DESIGNING RANDOM ALLOCATION MECHANISMS: THEORY AND APPLICATIONS

ERIC BUDISH, YEON-KOO CHE, FUHITO KOJIMA, AND PAUL MILGROM

Abstract. Randomization is an important feature of resource allocation when the resources assigned are indivisible and monetary transfers are limited. We expand the theory of efficient random assignment to accommodate multi-unit demand and supply, and certain real-world features such as group-specific quotas ("controlled choice") and endogenous capacities in school choice and house allocation, and scheduling and curriculum constraints in course allocation. We develop new mechanisms that are ex ante fair and efficient in these respective problems. Our method can also be applied to certain two-sided matching problems to produce fair matchups in interleague games and speed dating.

Keywords: Market Design, Random Assignment, Birkhoff-von Neumann Theorem, Probabilistic Serial, Pseudo-Market, Utility Guarantee, Assignment Messages.

Date: February 17, 2011.

Budish: University of Chicago Booth School of Business, 5807 S Woodlawn Ave, Chicago IL 60637, eric.budish@chicagobooth.edu. Che: Department of Economics, Columbia University, 420 West 118th Street, 1016 IAB New York, NY 10027, and YERI, Yonsei University, yeonkooche@gmail.com. Kojima: Department of Economics, Stanford University, CA 94305, fuhitokojima1979@gmail.com. Milgrom: Department of Economics, Stanford University, CA 94305, paul@milgrom.net. We are grateful to Emir Kamenica, Sebastien Lahaie, Mihai Manea, Herve Moulin, Tayfun Sönmez, Al Roth, Utku Ünver, and seminar participants at Boston College, Gakushuin, Johns Hopkins, Princeton, Tokyo, UC San Diego, Waseda, Yahoo! Research, and Yale for helpful comments. We especially thank Tomomi Matsui, Akihisa Tamura, Jay Sethuraman, and Rakesh Vohra. Yeon-Koo Che is grateful to Korea Research Foundation for World Class University Grant, R32-2008-000-10056-0. Paul Milgrom's research was supported by NSF grant SES-0648293.

Randomization is commonplace in everyday resource allocation. It is used to break ties among students applying for overdemanded public schools and for popular after-school programs, to ration offices, parking spaces, and tasks among employees, to allocate courses and university dormitories among college students, and to assign jury and military duties among citizens.[1] The ubiquitous "first-come first-served" method, or "queuing," is often only a less apparent way to include random elements in setting a priority order. Randomization is sensible in these examples and many others because the objects to be assigned are indivisible and monetary transfers are limited or unavailable.[2] In these circumstances, any non-random assignment of resources is likely to be asymmetric and, without compensating monetary transfers, "unfair." Randomization can sometimes restore ex ante symmetry and the perception that the mechanism is fair.

To find a desirable random allocation, it is often helpful to view agents as consuming lotteries over objects, subject to a joint production constraint and to evaluate incentives and welfare on that basis.[3] Since a **random assignment**— the lotteries over objects received by agents—is "divisible" in probability units, one can then apply the classical frameworks developed for divisible objects. Hylland and Zeckhauser (1979) ("HZ") were the first to apply this perspective to market design. Their pseudo-market mechanism endows agents with equal fixed budgets in a fictitious currency, allows them to use the currency to "buy" probability shares of alternative objects, and finds a competitive equilibrium by solving for prices (per unit probability of obtaining each object) that clear the market. The resulting allocation is then ex ante efficient and envy-free. Bogomolnaia and Moulin (2001) ("BM") also adopted the random assignment approach to find an allocation that is efficient in an ordinal sense and envy-free.

[1] Lotteries played historical roles in assigning public lands to homesteaders (Oklahoma Land Lottery of 1901), and radio spectra to broadcasting companies (FCC assignment of radio frequencies during 1981-1993). Lotteries are also used annually to select 50,000 winners of the US permanent residency visas ("green cards") from those qualified in the DoJ's immigration diversity program.

[2] The limitation of monetary transfers arises from moral objection to "commoditizing" objects such as human organs and from fairness consideration (cf. Roth (2007)). Assignment of resources based on prices often favor those best endowed with money rather than those most deserving, and can be regarded as unfair for many goods and services. See Che and Gale (2008) for an argument making this point based on utilitarian efficiency.

[3] An more common alternative perspective evaluates the efficiency of agents' ex post assignment of objects, rather than of the ex ante lotteries they receive. In that view, a mechanism identifies ex post desirable allocations and uses randomization only to ensure ex ante fairness. For example, the random serial dictatorship (RSD), widely used in practice, is often analyzed that way. In RSD, the agents are randomly ordered and, following that order, each agent is assigned his/her most preferred object not yet assigned. The resulting pure assignment is ex post efficient and ex ante envy-free, but may entail ex ante inefficiencies even in an ordinal sense (see Bogomolnaia and Moulin (2001)). Extensions of other well known mechanisms, such as Gale and Shapley's deferred acceptance and Gale's top trading cycles, suffer similar ex ante inefficiencies when priorities are set randomly and used to break ties.

The purpose of the current paper is to broaden the random assignment methodology (including HZ and BM) to enhance its practical applicability. To be applied to problems including school choice and course allocation, the random assignment model must be extended in two ways. First, the model needs to account for various policy constraints. A case in point is the so-called "controlled-choice" in school assignment. Schools often seek to balance their student bodies in terms of gender, ethnicity, race, test scores, and the geographic location of students' residence. For instance, public schools in Massachusetts are discouraged by the Racial Imbalance Law from having student enrollments that are more than 50% minority. Miami-Dade County Public Schools control for the socioeconomic status of students in order to diminish concentrations of low-income students at certain schools. In New York City, "Educational Option" (EdOpt) schools must balance their student bodies in terms of students' test scores.[4] Public schools in Seoul restrict the number of seats for those students residing in distant school districts, in order to alleviate morning commutes. In a course allocation problem, a student may wish to enroll in no more than a certain number of courses in a given subject (curriculum constraints) or in a given time slot (scheduling constraints). The preceding examples are ones in which we add constraints to the ones already present in the HZ and BM models, but some applications also involve removing or relaxing the traditional constraints. For instance, when schools assign their students to different foreign language programs, the exact composition may be adjustable to a degree determined by the available staff and resource.

Second, in order to accommodate applications like course allocation, in which a single course may accept multiple students and a single student may take multiple courses, we need to drop the one-to-one restriction. We do that by using the matrix of *expected assignments* to generalize the random assignment matrix of one-to-one matching. This expected assignment formulation is most useful when the agents' and mechanism designer's parties' payoffs are at least approximately linear functions of the pure assignment over the support of the relevant lottery. Part of our analysis below will focus on implementing using lotteries with supports for which this restriction might reasonably apply.

Even for one-to-one matching, describing the individual agent's consumption outcome in terms of their individual random assignments poses a technical challenge. For even if each agent is assigned just one item in expectation and each item is assigned just once in expectation, implementation still requires finding a joint lottery over feasible pure assignments with the right marginal distributions for each agent. For HZ and BM, the only feasibility constraints on the pure assignments are those of one-to-one matching: each agent is assigned

[4] In particular, 16 percent of students that attend an EdOpt school must score above grade level on the standardized English Language Arts test, 68 percent must score at grade level, and the remaining 16 percent must score below grade level (Abdulkadiroglu, Pathak, and Roth, 2005).

exactly one item and each item is assigned exactly once. Those papers were then able to solve their technical challenge by appealing to the celebrated celebrated Birkhoff-von Neumann theorem (Birkhoff, 1946; von Neumann, 1953), which asserts that every random assignment matrix, that is, every non-negative matrix with all the row sums and column sums equal to one, is a convex combination of pure assignment matrices. Consequently, every random assignment matrix can be implemented by some lottery over pure assignments.

Is there an extension of the BvN theorem available when there are additional constraints, such as the ones described above? Does the extension apply as well to many-to-many matching problems? The first part of the paper answers these questions by identifying a maximal generalization of the Birkhoff-von Neumann theorem. Appealing to results in the combinatorial optimization literature, we show that a certain underlying structure of constraints is sufficient for an expected assignment to be always implementable by a lottery over feasible pure assignments. Then we demonstrate that the same condition is not only sufficient but also necessary in canonical two-sided environments.

The second part of the paper applies the expected assignment methodology to specific market design contexts. One application is a generalization of BM's Probabilistic Serial mechanism that accommodates new kinds of supply-side constraints, which may arise in unit-demand applications such as school choice and dormitory assignment. We show how to modify BM's algorithm to accommodate constraints such as controlled choice and adjustable capacities and prove that the attractive properties of BM's algorithm extend to this more general environment.

Our second application is a generalization of HZ's pseudo-market mechanism, to accommodate new kinds of demand-side constraints that may express important aspects of participants' preferences in multi-unit demand applications such as course allocation and assignment of shifts to interchangeable workers.

In this section, we also relax the assumption of additive preferences. We adapt the "assignment messages" developed by Milgrom (2009), to accommodate some scheduling and curricular constraints ("I want just one course in finance and at most one course before noon") arising in course allocation, and to express nonlinear preferences such as diminishing marginal utilities for an item or category ("the second finance course is worth less to me than the first"). We then utilize these enriched messages to develop a generalized multi-unit pseudo-market mechanism, establish existence of competitive equilibrium prices in the pseudo-market, and invoke our earlier sufficiency result to ensure implementability of the expected assignment that results from this competitive equilibrium. We finally show that this generalized mechanism inherits the attractive efficiency and fairness properties of the original one-to-one mechanism. This extended mechanism may be useful for practice, especially because several multi-unit assignment mechanisms currently in use have been shown to allow outcomes that are inefficient and ex post quite unfair (Sönmez and Ünver, 2008; Budish and Cantillon, 2009).

Finally, our implementation result has an unexpected application for promoting ex post fairness in multi-unit resource allocation. When agents demand multiple objects, there can be many ways to implement a given expected assignment. For instance, suppose there are two agents, 1 and 2, dividing four objects, a, b, c, and d, which they prefer in the order listed. An ex ante fair expected assignment may assign each object to each agent with probability 0.5. One way to implement this expected assignment is to assign a and b to 1 and c and d to 2 with probability one half and a and b to 2 and c and d to 1 with the remaining probability one half. If utility is additive over objects, then this allocation may be ex ante efficient and fair, but ex post unfair, since one agent always gets the two best and the other gets the two worst. There is another implementation of the same expected assignment that is more fair ex post: whenever one agent gets one of the two best objects, he must also get one of the two worst objects. It turns out our implementation result can be utilized to avoid the former unfair implementation, and more generally to ensure that pure assignments used in the implementing lottery have a small variation in utility. The method also ensures that every pure assignment approximates the original expected assignment in terms of expected utilities. This procedure can be applied in the context of course allocation, for instance in conjunction with our generalization of HZ, or in other multi-unit demand environments such as task assignment and fair division of estates.

This utility guarantee method can also be adapted to a two-sided matching problem, in which both sides of the market are agents. Starting with any expected matching, we can introduce ex post utility guarantees on both sides, ensuring ex post utility levels that are close to the promised ex ante levels. This method can be used, for example, to design a fair schedule of inter-league sports matchups or a fair speed-dating mechanism.

The rest of the paper is organized as follows. Section 1 presents the model. Section 2 presents the sufficiency and necessity results for implementing expected assignments. Section 3 presents the generalization of Bogomolnaia and Moulin's (2001) Probabilistic Serial mechanism, for applications such as school choice. Section 4 presents the generalization of Hylland and Zeckhauser's (1979) Pseudo-market mechanism, for applications such as course allocation. Section 5 presents the utility guarantee results, including the application to two-sided matching. Section 6 collects some negative results for non-bilateral matching environments. Section 7 concludes.

1. Setup

Consider a problem in which a finite set N of agents are assigned to a finite set O of objects (as will be clear, our framework works with no loss if N are objects or O are agents). A (pure) **assignment** is described as a matrix $\overline{X} = [\overline{x}_{ia}]$ indexed by all agents and objects, where each entry $\overline{x}_{ia}$ is the integer quantity of object a that agent i receives. Note that we allow for assigning more than one unit of an object and even for assigning negative quantities. Negative quantities can be interpreted as supply obligations. The requirement that the matrix be integer-valued captures the indivisibility of the assignment.

We introduce a general class of constraints. A **constraint structure** $\mathcal{H}$ is a collection of subsets of $N \times O$ (that is, $\mathcal{H} \subset 2^{N \times O}$) that includes all singletons (i.e., sets of the form $\{(i,a)\}$). That is, each element $S \in \mathcal{H}$, called a **constraint set**, is a set of agent-object pairs. A vector $\mathbf{q} = (\underline{q}_S, \overline{q}_S)_{S \in \mathcal{H}}$ of integers are the **quotas** associated with each set in $\mathcal{H}$, where integers $\underline{q}_S$ and $\overline{q}_S$ are the floor and the ceiling constraints on the total amount assigned in S, respectively. The constraint structure $\mathcal{H}$ and the quotas $\mathbf{q}$ restrict the set of assignments. We say that a pure assignment $\overline{X}$ is **feasible under q** if

$$\underline{q}_S \leq \sum_{(i,a) \in S} \overline{x}_{ia} \leq \overline{q}_S \text{ for each } S \in \mathcal{H}. \tag{1.1}$$

For instance, the constraint structure $\mathcal{H}$ of the simple one-to-one matching setting consists of all singleton sets, "all rows" ($\{(i,a)|a \in O\}$ for each $i \in N$), and "all columns" ($\{(i,a)|i \in N\}$ for each $a \in O$), and the quotas satisfy $\underline{q}_S = \overline{q}_S = 1$ for every non-singleton constraint set $S \in \mathcal{H}$. The constraints say that each agent is assigned one object and each object is assigned to one agent.

Given a constraint structure $\mathcal{H}$ and associated quotas $\mathbf{q}$, a random allocation can be described as a lottery over pure assignments each of which is feasible under quotas $\mathbf{q}$. As with HZ and BM, however, our approach is to focus directly on an expected assignment—namely, the expected numbers of each item assigned to each agent—as a basic unit of analysis. Formally, an **expected assignment** is a matrix $X = [x_{ia}]$ indexed by agents and objects where $x_{ia} \in (-\infty, \infty)$ for every $i \in N$ and $a \in O$. In contrast to pure assignment matrices, an expected assignment allows for fractional allocations of objects. A natural question is: when can an expected assignment be implemented by some lottery over pure assignments?

Definition 1. *Given a constraint structure $\mathcal{H}$, an expected assignment X is **implementable** (as a lottery over feasible pure assignments) under quotas* $\mathbf{q}$ *if there exist positive numbers $\{\lambda^k\}_{k=1}^K$ that sum up to one and (pure) assignments $\{\overline{X}^k\}_{k=1}^K$, each of which is feasible under*

$\mathbf{q}$, *such that*

$$X = \sum_{k=1}^{K} \lambda^k \overline{X}^k. \tag{1.2}$$

In words, given quotas, an expected assignment is implementable if it can be expressed as a lottery over feasible assignments each of which satisfies the quotas.

If an expected assignment X is implementable under quotas $\mathbf{q}$, then the expected assignment will trivially satisfy the quotas $\mathbf{q}$:

$$\underline{q}_S \leq \sum_{(i,a)\in S} x_{ia} \leq \overline{q}_S \text{ for each } S \in \mathcal{H}. \tag{1.3}$$

The more challenging question is the reverse: when is an expected assignment implementable? More specifically, our interest is to identify the constraint structures for which any quotas are implementable. Our characterization will be provided in terms of the constraint structure, so the following definition will prove useful.

