

Popularity

For the data from the summer and the winter term, BPS is more *popular* than BRSD. 636 students prefer BPS to FCFS, while 96 students prefer FCFS to BPS. 683 students are indifferent (see Table 1). A positive popularity score as described in Definition 6 means, that BPS is more popular than the BRSD outcome and the score for BPS is 2.74 for the summer term and 3.41 for the winter term (compared to BRSD). For the data from the winter term 754 students prefer BPS to FCFS, while 120 students prefer FCFS to BPS. 862 students are indifferent. Table 1 summarizes popularity and stochastic dominance for the summer and the winter term. The syntax for the SD-*preference* is the number of students preferring (BPS|BRSD). It shows that BPS is preferable to BRSD according to SD-preference.

	Summer Term		Winter Term	
Metric	BPS	BRSD	BPS	BRSD
exp. rank	2.20163	2.20835	1.97372	1.97873
exp. size	1086.58	1085.79	1603.01	1600.84
prob. match (top 100)	0.767901	0.767345	0.922253	0.922142
AUPCR	0.747419	0.750782	0.889512	0.888058
weak envy	0	381	0	451
strong envy	0	1064	0	1202
popularity	2.73635		3.41499	
SD-preference	(636|96)		(754|120)	

Table 1. Summary statistics for the summer term 2017 and the winter term 2017/2018.

Rank and Size

Table 1 reports that in terms of average rank, average size, and the probability of being matched to one of the first 100 ranks BPS achieves higher scores in the summer term. Only the AUPCR for BRSD 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. In the BPS outcome 72.735% of the students receive an assignment ranked in their top ten while in BRSD 72.637% receive such an outcome (see Table 2 for BPS and Table 3 for BRSD). Table 2 reports the probability of being matched to a particular rank and the AUPC in percentage for BPS, and Table 3 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 2. 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. Rank profile BRSD in summer term 2017.

The second application in the winter term 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 1). In the BPS outcome 89.047% of the students receive an assignment ranked in their top ten while in BRSD 88.891% receive such an outcome (see Table 4 for BPS and 5 for BRSD). The computation

[6]For brevity we omit the number of runs in the following and only report the results for 1 Mio runs of BRSD.

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.

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 Definition 3), from which 381 students do not even fulfill the weak envy-freeness condition (see BRSD in Table 1). Similarly, for the winter term 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 in Table 1).

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 4. 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 5. Rank profile BRSD in winter term 2017/2018

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 6. Note that the students were exposed to FCFS in other semesters and now participated in BPS, which allowed them to compare both mechanisms.

Students did not have 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 as compared to those students who do not care about the assignment as much.

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 biased towards dissatisfaction. The ranking and profile information reported earlier provides alternative information about satisfaction of students with the outcome. Also, note that the reason for the introduction of BPS was the large dissatisfaction with FCFS.

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). 10.1% 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. This high percentage needs to be seen in conjunction with the exponentially large set of possible packages. If a student provides many possible time slots, then the list of packages grows very large. Therefore, there might have been a tendency to narrow down the selection of acceptable time slots, i.e., not rank the least preferred time slots.

	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 6. Survey results, values in %

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 package 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 level of envy in FCFS.

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 both 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 curve (AUPCR) are small, which is interesting given the fact that only a small number of preferences per student are considered via FCFS.

There are a number of reasons that help explaining the close performance of BPS and FCFS 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 that 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 might be small.

Third, an earlier comparison of FCFS with a deferred acceptance algorithm by Diebold et al. (2014) also showed that FCFS yielded 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 both studies. For applications of matching in practice it is important to understand these trade-offs.

Conclusions

We report two large field studies and show that BPS performs well on a number of criteria including average rank, average size, probability of a matching among the first 100 ranks, and the overall profile of ranks (in terms of AUPC of a specific rank) assuming a complete, truthful, and strict ranking of all packages. The matching based on BPS is also more popular than BRSD based on the preferences submitted for BPS. The level of envy in FCFS is significant, even though the size of the packages that can be submitted is limited to the number of classes (three to four groups per package) in our course assignment application.

As in many scheduling applications, student preferences in our course assignment application are about times of the week. We introduce a way to rank order the many possible schedules based on a few parameters. The feedback of the 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. Of course, other domains might have different requirements and the way how preferences are elicited needs to reflect the domain specifics.

Although BPS provides a convincing new alternative to scheduling problems with private preferences, there are trade-offs with first-come, first-served techniques.