Definition 2. *Constraint structure $\mathcal{H}$ is* ***universally implementable*** *if, for any quotas $\mathbf{q} = (\underline{q}_S, \overline{q}_S)_{S\in\mathcal{H}}$, every expected assignment satisfying $\mathbf{q}$ is implementable under $\mathbf{q}$.*

If a constraint structure $\mathcal{H}$ is universally implementable, then every expected assignment satisfying any quotas defined on $\mathcal{H}$ can be expressed as a convex combination of pure assignments that are feasible under the given quotas. In other words, for any given quotas, any expected assignment satisfying (1.3) can be implemented as a lottery over feasible pure assignments.

Universal implementability aims to capture the sort of information that is likely available to a planner when the mechanism is being designed. For example, in a school choice problem, the planner may consider whether to apply certain principled geographic and ethnic composition constraints — that is, what the constraint structure will be — before knowing the exact numbers of spaces in each school or the precise preferences of the students. By studying universal implementability, we characterize the kinds of constraint structures that are robust to these numerical details and certain to be implementable.

2. Implementing Expected Assignments

This section provides a condition which guarantees that a constraint structure is universally implementable. To do so, we introduce concepts that will be useful for our characterization. A constraint structure $\mathcal{H}$ is a **hierarchy** if, for every pair of elements S and S' in

$\mathcal{H}$, we have $S \subset S'$ or $S' \subset S$ or $S \cap S' = \emptyset$.[5] That is, $\mathcal{H}$ is a hierarchy if, for any two of its elements, one of them is a subset of the other or they are disjoint. The following concept proves to be central throughout the paper.

Definition 3. *A constraint structure $\mathcal{H}$ is a* ***bihierarchy*** *if there exist hierarchies $\mathcal{H}_1$ and $\mathcal{H}_2$ such that $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$ and $\mathcal{H}_1 \cap \mathcal{H}_2 = \emptyset$.*

A bihierarchy is a constraint structure that can be expressed as a union of two disjoint hierarchies. Note that the partition of $\mathcal{H}$ into $\mathcal{H}_1$ and $\mathcal{H}_2$ need not be unique. For instance, singleton sets can be put into the two families in any arbitrary fashion. The result offered below follows readily from the combinatorial optimization literature, establishing that the bihierarchy condition is sufficient for universal implementability.

Theorem 1. (SUFFICIENCY) *If a contraint structure is a bihierarchy, then it is universally implementable.*

Proof. Suppose that constraint structure $\mathcal{H}$ is a bihierarchy. Consider any expected assignment X that satisfies $\mathbf{q}$ given constraint structure $\mathcal{H}$. Since $\underline{q}_S$ and $\overline{q}_S$ are integers for each $S \in \mathcal{H}$, we must have $\underline{q}_S \le \lfloor x_S \rfloor \le x_S \le \lceil x_S \rceil \le \overline{q}_S$, where $x_S := \sum_{(i,a)\in S} x_{ia}$, $\lfloor x_S \rfloor$ is the largest integer no greater than x_S, and $\lceil x_S \rceil$ is the smallest integer no less than x_S. Hence, X must belong to the set

$$\left\{ X' = [x'_{ia}] \middle| \lfloor x_S \rfloor \le \sum_{(i,a)\in S} x'_{ia} \le \lceil x_S \rceil, \forall S \in \mathcal{H} \right\}. \tag{2.1}$$

The set (2.1) forms a bounded polytope. Hence, any point of it, including X, can be written as a convex combination of its vertices. To prove the result, therefore, it suffices to show that the vertices of (2.1) are integer-valued. Hoffman and Kruskal (1956) show that the vertices of (2.1) are integer-valued if and only if the incidence matrix $M = [m_{(i,a),S}]$, $(i,a) \in N \times O, S \in \mathcal{H}$, where $m_{(i,a),S} = 1$ if $(i,a) \in S$ and $m_{(i,a),S} = 0$ if $(i,a) \notin S$, is totally unimodular.[6] The total unimodularity of matrix M in turn follows from Edmonds (1970). A fuller self-contained proof is available in Web Appendix C. □

As is clear from the proof, the pure assignments used in the implementing lottery in Theorem 1 are not just feasible under the given quotas; rather, the implementation ensures that each of the resulting pure assignments is rounded up or down to the nearest integer for each constraint set, which is a stronger property.

[5] Hierarchies are usually called laminar families in the combinatorial optimization literature.

[6] A zero-one matrix is totally unimodular if the determinant of every square submatrix is 0, -1 or $+1$.

For practical purposes, knowing simply that an expected assignment is implementable is not satisfactory; implementation must be computable, preferably by a fast algorithm. Fortunately, there exists an algorithm, formally described in Web Appendix D with an illustrative example, that implements expected assignments in polynomial time.[7] At each step of the algorithm, an expected assignment X satisfying given quotas is decomposed into a convex combination $\gamma X' + (1-\gamma)X''$ of two expected assignments, each of which satisfies the quotas and has at least one more integer-valued constraint set than X. Then, a random number is generated and with probability γ the algorithm continues by similarly decomposing X', while with probability $1-\gamma$ the algorithm continues by decomposing X''. The algorithm stops when it reaches a pure assignment. As argued more formally in the appendix, this process has a run time polynomial in $|\mathcal{H}|$.

2.1. **Examples of Bihierarchy.** We show here that a number of constraints illustrated in the Introduction satisfy the bihierarchy condition.

2.1.1. *One-to-one assignment and the Birkhoff-von Neumann theorem.* Suppose n agents are to be assigned to n objects, one for each. Notice that any assignment is described as an $n \times n$ permutation matrix; namely, each entry is zero or one, each row sums to one, and each column sums to one. Any expected assignment is in turn represented as an $n \times n$ bistochastic matrix, i.e., a matrix with entries in $[0,1]$, satisfying the same row-sum and column-sum constraints. The Birkhoff-von Neumann theorem states that any bistochastic matrix can be expressed as a convex combination of permutation matrices. Clearly, this result follows from Theorem 1; all rows are disjoint and thus form a hierarchy, and all columns form another. (Singletons can be added arbitrarily to either hierarchy).

Corollary 1. (Birkhoff, 1946; von Neumann, 1953) *Any bistochastic matrix can be written as a convex combination of permutation matrices.*

2.1.2. *Endogenous Capacities.* Consider a school choice problem in which the school authority wishes to run several education programs within one school building. An assignment in such an environment can be described as a matrix in which rows correspond to students and columns correspond to education programs (rather than school buildings). With this representation, a school building corresponds to multiple columns. Formally, we can let $\mathcal{H}_1$ include all rows, which correspond to students, while $\mathcal{H}_2$ includes sets of the form $N \times O'$ where O' is a subset of educational programs: A singleton set O' represents an individual

[7] We thank Tomomi Matsui and Akihisa Tamura for suggesting the algorithm. An earlier draft of this paper included an alternative algorithm generalizing the stepping-stones algorithm described by Hylland and Zeckhauser (1979).

educational program while a larger set O' corresponds to a school building with multiple programs. The ceiling $\overline{q}_{N\times O'}$ then describes the total number of students who can be admitted within O', which may apply to a single program or a set of programs in the same building. If the sum of ceilings $\sum_{a\in O'}\overline{q}_{N\times\{a\}}$ is larger than the ceiling $\overline{q}_{N\times O'}$, then that means the sizes of programs within the same school building O' can be adjusted, subject to the (say physical) capacity of the school building. For instance, in Figure 1, columns b and c represent two programs (within a school) each of which is subject to a quota, and there is a school wide quota impinging on b and c, together. Note that a constraint structure composed of $\mathcal{H}_1$ and $\mathcal{H}_2$ described above is a bihierarchy, thus it is universally implementable. Notice also that the hierarchical structure $\mathcal{H}_2$ allows for nested constraints on program sizes.

Figure 1 about here

2.1.3. *Group-specific Quotas.* Affirmative action policies are sometimes implemented as quotas on students fitting specific gender, racial, or economic profiles.[8] A similar mathematical structure results from New York City's Educational Option programs, which achieve a mix of students by imposing quotas on students with test scores (Abdulkadiroğlu, Pathak and Roth, 2005). Quotas may be based on the residence of applicants as well: The school choice program of Seoul, Korea, limits the percentage of seats allocated to applicants from outside the district.[9] A number of school choice programs in Japan have similar quotas based on residential areas as well.

Again constraints pertaining to individual students (i.e., "rows") can be organized as a hierarchy $\mathcal{H}_1$. All constraints pertaining to schools' capacities are organized as a separate hierarchy $\mathcal{H}_2$. Group-specific quotas can be handled by including sets of the form $N' \times \{a\}$ for $a \in O$ and $N' \subsetneq N$ in the second hierarchy $\mathcal{H}_2$. The ceiling $\overline{q}_{N'\times\{a\}}$ then determines the maximum number of agents school a can admit from group N'. Quotas on multiple groups can be imposed for each a without violating the hierarchy structure as long as they do not overlap with each other.[10] Moreover, a nested series of constraints can be accommodated.

[8] Abdulkadiroğlu and Sönmez (2003*b*) and Abdulkadiroğlu (2005) analyze assignment mechanisms under affirmative action constraints.

[9] See "Students' High School Choice in Seoul Outlined," Digital Chosun Ilbo, October 16, 2008 (http://english.chosun.com/w21data/html/news/200810/200810160016.html).

[10] In fact, an overlap of constraint sets can be accommodated with a small error. Suppose a school has maximal quotas for white and male at 60 and 55, respectively. Suppose an expected assignment assigns 40.5 white male, 14.5 black male, and 19.5 white female students to that school. Notice both ceilings are binding at this expected assignment. This expected assignment can be implemented recognizing only white and male, white and female, and male as the constraint sets, which then forms a hierarchy. Implementing with this modified constraint structure may violate the maximal quota for whites, since the constraint set for "white" is not included in the structure; for instance, the school may get 41 white male, 14 black male and 20 white female students. However, the violation is by only one student. In fact, the degree of violation is at most one when there is only one overlap of constraint sets. Such a small violation can often be tolerated in realistic

For instance, a school system can require that a school admit at most 50 students from district one, at most 50 students from district two, and at most 80 students from either district one or two.

It is also possible to accommodate both flexible-capacity constraints and group-specific quota constraints within the same hierarchy $\mathcal{H}_2$. Flexible-capacity constraints are defined on multiple columns of an expected assignment matrix X, whereas group-specific quota constraints are defined on subsets of single columns of X. Any subset of a single column will be a subset of or disjoint from any set of multiple columns.

2.1.4. *Course Allocation.* In course allocation, each student may enroll in multiple courses, but cannot receive more than one seat in any single course. Moreover, each student may have preference or feasibility constraints that limit the number of courses taken from certain sets. For example, scheduling constraints prohibit any student from taking two courses that meet during the same time slot. Or, a student might prefer to take at most two courses on finance, at most three on marketing, and at most four on finance or marketing in total.

Many such restrictions can be modeled using a bihierarchy, with $\mathcal{H}_1$ including all rows and $\mathcal{H}_2$ including all columns. Setting $\overline{q}_{\{(i,a)\}} = 1$ and $\overline{q}_{\{i\}\times O} > 1$ for each $i \in N$ and $a \in O$ ensures that each student i can enroll in multiple courses but be assigned to at most one seat in each course. Letting F and M be the sets of finance courses and marketing courses, respectively, if $\mathcal{H}_1$ contains $\{i\} \times F$,$\{i\} \times M$ and $\{i\} \times (F \cup M)$, then we can express the constraints "student i can take at most $\overline{q}_{\{i\}\times F}$ courses in finance, $\overline{q}_{\{i\}\times M}$ courses in marketing, and $\overline{q}_{\{i\}\times(F\cup M)}$ in finance and marketing combined." Scheduling constraints are handled similarly; for instance, F and M are sets of classes offered at different times (e.g., Friday morning and Monday morning). It may be impossible, however, to express both subject and scheduling constraints while still maintaining a bihierarchy constraint structure.

Note that the flexible production and group-specific quota constraints described in Sections 2.1.2-2.1.3 can also be incorporated into the course allocation problem without jeopardizing the bihierarchical structure. These constraints can be included in $\mathcal{H}_2$, while the preference and scheduling constraints described above can be included in $\mathcal{H}_1$. So long as $\mathcal{H}_1$ and $\mathcal{H}_2$ are both hierarchies, $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$ is a bihierarchy.

2.1.5. *Interleague Play Matchup Design.* Some professional sports associations, including Major League Baseball (MLB) and the National Football League (NFL), have two separate leagues. In MLB, teams in the American League (AL) and National League (NL) had traditionally played against teams only within their own league during the regular season,

controlled-choice environments. In case the quotas are rigid, the quota can be set more conservatively; for instance in the above example, the quota for whites can be set at 59 instead of 60.

but play across the AL and NL, called interleague play, was introduced in 1997.[11] Unlike the intraleague games, the number of interleague games is relatively small, and this can make the indivisibility problem particularly difficult to deal with in designing the matchups. For example, suppose there are two leagues, N and O, each with 9 teams. Suppose each team must play 15 games against teams in the other league. Ignoring indivisibility, a fair matchup will require each team in a league to play every team in the other league the same number of times, that is, $15/9 \approx 1.67$ times. Of course, this fractional matchup itself is infeasible, but one can implement this expected matchup as a convex combination of feasible matchups. In doing so, one can also specify additional constraints: e.g., each team in N has a geographic rival in O, and they must play twice; teams in each league must face opponents in the other league of similar difficulty. Specifically, one could require each team to play at least 4 games with the top 3 teams, 4 games with the middle 3 teams and 4 games with the bottom 3 teams of the other league. It is not difficult to see that the resulting constraint structure forms a bihierarchy. Our approach can produce a feasible matchup that implements the uniform expected assignment satisfying these additional constraints.

2.2. **Necessity of a bihierarchical constraint structure.** Theorem 1 shows that bihierarchy is sufficient for universal implementability. This section identifies a sense in which it is also necessary. Doing so also provides an intuition about the role bihierarchy plays for implementation of expected assignments. We begin with an example of a non-bihierarchical constraint structure that is not universally implementable.

Example 1. *Consider the following environment with two objects a, b and two agents $1, 2$, and the constraint structure $\mathcal{H}$ that consists of the "first row" $\{(1,a),(1,b)\}$, the "first column" $\{(1,a),(2,a)\}$, and a "diagonal set" $\{(1,b),(2,a)\}$ (in addition to all singleton sets). Clearly, this constraint structure is not a bihierarchy as there is no way to partition it into two hierarchies. Suppose each constraint set in $\mathcal{H}$ has a common floor and ceiling quota of one. The following expected assignment*

$$X = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$$

cannot be implemented as a lottery over feasible pure assignments.[12] *To see this, first observe that the lottery implementing X must choose with positive probability a pure assignment $\overline{X}$ in which $\overline{x}_{1a} = 1$. Since the first row has a quota of one, it follows that $\overline{x}_{1b} = 0$. Since the*

[11] See "Interleague play", Wikipedia (http://en.wikipedia.org/wiki/Interleagueplay).

[12] Notationally, the convention throughout the paper is that the ith row of the expected assignment matrix from the top corresponds to agent i while the first column from the left corresponds to object a, the second column corresponds to object b, and so on.

diagonal set has a quota of one, it follows that $\overline{x}_{2a} = 1$*. This is a contradiction because the quota for the first column is violated at* $\overline{X}$ *since* $\overline{x}_{\{(1,a),(2,a)\}} = \overline{x}_{1a} + \overline{x}_{2a} = 2$*.*

Example 1 suggests that the failure of implementability is caused by a "cycle" formed by an odd number of constraint sets. In the above example, for instance, a cycle formed by three constraint sets (the first row, the first column, and the diagonal set $\{(1,b),(2,a)\}$) leads to a situation where at least one of the constraints is violated. Generalizing this idea, we say that a sequence of constraint sets $(S_1, \ldots, S_l)$ in $\mathcal{H}$ is an **odd cycle** if l is odd and there exists a sequence of agent-object pairs $(s_1, \ldots, s_l)$ in $N \times O$ such that for each $i = 1, \ldots, l$, we have $s_i \in S_i \cap S_{i+1}$ and $x_i \notin S_j$ for any $j \neq i, i+1$, where subscript $l+1$ is understood to be 1. An argument generalizing the above example yields the following (a formal proof is in the Appendix).

Lemma 1. (Odd Cycles) *If a constraint structure contains an odd cycle, then it is not universally implementable.*

An important role of the bihierarchy is to rule out odd cycles. To see this, suppose for contradiction that a bihierarchy $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$ contains an odd cycle $(S_1, \ldots, S_l)$. Assume $S_1 \in \mathcal{H}_1$ without loss of generality. Then, S_2 must belong to $\mathcal{H}_2$ since $S_1 \cap S_2 \neq \emptyset$ and neither is a subset of the other (since $s_1 \in S_1 \cap S_2$, $s_2 \in S_2 \setminus S_1$ and $s_l \in S_1 \setminus S_2$). Arguing in the same fashion, S_3 must be in $\mathcal{H}_1$, S_4 in $\mathcal{H}_2$, …, and S_l must be in $\mathcal{H}_1$ since l is an odd number. But $S_l \cap S_1 \neq \emptyset$ and neither is a subset of the other. So $\mathcal{H}_1$ cannot be a hierarchy, and $\mathcal{H}$ cannot be a bihierarchy, a contradiction.

Is bihierarchy necessary for universal implementation? It turns out this is not the case; nor is it the case that ruling out odd cycles is sufficient for universal implementation.[13] Figure 2 depicts how the sets of constraint structures satisfying different properties relate to

[13] To see that bihierarchy is not necessary, consider an environment with 2 objects and 2 agents as before, but let

$$\mathcal{H} = \{\{(1,a),(1,b)\}, \{(1,a),(2,a)\}, \{(1,a),(2,b)\}\},$$

and the floor and ceiling quotas for each constraint set be one. Note $\mathcal{H}$ is not a bihierachy. Yet, any expected assignment

$$X = \begin{pmatrix} s & t \\ t & t \end{pmatrix},$$

with $s + t = 1$, can be decomposed by a convex combination of pure assignments as

$$X = \begin{pmatrix} s & t \\ t & t \end{pmatrix} = s \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + t \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

Next, we show that an absence of odd cycles is not sufficient for universal implementability. Consider $\mathcal{H} = \{\{(1,a),(1,b)\}, \{(1,a),(2,a)\}, \{(1,a),(2,b)\}, \{(1,a),(1,b),(2,a),(2,b)\}\}$. This structure does not contain an odd cycle (and it is not a bihierarchy). Assume the quota for each of the first three sets is one and the quota for the last set is two. Now consider the expected assignment X of Example 1. Even though X satisfies the quotas, it is not implementable.

each other. In particular, there are gaps between the set of constraint structures containing no odd cycles, the set of those universally implementable, and the set of those that are bihierarchical.

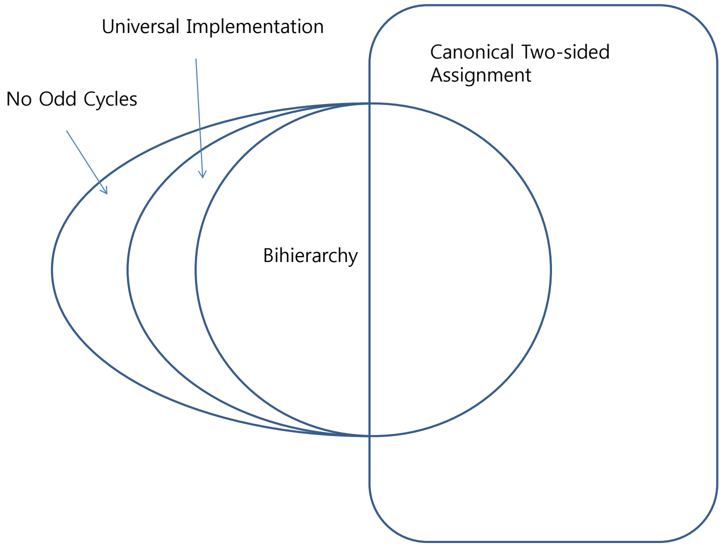


Figure 2: Constraint structures satisfying different properties

We now show that these gaps disappear for an important class of constraint structures (depicted on the right side of Figure 2). In a two-sided assignment problem, there are often quotas for each individual agent and quotas for each object. We say that $\mathcal{H}$ is a **canonical two-sided constraint structure** if $\mathcal{H}$ contains all "rows" (i.e., sets of the form $\{(i,a)|a \in O\}$ for each $i \in N$) and all "columns" (i.e., sets of the form $\{(i,a)|i \in N\}$ for each $a \in O$). The next result demonstrates that bihierarchy is necessary for universal implementability for such constraint structures.[14]

Theorem 2. (Necessity) *If a canonical two-sided constraint structure is not a bihierarchy, then it is not universally implementable.*

The formal proof of Theorem 2 is in the Appendix. The basic strategy of the proof is to show that there exists an odd cycle whenever a canonical two-sided constraint structure is not a bihierarchy.

[14] One may wonder if the restriction to canonical two-sided constraint structures has any real bite in light of the fact that one can seemingly convert any $\mathcal{H}$ into one containing each row and column simply by imposing non-binding quotas, e.g., $\underline{q}_S = -\infty$ and $\overline{q}_S = \infty$. This cosmetic conversion does not alter the fact that no constraint is ever binding for the added set, however. Recall that the notion of universal implementability requires the ability to implement an expected assignment subject to the tightest possible constraints on each set in $\mathcal{H}$. Hence, the sets with non-binding constraints cannot be added to a constraint structure in this manner.

Thus, in the context of typical two-sided assignment and matching problems, bihierarchy is both necessary and sufficient for universal implementation. We now turn to applications.

3. A Generalization of The Probabilistic Serial Mechanism for Assignment with Single-unit Demand

In this section, we consider a problem of assigning indivisible objects to agents who can consume at most one object each. Examples include university housing allocation, public housing allocation, office assignment, and student placement in public schools.

A common method for allocating objects in such a setting is the **random priority** mechanism. *In this mechanism, every agent reports preference rankings of the objects. The mechanism designer then randomly orders the agents with equal probability. The first agent in the realized order receives her stated favorite (the most preferred) object, the next agent receives his stated favorite object among the remaining ones, and so on.* Random priority is strategy-proof, that is, reporting ordinal preferences truthfully is a weakly dominant strategy for every agent. Moreover, random priority is ex-post efficient, that is, every pure assignment that occurs with positive probability under the mechanism is Pareto efficient.

Despite its many advantages, the random priority mechanism may entail unambiguous efficiency loss *ex ante*. Adapting an example by Bogomolnaia and Moulin (2001), suppose that there are two types of objects a and b with one copy each and the "null object" $\emptyset$ representing the outside option. There are four agents $1, 2, 3$ and 4, where agents 1 and 2 prefer a to b to $\emptyset$ while agents 3 and 4 prefer b to a to $\emptyset$. By calculation, the random priority mechanism results in the expected assignment

$$X = \begin{pmatrix} 5/12 & 1/12 & 1/2 \\ 5/12 & 1/12 & 1/2 \\ 1/12 & 5/12 & 1/2 \\ 1/12 & 5/12 & 1/2 \end{pmatrix}.$$

This assignment entails an unambiguous efficiency loss. Notice first that every agent consumes the less preferred of the two proper objects with positive probability. This happens since two agents of the same preference (e.g., agents 1 and 2) are chosen with positive probability to the be first two in the serial order, in which case the second agent will claim the less preferred of the two proper objects. Clearly, it will benefit all if agents 1 and 2 can trade off $1/12$ share of b for the same share of a with agents 3 and 4. In other words, every agent prefers an alternative expected assignment,

$$X' = \begin{pmatrix} 1/2 & 0 & 1/2 \\ 1/2 & 0 & 1/2 \\ 0 & 1/2 & 1/2 \\ 0 & 1/2 & 1/2 \end{pmatrix}.$$

An expected assignment is said to be **ordinally efficient** if it is not first-order stochastically dominated for all agents by some other expected assignment. The example implies that random priority may result in an ordinally inefficient expected assignment.

The **probabilistic serial mechanism**, introduced by Bogomolnaia and Moulin (2001) in the simple one-to-one assignment setting, eliminates this form of inefficiency. Imagine that each indivisible object is a divisible object of probability shares: If an agent receives fraction p of an object, we interpret that she receives the object with probability p. Given reported preferences, consider the following "eating algorithm." *Time runs continuously from 0 to 1. At every point in time, each agent "eats" her reported favorite object with speed one among those that have not been completely eaten up. At time $t = 1$, each agent is endowed with probability shares of objects. The probabilistic serial assignment is defined as the resulting probability shares.*

In the above example, agents 1 and 2 start eating a and agents 3 and 4 start eating b at $t = 0$ in the eating algorithm. Since two agents are consuming one unit of each object, both a and b are eaten away at time $t = \frac{1}{2}$. As no (proper) object remains, agents consume the null object between $t = \frac{1}{2}$ and $t = 1$. Thus the resulting probabilistic serial assignment is given by X' defined above. In particular, the probabilistic serial mechanism eliminates the inefficiency that was present under random priority. More generally, Bogomolnaia and Moulin (2001) show that the probabilistic serial random assignment is ordinally efficient with respect to any reported preferences.[15]

The main goal of this section is to generalize the probabilistic serial mechanism to accommodate constraints absent in the simple setting. To begin, we consider our basic setup with

[15] The contribution of Bogomolnaia and Moulin has led to much subsequent work on random assignment mechanisms for single-unit assignment problems. The probabilistic serial mechanism is generalized to allow for weak preferences and existing property rights by Katta and Sethuraman (2006) and Yilmaz (2009). Kesten (2007) defines two random assignment mechanisms and shows that these mechanisms are equivalent to the probabilistic serial mechanism. Ordinal efficiency is characterized by Abdulkadiroğlu and Sönmez (2003*a*), McLennan (2002) and Manea (2006). Behavior of the random priority and probabilistic serial mechanisms in large markets is studied by Kojima and Manea (2008), Manea (2009) and Che and Kojima (2008). In the scheduling problem (a special case of the current environment), Crès and Moulin (2001) show that the probabilistic serial mechanism is group strategy-proof and first-order stochastically dominates the random priority mechanism, and Bogomolnaia and Moulin (2002) give two characterizations of the probabilistic serial mechanism.

agents N and objects O, where O now contains a "null" object $\emptyset$ with unlimited supply.[16] We then consider a bihierarchy constraint structure $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$ such that $\mathcal{H}_1$ is composed of all rows while $\mathcal{H}_2$ includes (but is not restricted to) all columns. We assume that $\underline{q}_{\{i\}\times O} = \overline{q}_{\{i\}\times O} = 1$ for all $i \in N$, that is, each agent obtains exactly one object, rather than at most one. This is without loss of generality since O contains $\emptyset$. We assume that $\underline{q}_S = 0$ for any $S \in \mathcal{H}$ that is not a row, that is, there are no other floor constraints. The ceiling quota for each object a, $\overline{q}_{N\times\{a\}}$, can be an arbitrary nonnegative integer. Recall that an expected assignment X is said to satisfy quotas $\mathbf{q}$ if $\underline{q}_S \leq \sum_{(i,a)\in S} x_{ia} \leq \overline{q}_S$ for each $S \in \mathcal{H}$. Each agent i has a strict preference $\succ_i$ over the set of objects. We write $a \succeq_i b$ if either $a \succ_i b$ or $a = b$ holds. We write $\succ$ for $(\succ_i)_{i\in N}$ and $\succ_{-i}$ for $(\succ_j)_{j\in N\setminus\{i\}}$.

As mentioned earlier, the bihierarchy structure in this section accommodates a range of practical situations faced by a mechanism designer. First, the objects may be produced endogenously based on the reported preferences of the agents, as in the case of school choice with flexible capacity (Section 2.1.2). Second, a mechanism designer may need to treat different groups of agents differently, as in the case of school choice with group-specific quotas (Section 2.1.3).

Now we introduce the **generalized probabilistic serial** mechanism. As in BM, the basic idea is to regard each indivisible object as a divisible object of "probability shares." More specifically, the algorithm is described as follows: *Time runs continuously from 0 to 1. At every point in time, each agent "eats" her reported favorite object with speed one among those that are "available" at that instance, and the probabilistic serial assignment is defined as the probability shares eaten by each agent by time* 1. In order to obtain an implementable expected assignment in the presence of additional constraints, however, we modify the definition of the algorithm. More specifically, *we say that object a is "available" to agent i if and only if, for every constraint set S such that $(i,a) \in S$, the total amount of probability shares eaten away within S (the sum, over every agent-object pair $(j,b) \in S$, of shares of b eaten by j) is less than its ceiling quota $\overline{q}_S$.* This algorithm is formally defined in Appendix A. Given reported preferences $\succ$, the generalized probabilistic serial assignment is denoted $PS(\succ)$.

Note that a few modifications are made in the definition of the algorithm from the version of Bogomolnaia and Moulin (2001). First, we specify availability of objects with respect to both agents and objects in order to accommodate complex constraints such as affirmative action. Second, we need to keep track of multiple constraints for each agent-object pair (i,a)

[16] Formally, we assume that $\overline{q}_S = +\infty$ for each constraint set S that is not a row and has a nonempty intersection with $N \times \{\emptyset\}$.

during the algorithm, since there are potentially multiple constraints that would make the consumption of the object a by the agent i no longer feasible.

Recall that the constraint structure in this section is a bihierarchy. Building on this observation and our analysis from the previous section, we point out that any expected assignment produced by the generalized probabilistic serial mechanism is implementable.

Corollary 2. *For any preference profile $\succ$, the generalized probabilistic serial assignment $PS(\succ)$ is implementable.*

Proof. The result follows immediately from Theorem 1, because the constraint structure under consideration forms a bihierarchy and $PS(\succ)$ satisfies all quotas associated with that constraint structure by construction. □

In this sense, the mechanism is well-defined as a random assignment mechanism in the current setting.

3.1. **Properties of The Generalized Probabilistic Serial Mechanism.** We introduce the ordinal efficiency concept in our setup. A lottery $\mathbf{x}_i = [x_{ia}]_{a \in O}$ for an agent (first-order) **stochastically dominates** another lottery $\mathbf{x}'_i = [x'_{ia}]_{a \in O}$ **at** $\succ_i$ if

$$\sum_{b \succeq_i a} x_{ib} \geq \sum_{b \succeq_i a} x'_{ib},$$

for every object $a \in O$, and $\mathbf{x}_i$ **strictly stochastically dominates** $\mathbf{x}'_i$ if the former stochastically dominates the latter and $\mathbf{x}_i \neq \mathbf{x}'_i$.