In contrast to FCFS the BPS mechanism is not obviously strategy-proof and a part of the students in the survey actually indicated that they either hid their most preferred or least preferred time slots strategically. This might partly be due to the fact that students were unexperienced with this new mechanism.

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 packages. 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 (i.e., demand for only one course seat) is so successful in practice (Roth 2002). If the market outcome is unstable, there is an agent or pair of agents who have the incentive to circumvent the match. We argue that this property is as important for the assignment of course schedules. If envy-freeness matters, the elegant BPS mechanism has a number of attractive properties, which otherwise suffer from computational hardness of the allocation problem and strategic manipulation. Such provable properties are valuable in a time where algorithmic bias has become such an important concern. Envy-freeness and incentive compatibility are pivotal properties of BPS, which make the overall system design also an excellent candidate related scheduling applications within and across organizations.

References

Abdulkadiroğlu, A., Pathak, P., and Roth, A. (2009). "Strategy-proofness versus Efficiency in Matching with Indifferences: Redesigning the NYC High School Match," *The American Economic Review* (88:5), pp. 1954–1978.

Abdulkadiroğlu, A. and Sönmez, T. (2003). "School choice: A mechanism design approach," *American economic review* (93:3), pp. 729–747.

Adomavicius, G., Curley, S., Gupta, A., and Sanyal, P. (2012). "Effect of information feedback on bidder behavior in continuous combinatorial auctions," *Management Science* (58:4) 4 2012, pp. 811–830.

Adomavicius, G. and Gupta, A. (2005). "Toward Comprehensive Real-Time Bidder Support in Iterative Combinatorial Auctions," *Information Systems Research* (16:2) 2 2005, pp. 169–185.

Ashlagi, I. and Shi, P. (2016). "Optimal Allocation Without Money: An Engineering Approach," *Management Science* (62:4), pp. 1078–1097.

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

Banker, R. D. and Kauffman, R. J. (2004). "50th anniversary article: The evolution of research on information systems: A fiftieth-year survey of the literature in management science," *Management Science* (50:3), pp. 281–298.

Bapna, R., Das, S., Garfinkel, R., and Stallaert, J. (2007). "A Market Design for Grid Computing," *INFORMS Journal on Computing* (20:1) 1 2007, pp. 100–111.

Bapna, R., Dellarocas, C., and Rice, S. (2010). "Vertically Differentiated Simultaneous Vickrey Auctions: Theory and Experimental Evidence," *Management Science* (56), pp. 1074–1092.

Bapna, R., Goes, P., and Gupta, A. (2003). "Replicating Online Yankee Auctions to Analyze Auctioneers' and Bidders' Strategies," *Information Systems Research* (14:3), pp. 244–268.

Bichler, M. and Goeree, J. K. (2017). *Handbook of spectrum auction design,* Cambridge University Press.

Birkhoff, G. (1946). "Three observations on linear algebra," *Univ. Nac. Tucumán. Revista A* (5), pp. 147–151.

Bogomolnaia, A. and Moulin, H. (2001a). "A New Solution to the Random Assignment Problem," *J. Economic Theory* (100:2), pp. 295–328.