An expected assignment $X = [\mathbf{x}_i]_{i \in N}$ **ordinally dominates** another expected assignment $X' = [\mathbf{x}'_i]_{i \in N}$ at $\succ$ if $X' \neq X$, and, for each i, $\mathbf{x}_i$ stochastically dominates $\mathbf{x}'_i$ at $\succ_i$. If X ordinally dominates X' at $\succ$, then every agent i prefers $\mathbf{x}_i$ to $\mathbf{x}'_i$ according to any expected utility function with utility index consistent with $\succ_i$. An expected assignment that satisfies $\mathbf{q}$ is **ordinally efficient at** $\succ$ if it is not ordinally dominated at $\succ$ by any other expected assignment that satisfies $\mathbf{q}$. Note that our model allows for a variety of constraints, so the current notion has the flavor of "constrained efficiency" in that the efficiency is defined within the set of expected assignments satisfying the quota constraints.

Bogomolnaia and Moulin (2001) show that the probabilistic serial mechanism results in an ordinally efficient expected assignment in their setting. Their result can be generalized to our setting although its proof requires new arguments.[17]

[17] In BM's environment, ordinal efficiency is equivalent to the nonexistence of a Pareto-improving trade in probability shares among agents (in the sense of leading to a ordinally dominating expected assignment). This enables a characterization of ordinal efficiency in terms of a certain binary relation over objects, which is crucial in BM's proof of their mechanism's ordinal efficiency. There are two main difficulties for generalizing

Theorem 3. *For any preference profile $\succ$, the generalized probabilistic serial expected assignment $PS(\succ)$ is ordinally efficient at $\succ$.*

Bogomolnaia and Moulin (2001) also show that the probabilistic serial mechanism is fair in a specific sense in their setting. Formally, an expected assignment $X = [\mathbf{x}_i]_{i \in N}$ is **envy-free at** $\succ$ if $\mathbf{x}_i$ stochastically dominates $\mathbf{x}_j$ at $\succ_i$ for every $i, j \in N$. It turns out that the generalized probabilistic serial assignment is not necessarily envy-free in our environment. To see this point, consider the following environment: there are three agents $1, 2$, and 3, and two units of object a plus a null object $\emptyset$. Even though there are two copies of a, only one of them can be assigned to to 1 or 2. Suppose all agents prefer a over the null object. Then, by a simple calculation the generalized probabilistic serial expected assignment $PS(\succ)$ is given by

$$PS(\succ) = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \\ 1 & 0 \end{pmatrix}.$$

Note that $PS(\succ)$ is not envy-free since $PS_3(\succ)$ is not stochastically dominated by $PS_1(\succ)$ with respect to $\succ_1$ (indeed, $PS_3(\succ)$ strictly stochastically dominates $PS_1(\succ)$ in this example). However, there is a sense in which the above expected assignment should not be considered unfair despite the existence of envy. To see this point, note that it is infeasible to assign object a to agent 1 with higher probability simply by moving some probability share of a from agent 3 to agent 1, because such a change would violate the ceiling quota on $\{1, 2\} \times \{a\}$. In that sense the envy is based on a desire of agent 1 that cannot be feasibly accommodated.

Motivated by the above example, we introduce the following concept. Expected assignment X entails **no feasible envy at** $\succ$ if, whenever $\mathbf{x}_i$ does not stochastically dominate $\mathbf{x}_j$ at $\succ_i$, it is impossible to feasibly reassign agent i to $\mathbf{x}_j$ while keeping intact expected assignments of all agents except possibly of agent j; i.e., if no expected assignment Y defined by $\mathbf{y}_i = \mathbf{x}_j$ and $\mathbf{y}_k = \mathbf{x}_k$ for all $k \neq i, j$, satisfies $\mathbf{q}$.[18] Thus the above definition requires that either i does not envy j, or an alternative expected assignment in which i receives j's lottery violates

the result to our setting. First, not all trades in probability shares are feasible because the new expected assignment may violate constraints such as group-specific quotas. Second, the nonexistence of a Pareto-improving trade does not imply ordinal efficiency because, thanks to flexible capacity, a different aggregate supply of objects may exist that makes every agent better off. These differences make it impossible to directly apply BM's technique. We address these complications by defining a new binary relation over agent-object pairs. See Appendix F for details.

[18] Note we do not put any restriction on agent j's lottery in considering feasible reassignment. Not putting any restriction on j's lottery can only weaken the condition for a feasible reassignment, and thus can only strengthen the notion of envy-freeness, in comparison with imposing restrictions such as assigning $\mathbf{x}_i$ to agent j in the new assignment.

quotas. In the above example, $PS(\succ)$ entails no feasible envy. This property turns out to hold generally, as stated below.

Theorem 4. *For any preference profile $\succ$, the generalized probabilistic serial expected assignment $PS(\succ)$ entails no feasible envy at $\succ$.*

Neither ordinal efficiency nor no feasible envy is satisfied by random priority even in the simplest setting of one-to-one matching (Bogomolnaia and Moulin, 2001).

Unfortunately the probabilistic serial mechanism is not strategy-proof, that is, there are situations in which an agent is made better off by misreporting her preferences. However, Bogomolnaia and Moulin (2001) show that the probabilistic serial mechanism is **weakly strategy-proof** in their setting, that is, an agent cannot misstate his preferences and obtain an expected assignment that strictly stochastically dominates the one obtained under truth-telling. Formally, we claim that the generalized probabilistic serial mechanism is weakly strategy-proof, that is, there exist no $\succ$, $i \in N$ and $\succ'_i$ such that $PS_i(\succ'_i, \succ_{-i})$ strictly stochastically dominates $PS_i(\succ)$ at $\succ$ in our more general environment.[19]

Theorem 5. *The generalized probabilistic serial mechanism is weakly strategy-proof.*

Proof. The proof is an adaptation of Proposition 1 of Bogomolnaia and Moulin (2001) and we omit the proof. □

One limitation of our generalization is that the algorithm is defined only for cases with *maximum* quotas: The minimum quota for each group must be zero. In the context of school choice, this precludes the administrator from requiring that at least a certain number of students from a group attend a particular school. Despite this limitation, administrative goals can often be sufficiently represented using maximum quotas alone. For instance, if there are two groups of students, “rich” and “poor”, a requirement that at least a certain number of poor students attend some highly desirable school might be adequately replaced by a maximum quota on the number of rich students who attend.

4. A Generalization of The Pseudo-market Mechanism for Assignment with Multi-Unit Demand

In a seminal paper, Hylland and Zeckhauser (1979) propose an ex-ante efficient mechanism for the problem of assigning n objects amongst n agents with single-unit demand. Based

[19] Kojima and Manea (2008) show that truthtelling becomes a dominant strategy for a sufficiently large market under the probabilistic serial mechanism in a simpler environment than the current one. Showing a similar claim in our environment is beyond the scope of this paper, but we conjecture that the argument can be extended.

on the old idea of a competitive equilibrium from equal incomes, the mechanism can be described as follows. *Agents report their von Neumann-Morgenstern preferences over individual objects. Each agent is allocated an equal budget of an artificial currency. The mechanism then computes a competitive equilibrium of this market, where the objects being priced and allocated are probability shares of objects.* As the allocations are based on a competitive equilibrium, each agent is allocated a probability share profile that maximizes her expected utility subject to her budget constraint at the competitive equilibrium prices. This expected assignment is ex ante efficient by a version of the first welfare theorem. It is also envy-free in the sense that each agent weakly prefers her lottery over anyone else's according to her expected utility, because all agents have identical budget constraints. The resulting expected assignment can be implemented by appeal to the Birkhoff-von Neumann theorem.

By contrast, designing desirable mechanisms has been challenging in problems where agents have multi-unit demand, as in the assignment of course schedules to students or assignment of athletes to sports teams.[20] For instance, axiomatic results on the problem are mostly negative,[21] and the mechanisms used in practice suffer from inefficiency and fairness problems.[22] In this section, we consider a generalization of a pseudo-market mechanism that is ex-ante efficient and interim envy-free like HZ's to cases in which agents have multi-unit demand.[23]

Initially, we assume that the agent's preferences are additive subject to a set of constraints (we illustrate how to extend the framework to more general preferences in Section 4.2.) Specifically, for each agent i, there exists a cardinal value $v_{ia} \in \mathbb{R}$ for each $a \in O$, a collection $\mathcal{H}_i$ of hierarchical sets of the form $S_i = \{i\} \times O'$, $O' \subset O$, and ceiling quotas $\overline{q}_{S_i} \geq 0$ for each $S_i \in \mathcal{H}_i$, such that the agent's utility from pure consumption bundle $\overline{\mathbf{x}}_i = [\overline{x}_{ia}] \in \mathbb{Z}^{|O|}$ is

$$u_i(\overline{\mathbf{x}}_i) = \begin{cases} \sum_{a\in O} \overline{x}_{ia} v_{ia} & \text{if } 0 \leq \sum_{(i,a)\in S_i} \overline{x}_{ia} \leq \overline{q}_{S_i}, \forall S_i \in \mathcal{H}_i \\ -\infty & \text{otherwise.} \end{cases} \tag{4.1}$$

[20] Similar problems include the assignment of tasks within an organization, the division of heirlooms and estates among heirs, and the allocation of access to jointly-owned scientific resources.

[21] Papai (2001) shows that sequential dictatorships are the only deterministic mechanisms that are nonbossy, strategy-proof, and Pareto optimal; dictatorships are unattractive for many applications because they are highly unfair *ex post*. Ehlers and Klaus (2003), Hatfield (2008), and Kojima (2008) provide similarly pessimistic results.

[22] See Sönmez and Ünver (2008) and Budish and Cantillon (2009).

[23] Budish (2009) proposes a mechanism that accommodates arbitrary ordinal preferences over schedules, but which is only approximately ex-post efficient. It too is based on the idea of CEEI, but use the framework to find an ex post sure assignment; by contrast, we use CEEI framework to find an "expected" assignment. Additional discussion on the tradeoffs between these two approaches can be found in Section 8.2 of Budish (2009).

In course allocation, the constraints may capture a student's desire to avoid taking more than a certain number of courses in a given subject area or during a given time slot. We set all floor constraints equal to zero; this will play a role in our proof that equilibrium prices exist. For technical convenience we also require that each agent has a unique bliss point.

This form of utility function can easily be extended to "expected consumption bundles." Given linearity of preferences, an agent's expected utility from any expected consumption bundle $\mathbf{x}_i \in \mathbb{R}^{|O|}$ can be expressed by the same formula as in (4.1).

We now define the **generalized pseudo-market mechanism.**

The Generalized Pseudo-Market Mechanism:

(1) Each agent i reports her cardinal object values and her consumption constraints, as described above.

(2) Let b^* be a positive number, which we interpret as the equal budget endowment of each agent in artificial currency. The mechanism computes a vector of nonnegative item prices $\mathbf{p}^* = (p_a)_{a\in O}$ and an expected assignment $X^* = [\mathbf{x}_i^*]_{i\in N}$ which clears the market in the sense below:

- Each agent i is allocated a (possibly fractional) consumption bundle $\mathbf{x}_i^*$ which maximizes her (reported) utility subject to the budget constraint, that is,

$$\mathbf{x}_i^* \in \arg\max_{\mathbf{x}_i} u_i(\mathbf{x}_i) \text{ subject to} \sum_{a\in O} p_a^* x_{ia} \leq b^*.$$

- The probability-shares market clears in the sense that

$$\begin{aligned}\sum_{i\in N} x_{ia}^* &\leq q_a \text{ for all } a \in O \text{ (object constraints)}\\ &< q_a \text{ only if } p_a^* = 0 \text{ (complementary slackness)}\end{aligned}$$

(3) The expected assignment X^* is implemented.

Given our interpretation that the mechanism simulates a competitive market, we refer to a price vector $\mathbf{p}^*$ above as competitive equilibrium prices. To show that the mechanism is well defined we need the following results. The first result is that a competitive equilibrium exists in this problem.

Theorem 6. *There exist competitive equilibrium prices $\mathbf{p}^*$ and expected assignment X^* in the sense defined in Step (2) of the generalized pseudo-market mechanism.*

Proof. See the Appendix. □

Standard competitive equilibrium existence results cannot be applied here due to the failure of local non-satiation; for instance, a student may have a most-preferred schedule of courses that constitutes a bliss point where the student is satiated.[24]

The next result shows that the expected assignment produced by the mechanism can be implemented.

Corollary 3. *The expected assignment X^* produced in Step (3) of the generalized pseudo-market mechanism is implementable. Moreover, there exists a lottery over pure assignments implementing X^* such that the expected utility of each agent i is $u_i(\mathbf{x}_i^*)$.*

Proof. Follows immediately from Theorem 1, because the agents' constraints form one hierarchy, while the objects' capacity constraints form a second hierarchy. □

4.1. **Properties of the Generalized Pseudo-Market Mechanism.** The original HZ mechanism is attractive for single-unit assignment because it is ex-ante efficient and interim envy free. We show that these properties carry over to this more general environment.

Theorem 7. *The expected assignment X^* is ex-ante Pareto efficient, and every realization of the lottery is ex-post Pareto efficient.*

Proof. Suppose there exists an expected assignment $\tilde{X} = [\tilde{\mathbf{x}}_i]_{i \in N}$ that Pareto improves upon X^*. If $u_i(\tilde{\mathbf{x}}_i) > u_i(\mathbf{x}_i^*)$ then revealed preference implies that $\mathbf{p}^* \cdot \tilde{\mathbf{x}}_i > \mathbf{p}^* \cdot \mathbf{x}_i^*$. Suppose $u_i(\tilde{\mathbf{x}}_i) = u_i(\mathbf{x}_i^*)$. We claim that $\mathbf{p}^* \cdot \tilde{\mathbf{x}}_i \geq \mathbf{p}^* \cdot \mathbf{x}_i^*$. Towards a contradiction, suppose that $\mathbf{p}^* \cdot \tilde{\mathbf{x}}_i < \mathbf{p}^* \cdot \mathbf{x}_i^*$. Let $\hat{\mathbf{x}}_i$ denote i's bliss point. From our assumption that bliss points are unique it follows that $u_i(\hat{\mathbf{x}}_i) > u_i(\tilde{\mathbf{x}}_i) = u_i(\mathbf{x}_i^*)$. Since $\mathbf{p}^* \cdot \tilde{\mathbf{x}}_i < \mathbf{p}^* \cdot \mathbf{x}_i^* \leq b^*$ there exists $\lambda \in (0,1)$ such that $\mathbf{p}^* \cdot (\lambda \hat{\mathbf{x}}_i + (1-\lambda)\tilde{\mathbf{x}}_i) \leq b^*$. By concavity of u and uniqueness of the bliss point, $u_i(\lambda \hat{\mathbf{x}}_i + (1-\lambda)\tilde{\mathbf{x}}_i) > u_i(\tilde{\mathbf{x}}_i) = u_i(\mathbf{x}_i^*)$, which contradicts $\mathbf{x}_i^*$ being a utility maximizer for i in Step (2) of the generalized pseudo-market mechanism. Hence $\mathbf{p}^* \cdot \tilde{\mathbf{x}}_i \geq \mathbf{p}^* \cdot \mathbf{x}_i^*$.

From the above and from the assumption that $\tilde{X}$ is a Pareto improvement on X^*, we have established that $\mathbf{p}^* \cdot \tilde{\mathbf{x}}_i \geq \mathbf{p}^* \cdot \mathbf{x}_i^*$ for all i with at least one strict. Thus $\sum_i \mathbf{p}^* \cdot \tilde{\mathbf{x}}_i > \sum_i \mathbf{p}^* \cdot \mathbf{x}_i^*$. It then follows that there exists $a \in O$ such that $p_a^* > 0$ and $\sum_i \tilde{x}_{ia} > \sum_i x_{ia}^*$. But this contradicts the complementary slackness condition of the pseudo market defined earlier, which implies that $\sum_i x_{ia}^* = q_a$. We thus conclude that X^* is (ex-ante) Pareto efficient.

[24] The bliss point can be in the interior of the agent's budget set if the prices of the objects in his bliss point are sufficiently low. Hylland and Zeckhauser (1979) discuss the failure of standard existence proof techniques in their footnote 14. By endowing agents with budgets in a fictitious currency, and by assuming all floor constraints to be zero, we avoid some of the difficulties that the failure of nonsatistion may pose. Hylland and Zeckhauser (1979) assume strictly positive floor constraints – in HZ, each agent requires *exactly* one object – and for this reason their method of proof is more involved and does not readily generalize to our environment.

Ex-ante efficiency immediately implies ex-post efficiency; if some realization of a lottery were ex-post inefficient, then by executing Pareto improvements for that realization we could generate an ex-ante Pareto improvement. □

Theorem 8. *The expected assignment X^* is interim envy free. That is, for any agents $i \neq j$, $u_i(\mathbf{x}_i^*) \geq u_i(\mathbf{x}_j^*)$.*

Proof. Follows immediately from the definition of the mechanism given that all agents have the same budget. □

Theorem 8 concerns interim fairness. The main result of Section 6 can be used to enhance *ex-post* fairness in cases where agents' bids take a simple additive-separable form without additional constraints.

Aside from accommodating multi-unit demand, our generalization of the pseudo-market mechanism allows agents to express several kinds of constraints that may be useful for practice. To fix ideas we focus on constraints specific to the problem of course allocation.

- *Scheduling Constraints:* Scheduling constraints can often be expressed by means of a hierarchy. One example is students at Harvard Business School, who require 10 courses per school year, of which 5 should be in each of the two semesters, and of which no more than one should meet at any given time. More generally, if there is a set of time slots, constraints of the form "no more than one course at any time slot" form a simple kind of hierarchy.

- *Curricular Constraints:* Students often seek variety in their schedules due to diminishing returns. Our language can accommodate constraints of the form "at most 2 courses in Finance," and it can also accommodate more elaborate constraints like "at most 2 courses in Finance, at most 2 courses in Marketing, and at most 3 courses in Finance or Marketing."

A limitation of our formulation is that we may not be able to simultaneously accommodate multiple kinds of the constraints described above. While ruling out some practical applications, this restriction is necessitated by implementability. For instance, if there is a Finance course and a Marketing course that meet at time slot 1 (f_1, m_1), and another Finance course and another Marketing course that meet at time slot 2 (f_2, m_2), then scheduling constraints on $\{f_1, m_1\}$ and $\{f_2, m_2\}$ and curricular constraints on $\{f_1, f_2\}$ and $\{m_1, m_2\}$ cannot coexist in the same hierarchy.

4.2. **Accommodating nonlinear preferences.** So far, we have assumed that agents' preference are linear (additive) on a domain specified by certain constraints. This restriction is necessitated by the methodology that focuses on "expected" assignments as a primitive design variable. Focusing on expected assignments is without loss when the agents have linear preferences, but not if their preferences are nonlinear. For instance, if an agent values the first unit of a good at \$12, the second unit at \$8, and the third at \$4, consuming 1.5 units of the good in expectation can mean different expected utility levels, depending on the precise distribution.[25] Yet, certain nonlinear preferences seem quite relevant in real applications. In course allocation, for instance, diminishing marginal utilities for similar courses may be natural; an MBA student may value a course in finance more when he takes fewer other finance courses. Other kinds of non-additivity may also prove useful in some applications (as described below).

Fortunately, we can accommodate certain substitutable preferences, using the idea of Milgrom (2009)'s integer assignment messages. The idea is to encode nonlinear preferences into linear objectives by enriching the agents' message spaces. To illustrate, suppose an agent has two finance courses f and f', and she values each course more when it is the only finance course taken. For concreteness, her values of taking f and f' as the sole finance course are 10 and 8, respectively, and her value of taking both courses is 15. Our method involves defining two separate "roles" for a given course, say f, based on how it is used— whether it is used as the first or second finance course taken. Let f_1 and f_2 denote these two "usages" of course f, and let f_1' denote simply her use of f' as the first course. With this expanded language, the agent can encode her (nonlinear) preferences by assigning appropriate "additive" values to the separate usages of the courses, together with an appropriate set of constraints. For concreteness, the agent in this example can submit values $v_{f_1} = 10, v_{f_2} = 15 - 8 = 7, v_{f_1'} = 8$, along with constraints that she can take at most one out of f_1 and f_1' and at most one out of f_1 and f_2, ensuring that whenever she takes two courses, she consumes only bundle (f_1', f_2) for which she assigns the right value of 15.[26]

Formally, for each object a, we allow each agent i to define role r for which she will use a, and to submit associated utility value v_{ira}, interpreted as her value of using a in role r. With this enriched language, we must also extend the expected assignment matrix as well. We do so by associating each pair (i, r) with a separate row, and associating each good a, just as before, with a column. Intuitively, we have each agent "own" several rows of the expected

[25] If one implements 1.5 units via lottery $\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 2$, her expected utility will be \$16, but if it is implemented via lottery $\frac{3}{4} \cdot 1 + \frac{1}{4} \cdot 3$, her expected utility will be \$15.

[26] Observe that this language does not allow the agent to ensure that whenever she consumes only one course, it must be either f_1 or f_1'. That is, she cannot exclude consuming f_2 alone using constraints (that are bihierarchical, as will be seen below). In the case of diminishing utility, however, this is not a problem since she assigned higher value to f_1 than f_2; so utility maximization in our equilibrium will ensure that f_2 cannot be consumed alone (i.e., without consuming f_1'). But the inability to exclude consuming f_2 alone presents a problem for encoding "complementary" or increasing marginal utility. Suppose instead that the agent derives a value of 20 from taking the two finance courses. If she submits $v_{f_2} = 20 - 8 = 12$, our pseudo-market mechanism may return an assignment where the agent consumes f_2 alone, supposedly realizing value of 12, although if she takes only f, her value is 10 and not 12. Complementary preferences may also present difficulties with guaranteeing existence of the pseudo-market equilibrium.

assignment matrix, with the interpretation that individual rows represent different "roles" that each object can play in generating utility. Along with the additive values $(v_{ira})_{(r,a)}$, agent i submits a set of ceiling/floor quotas on subsets of (i, r, a)'s, with the restriction that the constraint sets form a bihierarchy, and one of the hierarchies contains only subsets of columns. Then, since constraint sets submitted by different agents are disjoint, the entire constraint structure (across all the agents) also forms a bihierarchy.[27] Given the bihierarchical constraints, the pseudo-market mechanism can be extended in a rather straightforward way, as described in Appendix B.

Although integer assignment messages induce substitutable preferences (Milgrom, 2009), it is unknown exactly what class of substitutable preferences can be expressed via such a language. But one can see that the diminishing marginal utility in the example can be expressed in this way; in particular, the bihierarchy condition is satisfied. The constraint sets required to express diminishing marginal utility form multiple rows or sub-columns, satisfying the bihierarchy condition not only within the rows associated with that agent but also within the entire matrix.

The enriched language can be used to capture other constraints and preferences. For instance, suppose that a college football team seeks to recruit a high-school player to play wide-receiver or cornerback. The player's utility value is 50 in the role of wide-receiver or 30 in the role of defensive cornerback, but he will fill only one role. The team might further refine the roles (starting receiver versus back-up receiver) to describe its needs more completely. The degree to which another player substitutes for him depends on the values of other potential receivers and cornerbacks and their value in the other roles that they might play. Rich patterns of substitutable preferences like this can be incorporated into the current framework.

5. The Utility Guarantee for Multi-Unit Assignment and Matching

We call our third application the "utility guarantee." In general, there can be many ways to implement a given expected assignment, and the choice among them may be important. To fix ideas, suppose that two agents are to divide $2n$ objects (with $n \geq 2$), that the agents' preferences are additive, and that agents' ordinal rankings of the items are the same.[28] Suppose the "fair" expected assignment specifies that each agent receive half of each object. One way to implement this is to randomly choose n objects to assign to the first agent and

[27] More precisely, we can form one hierarchy by placing all sub-column constraints submitted by the agents and all column constraints resulting from supply constraints of the objects, and aother hierarchy by placing all other constraints submitted by the agents.

[28] We described a special case with $n = 2$ in the introduction.

then give the remaining n objects to the other. This method, however, could entail a highly "unfair" outcome *ex post*, in which one agent gets the n best objects and the other gets the n worst ones.

Based on Theorem 1 we provide a method to implement a given expected assignment with a small variation in realized utility. Formally, consider an environment with constraint structure $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$ such that $\mathcal{H}_1$ is a hierarchy and $\mathcal{H}_2$ is a hierarchy composed of all rows, and expected assignment X that satisfies given quotas associated with the constraint structure. Assume without loss of generality that the row sum $\sum_a x_{ia}$ is an integer for each $i \in N$.[29] We say that a preference of agent i is additive if there exist associated values $(v_{ia})_{a\in O}$ such that, for any pure assignment $\bar{\mathbf{x}}_i$ for i, her utility for $\bar{\mathbf{x}}_i$ is given by $\sum_{a\in O} \bar{x}_{ia} v_{ia}$.

Theorem 9. (Utility Guarantee) *Suppose that each agent i has additive preferences with associated values $(v_{ia})_{a\in O}$. Then, any expected assignment X is implementable by a lottery such that, for each i,*

(1) *for any pair $\bar{X}$ and $\bar{X}'$ of pure assignments used in the lottery, the difference between i's utility under $\bar{X}$ and the one under $\bar{X}'$ is at most Δ_i,*
(2) *for any pure assignment $\bar{X}$ used in the lottery, the difference between i's utility under $\bar{X}$ and her expected utility under (any lottery implementing) X is at most Δ_i,*

where $\Delta_i := \max\{v_{ia} - v_{ib} | a, b \in O, x_{ia}, x_{ib} \notin \mathbb{Z}\}$ is the utility difference between i's most valuable and least valuable fractionally assigned objects at X.

This theorem establishes that, given an expected assignment, there exists a lottery over pure assignments implementing it with small utility variation. More specifically, the first property (1) implies that the utility difference between any two pure assignments used in the lottery is at most the utility difference between the agent's most valuable and least valuable (fractionally assigned) objects.[30] The second property (2) is an immediate corollary of the first one, providing a bound on the difference between the expected utility of the given expected assignment and the utility of any pure assignment used in the lottery.

A proof sketch of Theorem 9 can be given based on Theorem 1. The idea is to supplement the actual constraints of the problem with a set of artificial "utility proximity" constraints as follows. For each agent i and integer k, the k^{th} constraint set of agent i, S_{ik}, consists of his $1^{\text{st}}, 2^{\text{nd}}, \dots, k^{\text{th}}$ most preferred objects; its floor and ceiling constraints are $\lfloor \sum_{a\in S_{ik}} x_{ia} \rfloor$

[29] This assumption is without loss of generality because any expected assignment with non-integral row sums is equivalent to an expected assignment with an additional column representing a null object, the sole purpose of which is to ensure that rows sum to integer amounts.

[30] In the example at the beginning of this section where $2n$ objects are assigned to two agents, for instance, lotteries with large utility variations – such as a lottery under which an agent sometimes gets her best n objects and sometimes gets her n worst objects – can be avoided.

and $\lceil\sum_{a\in S_{ik}} x_{ia}\rceil$, respectively. The resulting constraint structure is still a bihierarchy after this addition, so Theorem 1 guarantees that the expected assignment can be implemented with all of the constraints satisfied. Satisfying the artificial "utility proximity" constraints means that in each realized assignment, each agent receives her k most preferred objects, for each k, with approximately the same probability as in the original expected assignment, thus resulting in a small utility variation. To illustrate, recall the example in the Introduction where two agents are assigned to two of four objects which they rank the same way. The supplementary constraints are depicted in Figure 3, and entail the requirement: *each agent must get one of a and b, at most two out of a, b, and c, and two objects in total.*[31]

Figure 3 about here

The utility difference between the best and worst implementation is then no more than the difference between one's values of a (most preferred) and d (least preferred), as stated by Theorem 9.

Theorem 9 can be used to augment ex-post fairness in conjunction with some multi-unit random assignment algorithm. We say that an expected assignment X satisfies interim equal treatment of equals if, for any pair of agents i and j whose utility functions are identical, i and j are indifferent between $\mathbf{x}_i$ and $\mathbf{x}_j$. The following claim is an immediate corollary of Theorem 9.

Corollary 4. *Suppose that an expected assignment satisfies interim equal treatment of equals and i and j have an identical additive utility function. Then there exists a lottery implementing the expected assignment such that, for any pure assignment used in the lottery, the difference between i's utility under his pure assignment and her utility under j's pure assignment is at most Δ_i.*

One useful application of this conjunction idea may be to our generalized pseudo-market mechanism developed in Section 4. Any assignment produced by the generalized pseudo-market mechanism is interim envy free (Theorem 8) because all agents have the same budget and face the same prices, implying that it satisfies interim equal treatment of equals. By utilizing Corollary 4 as well, we can also ensure that ex-post envy is bounded.

5.1. **Application: Two-Sided Matching.** With slight modification, the utility guarantee method can also be applied to two-sided matching environments. Let both N and O be sets of agents and consider many-to-many matching in which each agent in N can be matched

[31] In the $2n$ object example at the beginning of this section, for instance, the above artificial constraints require that an agent receive either zero or one unit of her top object, exactly one from her top two objects, either one or two from her top three, exactly two from her top four, and so on.

with multiple agents in O, and vice versa. Similarly to agents in N, we say that a preference of agent $a \in O$ is additive if there exist associated values $(w_{ia})_{i \in N}$ such that, for any pure assignment $\bar{\mathbf{x}}_a$ for a, her utility for $\bar{\mathbf{x}}_a$ is given by $\sum_{i \in N} \bar{x}_{ia} w_{ia}$. We focus on a problem where the constraint structure consists of all rows and columns (in addition to all singletons).

Theorem 10 (Two-Sided Utility Guarantee)**.** *Suppose that each agent has additive preferences, with associated values $(v_{ia})_{a \in O}$ for each $i \in N$ and $(w_{ia})_{i \in N}$ for each $a \in O$. Then, any expected assignment X is implementable by a lottery such that,*

(1) *for any agent $i \in N$,*
   (a) *for any pair $\bar{X}$ and $\bar{X}'$ of pure assignments used in the lottery, the difference between i's utility under $\bar{X}$ and the one under $\bar{X}'$ is at most Δ_i,*
   (b) *for any pure assignment $\bar{X}$ used in the lottery, the difference between i's utility under $\bar{X}$ and her expected utility under (any lottery implementing) X is at most Δ_i,*

   where $\Delta_i = \max\{v_{ia} - v_{ib} | a, b \in O, x_{ia}, x_{ib} \notin \mathbb{N}\}$, and
(2) *for any agent $a \in O$,*
   (a) *for any pair $\bar{X}$ and $\bar{X}'$ of pure assignments used in the lottery, the difference between a's utility under $\bar{X}$ and the one under $\bar{X}'$ is at most Δ_a,*
   (b) *for any pure assignment $\bar{X}$ used in the lottery, the difference between a's utility under $\bar{X}$ and her expected utility under (any lottery implementing) X is at most Δ_a,*

   where $\Delta_a = \max\{w_{ia} - w_{ja} | i, j \in N, x_{ia}, x_{ja} \notin \mathbb{N}\}$.