Bogomolnaia, A. and Moulin, H. (2001b). "A new solution to the random assignment problem," *Journal of Economic theory* (100:2), pp. 295–328.
Brandt, F., Hofbauer, J., and 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), pp. 1061–1103.
Budish, E., Cachon, G. P., Kessler, J. B., and Othman, A. (2017). "Course Match: A Large-Scale Implementation of Approximate Competitive Equilibrium from Equal Incomes for Combinatorial Allocation," *Operations Research* (65:2), pp. 314–336.
Budish, E., Che, Y.-K., Kojima, F., and Milgrom, P. (2013). "Designing random allocation mechanisms: Theory and applications," *American Economic Review* (103:2), pp. 585–623.
Budish, E. and Kessler, J. (2017). "Can Agents "Report Their Types"? An Experiment that Changed the Course Allocation Mechanism at Wharton," *Chicago Booth Working Paper* ().
Burke, E. K., De Causmaecker, P., Berghe, G. V., and Van Landeghem, H. (2004). "The state of the art of nurse rostering," *Journal of scheduling* (7:6), pp. 441–499.
Cardoen, B., Demeulemeester, E., and Beliën, J. (2010). "Operating room planning and scheduling: A literature review," *European journal of operational research* (201:3), pp. 921–932.
Che, Y.-K. and Kojima, F. (2010). "Asymptotic equivalence of probabilistic serial and random priority mechanisms," *Econometrica* (78:5), pp. 1625–1672.
Diebold, F., Aziz, H., Bichler, M., Matthes, F., and Schneider, A. (2014). "Course allocation via stable matching," *Business & Information Systems Engineering* (6:2), pp. 97–110.
Diebold, F. and Bichler, M. (2017). "Matching with indifferences: A comparison of algorithms in the context of course allocation," *European Journal of Operational Research* (260:1), pp. 268–282.
Ehlers, L. and Klaus, B. (2003). "Coalitional strategy-proof and resource-monotonic solutions for multiple assignment problems," *Social Choice and Welfare* (21:2), pp. 265–280.
Ernst, A. T., Jiang, H., Krishnamoorthy, M., and Sier, D. (2004). "Staff scheduling and rostering: A review of applications, methods and models," *European journal of operational research* (153:1), pp. 3–27.
Gale, D. and Shapley, L. S. (1962). "College Admissions and the Stability of Marriage," *American Mathematical Monthly* (69:1), pp. 9–15.
Gibbard, A. (1977). "Manipulation of schemes that mix voting with chance," *Econometrica: Journal of the Econometric Society* (), pp. 665–681.
Goetzendorff, A., Bichler, M., Day, B., and Shabalin, P. (2015). "Compact bid languages and core-pricing in large multi-item auctions," *Management Science* (). forthcoming.
Hanley, J. A. and McNeil, B. J. (1982). "The meaning and use of the area under a receiver operating characteristic (ROC) curve." *Radiology* (143:1), pp. 29–36.
Hevner, A. and Chatterjee, S. (2010). "Design science research in information systems," in *Design research in information systems,* Springer, pp. 9–22.
Kamada, Y. and Kojima, F. (2017). "Recent developments in matching with constraints," *American Economic Review* (107:5), pp. 200–204.
Kavitha, T., Mestre, J., and Nasre, M. (2011). "Popular mixed matchings," *Theoretical Computer Science* (412:24), pp. 2679–2690.
Klemperer, P. (1999). "Auction theory: A guide to the literature," *Journal of economic surveys* (13:3), pp. 227–286.
Krishna, A. and Ünver, M. U. (2008). "Research Note—Improving the Efficiency of Course Bidding at Business Schools: Field and Laboratory Studies," *Marketing Science* (27:2), pp. 262–282.
Lee, Y. E. and 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), pp. 867–884.
Li, S. (2017). "Obviously strategy-proof mechanisms," *American Economic Review* (107:11), pp. 3257–87.
Milgrom, P. (2010). "Simplified Mechanisms with Applications to Sponsored Search and Package Auctions," *Games and Economic Behavior* (70:1), pp. 62–70.
Milgrom, P. (2004). *Putting auction theory to work,* Cambridge University Press.

Müller, A. and Stoyan, D. (2002). *Comparison methods for stochastic models and risks,* vol. 389. Wiley New York.

Nguyen, T., Peivandi, A., and Vohra, R. (2016). "Assignment Problems with Complementarities," *Journal of Economic Theory* (165), pp. 209–241.

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

Pápai, S. (2001). "Strategyproof and nonbossy multiple assignments," *Journal of Public Economic Theory* (3:3), pp. 257–271.

Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). "A design science research methodology for information systems research," *Journal of management information systems* (24:3), pp. 45–77.

Pycia, M. and Troyan, P. (2016). "Obvious dominance and random priority," ().

Roth, A. E. (2015). *Who Gets What—and Why: The New Economics of Matchmaking and Market Design,* Houghton Mifflin Harcourt.

Roth, A. E. (2002). "The Economist as Engineer: Game Theory, Experimentation, and Computation as Tools for Design Economics," *Econometrica* (40:4), pp. 1341–1378.

Santos, B. L. D. and Bariff, M. L. (1988). "A Study of User Interface Aids for Model-Oriented Decision Support Systems," *Management Science* (34:4), pp. 461–468.

Scheffel, T., Pikovsky, A., Bichler, M., and Guler, K. (2011). "An Experimental Comparison of Linear and Non-Linear Price Combinatorial Auctions," *Information Systems Research* (22:2) 2 2011, pp. 346–368.

Sönmez, T. and Ünver, M. U. (2010). "COURSE BIDDING AT BUSINESS SCHOOLS," *International Economic Review* (51:1), pp. 99–123.

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

Zhou, L. (1990). "On a conjecture by Gale about one-sided matching problems," *Journal of Economic Theory* (52:1), pp. 123–135.