Proof. The proof is a straightforward adaptation of the proof of Theorem 9 and hence is omitted. □

As a possible application, consider two leagues of sports teams N and O, say the National League (NL) and the American League (AL) in professional baseball, and the planner who wants to schedule interleague play. For concreteness, suppose there are four teams in each league, and each team must play 6 games against teams on the other league.

The planner wants to ensure that the strength of opponents that teams in a league play against is as equalized as possible among teams in the same league. For that goal, the panner could first order teams in each league by some measure of their strength (e.g., win/loss ratio from the prior season), and give a uniform probability for each match, which requires each team to play every team of the other league 1.5 times: That will give one specific expected assignment in which each pair of teams in the same league is treated equally. Theorem 10 can then be used to find a pure assignment, in which differences in schedule strength are bounded by the difference between one game with the strongest opponent and one with the weakest opponent in the other league. The idea is to add artificial constraints, one for each upper contour set for each team, but on both sides, as depicted in Figure 4.

Figure 4 about here

Implementing with the appropriate quota constraints produces a feasible (i.e., integer) matchup schedule which is approximately fair. An example outcome is depicted in Figure 5.

Figure 5 about here

We note that transforming this feasible match into a specific schedule — i.e., not only how often does Team A play Team B, but *when* — is considerably more complicated. For example, the problem involves scheduling both intraleague and interleague matches simultaneously, dealing with geographical constraints and so forth. See Nemhauser and Trick (1998) for further discussion of sport scheduling.

6. Beyond Two-Sided Assignment

Throughout the paper we have focused on an environment in which the participants are divided into two sides such as agents and objects. However, some of our results can be extended beyond two-sided assignment, as described below.

Let Ω be a finite set. An expected assignment is a profile $X = [x_\omega]_{\omega \in \Omega}$ where $x_\omega \in (-\infty, \infty)$ for all $\omega \in \Omega$. A pure assignment is an expected assignment each of whose entries is an integer. A constraint structure $\mathcal{H}$ is a collection of subsets of Ω. The model in the previous sections corresponds to the case in which Ω is $N \times O$, the set of all agent-object pairs. The notions of implementability and universal implementability are defined analogously, just as the notion of bihierarchy for constraint structures. In this setting, the previous sufficient condition for universal implementability holds without modification.

Theorem 11. *A constraint structure is universally implementable if it forms a bihierachy.*

The proof of Theorem 11 is identical to the one for Theorem 1 and thus is omitted. Note that the proof of the latter did not rely on any two-sided structure.

We note that Lemma 1 clearly holds in this general environment (with an identical proof), providing a necessary condition for universal implementability. We can apply this lemma to show the difficulty one faces in implementing expected assignments in multi-sided assignment and roommate matching.

6.1. **Multi-Sided Assignment.** Thus far, we have focused on two-sided assignment environments in which agents on one side are assigned to objects (or agents) on the other side. As noted before, many important market design problems fall into the two-sided assignment environment. Sometimes, however, matching involves more than two sides. For instance, students may be assigned to different schools and after-school programs, in which case the matching must be three-sided, consisting of student/school/after-school triples. Or, manufacturers may need to match with multiple suppliers, ensuring mutual compatibility of products or the right combination of capabilities.

Our main point is most easily made by starting with a three-sided matching problem in which we introduce another finite set L of (say) agents, in addition to N and O. A matching then consists of a triple $(i, a, l) \in N \times O \times L$, and an expected assignment is defined by a profile $X = [x_{(i,a,l)}]_{(i,a,l)\in N\times O\times L}$ that assigns a real number to each triple (i, a, l). Constraints on the expected assignment can be described as before via the constraint structure, i.e., the sets of (i, a, l)'s whose entries are subject to ceiling or floor quota constraints. That is, the constraint structure $\mathcal{H} \subset 2^{N\times O\times L}$ is a collection of subsets of $N \times O \times L$. As in the classical setup, the basic constraints arise from the fact that each agent in N, each object in O, and each agent in L are assigned to some pair in the other two sides (which may include a null object or a null agent). Hence, it is natural to assume that $\mathcal{H}$ contains the sets $\{\{i\} \times O \times L | i \in N\}$, $\{N \times \{a\} \times L | a \in O\}$, and $\{N \times O \times \{l\} | l \in L\}$. We call such a constraint structure a **canonical three-sided constraint structure**.

Notice that the problem reduces to that of the two-sided assignment if N or O or L is a singleton set. It turns out that, except for such cases, no analogue of the Birkhoff-von Neumann theorem holds in three-sided matching.

Theorem 12. (Impossibility with Three-Sided Matching) *No canonical three-sided constraint structure with $|N|, |O|, |L| \geq 2$ is universally implementable.*

Proof. We prove the result by showing that any canonical three-sided constraint structure $\mathcal{H}$ with $|N|, |O|, |L| \geq 2$ contains an odd cycle. By Lemma 1, this is sufficient for the failure of universal implementability (as pointed out before, even though the proof of Lemma 1 formally deals with the two-sided matching setup, its proof does not depend on it).

Fix $i \in N, a \in O, l \in L$ and consider three sets $S_i := \{i\} \times O \times L, S_a := N \times \{a\} \times L$, and $S_l := N \times O \times \{l\}$. Fix $i' \in N, a' \in O, l' \in L$ such that $i' \neq i$, $a' \neq a$, and $l' \neq l$ (such i', a', and l' exist since $|N|, |O|, |L| \geq 2$). Then $(i, a, l') \in S_i \cap S_a \setminus S_l$, $(i, a', l) \in S_i \cap S_l \setminus S_a$, and $(i', a, l) \in S_a \cap S_l \setminus S_i$. We thus conclude that S_i, S_a, and S_l form an odd cycle. □

It is clear from the proof that the same impossibility result holds for any multi-sided matching of more than two kinds of agents.

Remark 1. (Matching with Contracts) *Firms sometimes hire workers for different positions with different terms of contract. For instance, hospitals hire medical residents for different kinds of positions (such as research and clinical positions), and different positions may entail different duties and compensations. To encompass such situations, Hatfield and Milgrom (2005) develop a model of "matching with contracts," in which a matching specifies not only which firm employs a given worker but also on what contract terms. At first glance, introducing contract terms may appear to transform the environment into a canonical three-sided matching setting. This is in fact not the case. If we let* L *denote the set of possible contract terms, there is no sense in which the constraint structure contains sets of the form* $N \times O \times \{l\}$*. In words, there is no reason that each contract term should be chosen by some worker-firm pair. Rather, matching with contracts can be subsumed into our two-sided assignment setup by redefining the object set as* $O' := O \times L$*.*

6.2. **Roommate Matching.** The "roommate problem" describes another interesting matching problem, in which any agent can, in principle, be matched to any other. One example is "pairwise kidney exchange" (Roth, Sonmez and Ünver, 2005), in which a kidney patient with a willing-but-incompatible donor is to be matched to another patient-donor pair. If two such pairs are successfully matched, then the donor in each pair donates her kidney to the patient of the other pair.

For our analysis, the important elements of a roommate matching problem include a (finite) set of agents, N, and a set $\Omega := \{\{i, j\}|i, j \in N\}$ of possible (unordered) pairs of agents who can be matched as roommates. If the pair $\{i, i\}$ is formed, that means that i is unmatched. An expected assignment is a profile $X = [x_\omega]_{\omega \in \Omega}$ where $x_\omega \in [0, 1]$ for all $\omega \in \Omega$ and a constraint structure $\mathcal{H}$ is a collection of subsets of Ω. We assume that each i must be assigned to some agent (possibly himself), so $\mathcal{H}$ must contain set $S_i := \{\{i, j\}|j \in N\}$ for each $i \in N$. We call a constraint $\mathcal{H}$ satisfying this property a **canonical roommate matching constraint structure**.

Notice that the problem reduces to that of two-sided matching if there are two or fewer agents, implying that a canonical roommate matching constraint structure in such problems is universally implementable. The next result shows that these are the only cases for which universal implementability holds.

Theorem 13. (Impossibility with Roommate Matching) *No canonical roommate matching constraint structure with at least three agents is universally implementable.*

Proof. We prove the result by showing that any canonical roommate matching constraint structure contains an odd cycle if there are at least three agents. Consider $i, j, k \in N$, who are all distinct (such agents exist since $|N| \geq 3$). Then, $\{i, j\} \in (S_i \cap S_j) \setminus S_k$, $\{j, k\} \in (S_j \cap S_k) \setminus S_i$, and $\{i, k\} \in (S_i \cap S_k) \setminus S_j$. We thus conclude that S_i, S_j, and S_k form an odd cycle. □

7. Conclusion

This paper extends the applicability of random assignment methods to an expanded class of problems, including problems with certain auxillary constraints and many-to-many matching. We apply our results to extend two prominent mechanisms — Bogomolnaia and Moulin's (2001) probabilistic serial mechanism and Hylland and Zeckhauser's (1979) pseudo-market mechanism — to accommodate features such as endogenous capacities, group-specific quotas, multi-unit demand, and scheduling constraints. We also developed a "utility guarantee" method which can be used to supplement the ex ante fairness promoted by randomization, by limiting the extent of ex post unfairness.

Methodologically, we identified a maximal generalization of the Birkhoff-von Neumann theorem. Specifically, we demonstrated that the bihierarchy condition is both necessary and sufficient for a constraint structure to be universally implementable in canonical two-sided environments. We found that there is no similar universally implementability property for matching with three sides or more, nor for roommate problems.

The central goal of research in market design is to facilitate applications, and we are most hopeful that the tools and mechanisms described here herald still further applications to come.

References

Abdulkadiroğlu, Atila. 2005. "College Admission with Affirmative Action." *International Journal of Game Theory*, 33: 535–549.

Abdulkadiroğlu, Atila, and Tayfun Sönmez. 2003*a*. "Ordinal Efficiency and Dominated Sets of Assignments." *Journal of Economic Theory*, 112: 157–172.

Abdulkadiroğlu, Atila, and Tayfun Sönmez. 2003*b*. "School Choice: A Mechanism Design Approach." *American Economic Review*, 93: 729–747.

Abdulkadiroğlu, Atila, Parag A. Pathak, and Alvin E. Roth. 2005. "The New York City High School Match." *American Economic Review Papers and Proceedings*, 95: 364–367.

Birkhoff, Garrett. 1946. "Three Observations on Linear Algebra." *Revi. Univ. Nac. Tucuman, ser A*, 5: 147–151.

Bogomolnaia, Anna, and Herve Moulin. 2001. "A New Solution to the Random Assignment Problem." *Journal of Economic Theory*, 100: 295–328.

Bogomolnaia, Anna, and Herve Moulin. 2002. "A Simple Random Assignment Problem with a Unique Solution." *Economic Theory*, 19: 623–635.

Budish, Eric. 2009. "The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes." Harvard University, Unpublished mimeo.

Budish, Eric, and Estelle Cantillon. 2009. "The Multi-Unit Assignment Problem: Theory and Evidence from Course Allocation at Harvard." Harvard University, Unpublished mimeo.

Che, Yeon-Koo, and Fuhito Kojima. 2008. "Asymptotic Equivalence of the Random Priority and Probabilistic Serial Mechanisms." forthcoming, *Econometrica.*

Che, Yeon-Koo, and Ian Gale. 2008. "Market versus Non-Market Assignment of Ownership." Unpublished mimeo.

Crès, Herve, and Herve Moulin. 2001. "Scheduling with Opting Out: Improving upon Random Priority." *Operations Research*, 49: 565–577.

Edmonds, J. 1970. "Submodular functions, matroids, and certain polyhedra." *Combinatorial Structures and Their Applications, R. Guy, H. Hanani, N. Sauer, and J. Schonheim, eds, Gordon and Breach, New York*, 69–87.

Ehlers, Lars, and Betina Klaus. 2003. "Coalitional strategy-proof and resource-monotonic solutions for multiple assignment problems." *Social Choice and Welfare*, 21: 265–280.

Ghouila-Houri, A. 1962. "Caractérisation des matrices totalment unimodulaires." *Comptes Rendus Hebdomadaires des Séances de l' Académie des Sciences (Paris)*, 254: 1192–1194.

Hatfield, John William. 2008. "Strategy-Proof, Efficient, and Nonbossy Quota Allocations." forthcoming, *Social Choice and Welfare.*

Hatfield, John William, and Paul Milgrom. 2005. "Matching with Contracts." *American Economic Review*, 95: 913–935.

Hoffman, AJ, and JB Kruskal. 1956. "Integral boundary points of convex polyhedra." *in "Linear Inequalities and Related Systems" (H. Kuhn and A. Tucker, Eds.) Annals of Mathematics Studies*, 38: 223–246.

Hylland, Aanund, and Richard Zeckhauser. 1979. "The Efficient Allocation of Individuals to Positions." *Journal of Political Economy*, 87: 293–314.

Katta, A-K., and J. Sethuraman. 2006. "A Solution to The Random Assignment Problem on The Full Preference Domain." forthcoming, *Journal of Economic Theory.*

Kesten, Onur. 2007. "Why do popular mechanisms lack efficiency in random environments?" *Journal of Economic Theory*, forthcoming.

Kojima, Fuhito. 2008. "Random Assignment of Multiple Indivisible Objects." forthcoming, *Mathematical Social Sciences.*

Kojima, Fuhito, and Mihai Manea. 2008. "Incentives in the Probabilistic Serial Mechanism." forthcoming, *Journal of Economic Theory.*

Manea, Mihai. 2006. "A Constructive Proof of The Ordinal Efficiency Welfare Theorem." forthcoming, *Journal of Economic Theory.*

Manea, Mihai. 2009. "Asymptotic Ordinal Inefficiency of Random Serial Dictatorship." forthcoming, *Theoretical Economics.*

McLennan, Andrew. 2002. "Ordinal Efficiency and The Polyhedral Separating Hyperplane Theorem." *Journal of Economic Theory*, 105: 435–449.

Milgrom, P. 2009. "Assignment messages and exchanges." *American Economic Journal: Microeconomics*, 1(2): 95–113.

Nemhauser, George, and Michael Trick. 1998. "Scheduling a Major College Basketball Conference." *Operations Research*, 46: 1–8.

Papai, Szilazi. 2001. "Strategyproof and nonbossy multiple assignments." *Journal of Public Economic Theory*, 3: 257–271.

Roth, Alvin E. 2007. "Repugnance as a Constraint on Markets." *Journal of Economic Perspectives*, 21: 37–58.

Roth, Alvin E., Tayfun Sonmez, and Utku Ünver. 2005. "Pairwise Kidney Exchange." *Journal of Economic Theory*, 125: 151–188.

Sönmez, Tayfun, and Uktu Ünver. 2008. "Course Bidding at Business Schools." forthcoming, *International Economic Review.*

von Neumann, John. 1953. "A certain zero-sum two-person game equivalent to the optimal assignment problem." In *Contributions to the theory of games, Vol. 2.* , ed. H. W. Kuhn and A. W. Tucker. Princeton, New Jersey:Princeton University Press.

Yilmaz, Ozgur. 2009. "Random assignment under weak preferences." *Games and Economic Behavior*, 66: 546–558.

Appendix A. Definition of the generalized probabilistic serial mechanism

Formally, the generalized probabilistic serial mechanism is defined through the following **symmetric simultaneous eating algorithm**, or the eating algorithm for short.

Generalized Probabilistic Serial Mechanism. For any $(i, a) \in S \subseteq N \times O$, let

$$\chi(i, a, S) = \begin{cases} 1 & \text{if } (i, a) \in S \text{ and } a \succeq_i b \text{ for any } b \text{ with } (i, b) \in S, \\ 0 & \text{otherwise,} \end{cases}$$

be the indicator function that a is the most preferred object for i among objects b such that (i, b) is listed in S.

Given a preference profile $\succ$, the eating algorithm is defined by the following sequence of steps. Let $S^0 = N \times O, t^0 = 0$, and $x^0_{ia} = 0$ for every $i \in N$ and $a \in O$. Given $S^0, t^0, X^0 = [x^0_{ia}]_{i\in N, a\in O}, \ldots, S^{v-1}, t^{v-1}, X^{v-1} = [x^{v-1}_{ia}]_{i\in N, a\in O}$, for any $(i, a) \in S^{v-1}$ define

$$\text{(A.1)} \qquad t^v(i, a) = \min_{S\in\mathcal{H}_2:(i,a)\in S} \sup \left\{ t \in [0, 1] | \sum_{(j,b)\in S} [x^{v-1}_{jb} + \chi(j, b, S^{v-1})(t - t^{v-1})] < \overline{q}_S \right\},$$

$$\text{(A.2)} \qquad t^v = \min_{(i,a)\in S^{v-1}} t^v(i, a),$$

$$\text{(A.3)} \qquad S^v = S^{v-1} \setminus \{(i, a) \in S^{v-1} | t^v(i, a) = t^v\},$$

$$\text{(A.4)} \qquad x^v_{ia} = x^{v-1}_{ia} + \chi(i, a, S^{v-1})(t^v - t^{v-1}).$$

Since $N \times O$ is a finite set, there exists $\overline{v}$ such that $t^{\overline{v}} = 1$. We define $PS(\succ) := X^{\overline{v}}$ to be the generalized probabilistic serial expected assignment for the preference profile $\succ$.

Appendix B. Extended Framework for the Pseudo-Market Mechanism.

Here, we discuss how the basic framework can be extended to accommodate more general preferences, following Milgrom's (2009) class of integer assignment messages. As before, each agent $i \in N$ submits cardinal values and a set of constraints. The agent is given a set of rows, corresponding to different roles that objects can play (the set of roles is denoted R_i). We first describe the basic requirement of the constraints. First, the agent may submit a hierarchical set $\mathcal{H}_{ir}$ of constraints for each "role" r, with each $\mathcal{H}_{ir}$ containing the "row" constraint for row (i, r) ("single-role constraints"). Second, the agent submits a hierarchical set $\mathcal{H}_{i0}$ of constraints pertaining to i's total quantity across multiple roles; each of these sets contains multiple rows ("multi-role constraints"). Last, for each object $a \in O$ the agent submits a hierarchical set of constraints pertaining to i's consumption of object a across his multiple rows; each of these sets corresponds to a subset of the column for object a ("object-specific constraints"). In course allocation, for instance, these object-specific constraints ensure that each student gets at most one seat in any course, even if a course appears in multiple rows. Let $\mathcal{H}_i$ denote the union of all of i's constraints.

As described in the main text, each agent's valuations and constraints together define her utility on an extended space of consumption bundles used in particular roles. This formulation induces the agent's utility function over consumption bundles in a natural manner, as follows. Let integer-valued vector $\overline{\mathbf{x}}_i = (\overline{x}_{ia})_{a \in O}$ denote a consumption bundle for agent i. The utility for i from consumption bundle $\overline{\mathbf{x}}_i$ is the solution to the following linear program

$$\begin{aligned} u_i(\overline{\mathbf{x}}_i) = \max \sum_{r \in R_i} \sum_{a \in O} v_{ira} \overline{x}_{ira} \ \ &\text{subject to} \\ (\sum_{r \in R_i} \overline{x}_{ira})_{a \in O} = \overline{\mathbf{x}}_i \ &\text{(adding up constraint)} \\ 0 \leq \sum_{\{((i,r),a)\} \in S_i} \overline{x}_{ira} \leq \overline{q}_{S_i} \ &\text{for all } S_i \in \mathcal{H}_i \text{ (agent constraints)} \\ \overline{x}_{ira} \in \mathbb{Z} \ &\text{for all } r, a. \end{aligned}$$

As with the initial model, this utility function can be extended to a fractional assignment $\mathbf{x}_i \in \mathbb{R}^{|O|}$, so we can write $u_i(\mathbf{x}_i)$ for the agent's utility from fractional bundle $\mathbf{x}_i$.

Given the assignment messages and utility functions defined this way, it can be readily seen that the demand correspondence resulting from such a utility function is nonempty, convex-valued and upper-hemicontinuous. Given these properties, the generalized pseudo market mechanism can be constructed precisely as in the baseline case in the main text, and all subsequent results follow without modifications of proofs.

Web Appendix: not intended for publication.

Appendix C. Proofs of Theorems 1 and 11

Since Theorem 1 is a special case of Theorem 11, we prove the latter.

A matrix is **totally unimodular** if the determinant of every square submatrix is 0 or -1 or $+1$. We make use of the following result.

Lemma 2. *(Hoffman and Kruskal (1956)) If a matrix A is totally unimodular, then the vertices of the polyhedron defined by linear integral constraints are integer valued.*

The proof strategy for Theorem 11 proceeds in two steps. First we show that if a constraint structure forms a bihierarchy, then the incidence matrix of the constraint structure is totally unimodular. Second we apply Lemma 2 to show that the constraint structure is universally implementable.

After an earlier draft was circulated, we were informed that Edmonds (1970) has previously shown that the incidence matrix of a bihierarchical constraint structure is totally unimodular. We include our proof for completeness below. We utilize the following result for our proof.

Lemma 3. *(Ghouila-Houri (1962)) A $\{0,1\}$ incidence matrix is totally unimodular if and only if each subcollection of its columns can be partitioned into red and blue columns such that for every row of that collection, the sum of entries in the red columns differs by at most one from the sum of the entries in the blue columns.*

Proof of Theorem 11. Suppose first $\mathcal{H}$ forms a bihierarchy, with $\mathcal{H}_1$ and $\mathcal{H}_2$ such that $\mathcal{H}_1 \cup \mathcal{H}_2 = \mathcal{H}$, $\mathcal{H}_1 \cap \mathcal{H}_2 = \emptyset$ and both $\mathcal{H}_1$ and $\mathcal{H}_2$ are hierarchies. Let A be the associated incidence matrix. Take any collection of columns of A, corresponding to a subcollection E of $\mathcal{H}$. We shall partition E into two sets, B and R. First, for each $i = 1, 2$, we partition $E \cap \mathcal{H}_i$ into nonempty sets $E_i^1, E_i^2, \dots, E_i^{k_i}$ defined recursively as follows: Set $E_i^0 \equiv \emptyset$ and, for each $j = 1, \dots,$ we let

$$E_i^j := \{S \in (E \cap \mathcal{H}_i) \setminus (\bigcup_{j'=1}^{j-1} E_i^{j'}) \,|\, \nexists S' \in (E \cap \mathcal{H}_i) \setminus (\bigcup_{j'=1}^{j-1} E_i^{j'} \cup \{S\}) \text{ such that } S' \supset S\}.$$

(The non-emptiness requirment means that once all sets in $E \cap \mathcal{H}_i$ are accounted for, the recursive definition stops, which it does at a finite $j = k_i$.) Since $\mathcal{H}_i$ is a hierarchy, any two sets in E_i^j must be disjoint, for each $j = 1, \dots, k_i$. Hence, any element of Ω can belong to at most one set in each E_i^j. Observe next for $j < l$, $\bigcup_{S \in E_i^l} S \subset \bigcup_{S \in E_i^j} S$. In other words, if an element of Ω belongs to a set in E_i^l, it must also belong to a set in E_i^j for each $j < l$.

We now define sets B and R that partition E:

$$B := \{S \in E | S \in E_i^j, i+j \text{ is an even number }\},$$

and

$$R := \{S \in E | S \in E_i^j, i+j \text{ is an odd number }\}.$$

We call the elements of B "blue" sets, and call the elements of R "red" sets.

Fix any $\omega \in \Omega$. If ω belongs to any set in $E \cap \mathcal{H}_1$, then it must belong to exactly one set $S_1^j \in E_1^j$, for each $j = 1, \dots, l$ for some $l \leq k_1$. These sets alternate in colors in $j = 1, 2, \dots,$ starting with blue: S_1^1 is blue, S_1^2 is red, S_1^3 is blue, and so forth. Hence, the number of blue sets in $E \cap \mathcal{H}_1$ containing ω either equals or exceeds by one the number of red sets in $E \cap \mathcal{H}_1$ containing ω. By the same reasoning, if ω belongs to any set in $E \cap \mathcal{H}_2$, then it must belong to one set $S_2^j \in E_2^j$, for each $j = 1, \dots, m$ for some $m \leq k_2$. These sets alternate in colors in $j = 1, 2, \dots,$ starting with red: S_2^1 is red, S_2^2 is blue, S_2^3 is red, and so forth. Hence, the number of blue sets in $E \cap \mathcal{H}_2$ containing ω is less by one than or equal to the number of red sets in $E \cap \mathcal{H}_2$ containing ω. In sum, the number of blue sets in E containing ω differs at most by one from the number of red sets in E containing ω. Thus A is totally unimodular by Lemma 3.

Choose an arbitrary expected assignment X and consider the set

$$\{X' | \lfloor x_S \rfloor \leq x'_S \leq \lceil x_S \rceil, \forall S \in \mathcal{H}\}. \tag{C.1}$$

By Lemma 2, every vertex of the set (C.1) is integer valued. Since (C.1) is a convex polyhedron, any point of it (including X) can be written as a convex combination of its vertices. Since we chose X arbitrarily, the constraint structure $\mathcal{H}$ is universally implementable. □

Appendix D. Algorithm for Implementing Expected Assignments

This section provides a computable algorithm, which also serves as a constructive proof for Theorem 11 (and hence Theorem 1). For ease of understanding, we first illustrate the algorithm, using an example. We then formally define the algorithm.

Consider $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}$ and $\mathcal{H} = \{\{\omega_1\}, \{\omega_2\}, \{\omega_3\}, \{\omega_4\}, S_1, S_2\}$, where $S_1 := \{\omega_2, \omega_3\}$ and $S_2 := \{\omega_3, \omega_4\}$. Observe that $\mathcal{H}$ is a bihierarchy consisting of two hierarchies, $\mathcal{H}_1 = \{\{\omega_1\}, \{\omega_2\}, \{\omega_3\}, \{\omega_4\}, S_1\}$ and $\mathcal{H}_2 = \{S_2\}$. Suppose we wish to implement an expected assignment X with $x_{\{\omega_1\}} = 0.3, x_{\{\omega_2\}} = 0.7, x_{\{\omega_3\}} = 0.3$ and $x_{\{\omega_4\}} = 0.7$. We represent the given expected assignment X as a network flow. The particular way in which the flow network is constructed is crucial for the algorithm, and we first illustrate the construction informally based on the example (depicted in Figure A1).

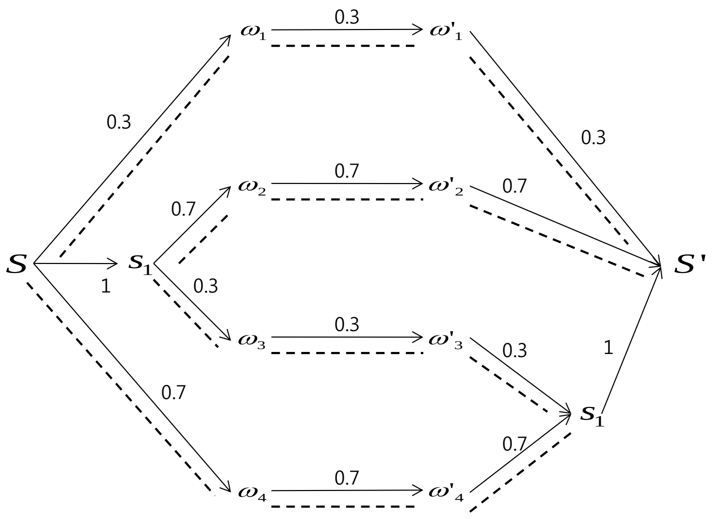


Figure A1: A network flow representation of the example X.

Intuitively, we view the total assignment as flows that travel from source s to sink s' of a network (s and s' can be interpreted as corresponding to the entire set Ω). First, the flows travel through the sets in one hierarchy $\mathcal{H}_1$, arranged in "descending" order of set-inclusion; the flows move from bigger to smaller sets along the directed edges representing the set-inclusion tree, reaching at last the singleton sets. This accounts for the left side of the flow network in Figure 1, where the numbers on the edges depict the flows. From then on, the flows travel through the sets in the other hierarchy $\mathcal{H}_1$ which is augmented, without loss, to include the singleton sets and the entire set Ω, with primes attached for notational clarity. These sets are now arranged in "ascending" order of set-inclusion; the flows travel from smaller to bigger sets along the directed edges representing the reverse set-inclusion tree, reaching at the end the total set s', or the sink.

Notice that the flow associated with each edge reflects the expected assignment for the corresponding set. For instance, the flow from ω_2 to ω_2' is the expected assignment $x_{\{\omega_2\}} = 0.7$ for set ω_2, and likewise the flow from ω_3 to ω_3' is $x_{\{\omega_3\}} = 0.3$. The flow from s to S_1 represents the expected assignment $\omega_{S_1} = 1$ for set S_1. Naturally, the latter flow must be the sum of the two former flows. More generally, the additive structure of the expected assignment is translated into the "law of conservation": *the flow reaching each vertex except for s and s' must equal the flow leaving that vertex.*

Given the flow network, the algorithm identifies a cycle of agent-object pairs with fractional assignments. Starting with any edge with fractional flow, say (ω_2, ω_2'), we find another edge with a fractional flow that is adjacent to ω_2'. Such an edge, (ω_2', s'), exists due to the law of conservation: if all neighboring flows were integer we would have a contradiction. We keep adding new edges with fractional flows in this fashion, the ability to do so ensured by the law of conservation, until we create a cycle. In this case, the cycle of vertices is $\omega_2 - \omega_2' - s' - \omega_1' - \omega_1 - s - \omega_4 - \omega_4' - S_2 - \omega_3' - \omega_3 - S_1 - \omega_2$. This cycle is denoted by the dotted lines in Figure 1.

We next modify the flows of the edges in the cycle. First, we raise the flow of each forward edge and reduce the flow of each backward edge at the same rate until at least one flow reaches an integer value. In our example, the flows along all the forward edges rise from 0.7 to 1 and the flows along all the backward edges fall from 0.3 to 0. Importantly, this process preserves the law of conservation, meaning that the operation maintains the feasibility of the new expected assignment. The resulting network flow then gives rise to an expected assignment X' where $x'_{\{\omega_1\}} = 0, x'_{\{\omega_2\}} = 1, x'_{\{\omega_3\}} = 0$, and $x'_{\{\omega_4\}} = 1$. Next, we readjust the flows of the edges in the cycle in the reverse direction, raising those with backward edges and reducing those with forward edges in an analogous manner, which gives rises to another expected assignment X'' where $x''_{\{\omega_1\}} = 1, x''_{\{\omega_2\}} = 0, x''_{\{\omega_3\}} = 1$, and $x''_{\{\omega_4\}} = 0$. We can now decompose P into these two matrices, i.e., $X = 0.7X' + 0.3X''$.

The random algorithm then selects X' with probability 0.7 and X'' with probability 0.3. Since in this particular example both X' and X'' are integer valued, there is no need to reiterate the decomposition process. In general, each step in the algorithm reduces the number of fractional flows in the network, converting at least one to an integer. The total number of steps in the random algorithm is therefore limited to the number of fractional flows. Also, each step visits each remaining fractional flow at most once, so the total number of visits grows at most as the square of the number of fractional flows. Thus, the run time of the algorithm is polynomial in $|\mathcal{H}|$.

We now define the algorithm formally. Let $\mathcal{H}$ be a constraint structure associated with a set Ω and assume that $\mathcal{H}$ is a bihierarchy, where $\mathcal{H}_1$ and $\mathcal{H}_2$ are hierarchies such that $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$. Let $X = [x_\omega]$ be an expected assignment whose entries sum up to an integer (the generalization to the case with a fractional sum is straightforward). We construct a flow network as follows. The set of vertices is composed of the source s and the sink s', two vertices v_ω and $v_{\omega'}$ for each element $\omega \in \Omega$, and v_S for each $S \in \mathcal{H} \setminus [(\bigcup_{\omega\in\Omega}\{\omega\}) \cup (N \times O)]$. We place (directed) edges according to the following rule.[32]

(1) For each $\omega \in \Omega$, an edge $e = (v_\omega, v_{\omega'})$ is placed from v_ω to $v_{\omega'}$.

[32] An edge is defined as an ordered pair of verticies. All edges in this paper are directed, so we omit the adjective "directed."

(2) An edge $e = (v_S, v_{S'})$ is placed from S to $S' \neq S$ where $S, S' \in \mathcal{H}_1$, if $S' \subset S$ and there is no $S'' \in \mathcal{H}_1$ where $S' \subset S'' \subset S$.[33]
(3) An edge $e = (v_S, v_{S'})$ is placed from S to $S' \neq S$ where $S, S' \in \mathcal{H}_2$, if $S \subset S'$ and there is no $S'' \in \mathcal{H}_2$ where $S \subset S'' \subset S'$.
(4) An edge $e = (s, v_S)$ is placed from the source s to v_S if $S \in \mathcal{H}_1$ and there is no $S' \in \mathcal{H}_1$ where $S \subset S'$.
(5) An edge $e = (v_S, s')$ is placed from v_S to the sink s' if $S \in \mathcal{H}_2$ and there is no $S' \in \mathcal{H}_2$ where $S \subset S'$.

We associate flow with each edge as follows. For each $e = (v_\omega, v_{\omega'})$, we associate flow $x_e = x_\omega$. For each e that is not of the form $(v_\omega, v_{\omega'})$ for some $\omega \in \Omega$, the flow x_e is (uniquely) set to satisfy the flow conservation, that is, for each vertex v different from s and s', the sum of flows into v is equal to the sum of flows from v. Observe that the construction of the network (specifically items (2)-(5) above) utilizes the fact that $\mathcal{H}$ is a bihierarchy.

We define the **degree of integrality** of X with respect to $\mathcal{H}$:

$$\deg[X(\mathcal{H})] := \#\{S \in \mathcal{H} | x_S \in \mathbb{Z}\}.$$

Lemma 4. *(Decomposition) Suppose a constraint structure $\mathcal{H}$ forms a bihierarchy. Then, for any X such that deg$[X(\mathcal{H})] < |\mathcal{H}|$, there exist X^1 and X^2 and $\gamma \in (0,1)$ such that*

(i) $X = \gamma X^1 + (1-\gamma) X^2$:

(ii) $x^1_S, x^2_S \in [\lfloor x_S \rfloor, \lceil x_S \rceil]$, $\forall S \in \mathcal{H}$.

(iii) deg$[X^i(\mathcal{H})] >$ deg$[X(\mathcal{H})]$ for $i = 1, 2$.

The following algorithm gives a constructive proof of Lemma 4 and hence the Theorem. Let X be an expectedassignment on a bihierarchy $\mathcal{H}$ with $\deg[X(\mathcal{H})] < |\mathcal{H}|$.

□ **Decomposition Algorithm**

(1) **Cycle-Finding Procedure**
   (a) **Step 0:** Since $\deg[X(\mathcal{H})] < |\mathcal{H}|$ by assumption, there exists an edge $e_1 = (v_1, v_1')$ such that its associated flow x_{e_1} is fractional. Define an edge $f_1 = (v_1, v_1')$ from v_1 to v_1'.
   (b) **Step t=1,. . . :** Consider the vertex v_t' that is the destination of edge f_t.

[33] For the purpose of placing edges, we regard v_ω as a vertex corresonding to a singleton set $\{\omega\} \in H_1$, and v'_ω as a vertex corresonding to a singleton set $\{\omega\} \in H_2$.

(i) If v'_t is the origin of some edge $f_{t'} \in \{f_1, \ldots, f_{t-1}\}$, then stop.[34] The procedure has formed a cycle $(f_{t'}, f_{t'+1}, \ldots, f_t)$ composed of edges in $\{f_1, \ldots, f_t\}$. Proceed to **Termination - Cycle**.

(ii) Otherwise, since the flow associated with f_t is fractional by construction and the flow conservation holds at v'_t, there exists an edge $e_{t+1} = (u_{t+1}, u'_{t+1}) \neq e_t$ with fractional flow such that v'_t is either its origin or destination. Draw an edge f_{t+1} by $f_{t+1} = e_{t+1}$ if v'_t is the origin of e_{t+1} and $f_{t+1} = (u'_{t+1}, u_{t+1})$ otherwise. Denote $f_{t+1} = (v_{t+1}, v'_{t+1})$.

(2) **Termination - Cycle**

(a) Construct a set of flows associated with edges (x^1_e) which is the same as (x_e), except for flows $(x_{e_\tau})_{t' \leq \tau \leq t}$, that is, flows associated with edges that are involved in the cycle from the last step. For each edge e_τ such that $f_\tau = e_\tau$, set $x^1_{e_\tau} = x_{e_\tau} + \alpha$, and each edge e_τ such that $f_\tau \neq e_\tau$, set $x^1_{e_\tau} = x_{e_\tau} - \alpha$, where $\alpha > 0$ is the largest number such that the induced expected assignment $X^1 = (x^1_\omega)_{\omega \in \Omega}$ still satisfies all constraints in $\mathcal{H}$. By construction, $x^1_S = x_S$ if x_S is an integer, and there is at least one constraint set $S \in \mathcal{H}$ such that x^1_S is an integer while x_S is not. Thus $\deg[X^1(\mathcal{H})] > \deg[X(\mathcal{H})]$.

(b) Construct a set of flows associated with edges (x^2_e) which is the same as (x_e), except for flows $(x_{e_\tau})_{t' \leq \tau \leq t}$, that is, flows associated with edges that are involved in the cycle from the last step. For each edge e_τ such that $f_\tau = e_\tau$, set $x^1_{e_\tau} = x_{e_\tau} - \beta$, and each edge e_τ such that $f_\tau \neq e_\tau$, set $x^1_{e_\tau} = x_{e_\tau} + \beta$, where $\beta > 0$ is the largest number such that the induced expected assignment $X^2 = (x^2_\omega)_{\omega \in \Omega}$ still satisfies all constraints in $\mathcal{H}$. By construction, $x^2_S = x_S$ if x_S is an integer, and there is at least one constraint set $S \in \mathcal{H}$ such that x^2_S is an integer while x_S is not. Thus $\deg[X^2(\mathcal{H})] > \deg[X(\mathcal{H})]$.

(c) Set γ by $\gamma\alpha + (1-\gamma)(-\beta) = 0$, i.e., $\gamma = \frac{\beta}{\alpha+\beta}$.

(d) The decomposition of X into $X = \gamma X^1 + (1-\gamma)X^2$ satisfies the requirements of the Lemma by construction.

Appendix E. Proofs of Lemma 1 and Theorem 2

Proof of Lemma 1. Suppose for contradiction that $\mathcal{H}$ is universally implementable and contains an odd cycle $S_1, \ldots, S_l$, with $\omega_i \in S_i \cap S_{i+1}$, $i = 1, \ldots, l-1$ and $\omega_l \in S_l \cap S_1$. Consider

[34] Since there are a finite number of vertices, this procedure terminates in a finite number of steps.

an expected assignment X specified by

$$x_\omega = \begin{cases} \frac{1}{2} & \text{if } \omega \in \{\omega_1, \dots, \omega_l\}, \\ 0 & \text{otherwise,} \end{cases}$$

where x_ω is the entry corresponding to $\omega \in N \times O$. By definition of an odd cycle, $x_{S_i} = 1$ for all $i \in \{1, \dots, k\}$. Since $\mathcal{H}$ is universally implementable, there exist $X^1, X^2, \dots, X^K$ and $\lambda^1, \lambda^2, \dots, \lambda^K$ such that

(1) $X = \sum\limits_{k=1}^{K} \lambda^k X^k$,
(2) $\lambda^k \in (0, 1]$ for all k and $\sum_{k=1}^{K} \lambda^k = 1$,
(3) $x_S^k \in \{\lfloor x_S \rfloor, \lceil x_S \rceil\}$ for all $k \in \{1, \dots, K\}$ and $S \in \mathcal{H}$.

In particular, it follows that $x^k_{S_i} = 1$ for each i and k. Thus there exists k such that $x^k_{\omega_1} = 1$. Since $x^k_{S_2} = 1$, it follows that $x^k_{\omega_2} = 0$. The latter equality and the assumption that $x^k_{S_3} = 1$ imply $x^k_{\omega_3} = 1$. Arguing inductively, it follows that $x^k_{\omega_i} = 0$ if i is even and $x^k_{\omega_i} = 1$ if i is odd. In particular, we obtain $x^k_{\omega_l} = 1$ since l is odd by assumption. Thus $x^k_{S_l} = x^k_{\omega_l} + x^k_{\omega_1} = 2$, contradicting $x^k_{S_l} = 1$. □

Proof of Theorem 2. In order to prove the Theorem, we study several cases.

- Assume there is $S \in \mathcal{H}$ such that $S = N' \times O'$ where $2 \leq |N'| < |N|$ and $2 \leq |O'| < |O|$. Let $\{i, j\} \times \{a, b\} \subseteq S$, $k \notin N'$ and $c \notin O'$ (observe that such $i, j, k \in N$ and $a, b, c \in O$ exist by the assumption of this case). Then the sequence of constraint sets

$$S_1 = S, S_2 = \{i\} \times O, S_3 = N \times \{c\}, S_4 = \{k\} \times O, S_5 = N \times \{b\},$$

  is an odd cycle together with

$$\omega_1 = (i, a), \omega_2 = (i, c), \omega_3 = (k, c), \omega_4 = (k, b), \omega_5 = (j, b).$$

  Therefore, by Lemma 1, $\mathcal{H}$ is not universally implementable.
- Assume there is $S \in \mathcal{H}$ such that, for some $i, j \in N$ and $a, b \in O$, we have $(i, a), (j, b) \in S$ with $i \neq j$ and $a \neq b$, and $(i, b) \notin S$. Then the sequence of constraint sets

$$S_1 = S, S_2 = \{i\} \times O, S_3 = N \times \{b\},$$

  is an odd cycle together with

$$\omega_1 = (i, a), \omega_2 = (i, b), \omega_3 = (j, b).$$

  Thus, by Lemma 1, $\mathcal{H}$ is not universally implementable.

By the above arguments, it suffices to consider cases where all constraint sets in $\mathcal{H}$ have one of the following forms.

(1) $\{i\} \times O'$ where $i \in N$ and $O' \subseteq O$,
(2) $N' \times O$ where $N' \subseteq N$,
(3) $N' \times \{a\}$ where $a \in O$ and $N' \subseteq N$,
(4) $N \times O'$ where $O' \subseteq O$.

Therefore it suffices to consider the following cases.

(1) Assume that there are $S', S'' \in \mathcal{H}$ such that $S' = \{i\} \times O'$ and $S'' = \{i\} \times O''$ for some $i \in N$ and some $O', O'' \subset O$, $S' \cap S'' \neq \emptyset$ and S' is neither a subset nor a superset of S''. Then we can find $a, b, c \in O$ such that $a \in O' \setminus O''$, $b \in O' \cap O''$ and $c \in O'' \setminus O'$. Fix $j \neq i$, who exists by assumption $|N| \geq 2$. Then the sequence of constraint sets

$$S_1 = S', S_2 = S'', S_3 = N \times \{c\}, S_4 = \{j\} \times O, S_5 = N \times \{a\},$$

is an odd cycle together with

$$\omega_1 = (i, a), \omega_2 = (i, b), \omega_3 = (i, c), \omega_4 = (j, c), \omega_5 = (j, a).$$

Therefore, by Lemma 1, $\mathcal{H}$ is not universally implementable.

(2) Assume that there are $S', S'' \in \mathcal{H}$ such that $S' = N' \times O$ and $S'' = N'' \times O$ for some $N', N'' \subset N$, $S' \cap S'' \neq \emptyset$ and S' is neither a subset nor a superset of S''. In such a case, we can find $i, j, k \in N$ such that $i \in N' \setminus N''$, $j \in N' \cap N''$ and $k \in N'' \setminus N'$. Fix $a, b \in O$. The sequence of constraint sets

$$S_1 = S', S_2 = S'', S_3 = N \times \{b\},$$

is an odd cycle together with

$$\omega_1 = (j, a), \omega_2 = (k, b), \omega_3 = (i, b).$$

Hence, by Lemma 1, $\mathcal{H}$ is not universally implementable.

(3) Assume that there are $S', S'' \in \mathcal{H}$ such that $S' = N' \times \{a\}$ and $S'' = N'' \times \{a\}$ for some $a \in O$ and some $N', N'' \subset N$, $S' \cap S'' \neq \emptyset$ and S' is neither a subset nor a superset of S''. This is a symmetric situation with Case 1, so an analogous argument as before goes through.

(4) Assume that there are $S', S'' \in \mathcal{H}$ such that $S' = N \times O'$ and $S'' = N \times O''$ for some $O', O'' \subset O$, $S' \cap S'' \neq \emptyset$ and S' is neither a subset nor a superset of S''. This is a symmetric situation with Case 2, so an analogous argument as before goes through.

□

Appendix F. Proof of Theorems 3 and 4

As with Bogomolnaia and Moulin (2001), a different characterization of ordinal efficiency proves useful. To this end, we first define the **minimal constraint set containing** (i,a):

$$\nu(i,a) := \bigcap_{S \in \mathcal{H}(i,a)} S,$$

if the set $\mathcal{H}(i,a) := \{S \in \mathcal{H}_2 : (i,a) \in S, \sum_{(j,b)\in S} x_{jb} = \overline{q}_S\}$ is nonempty. If $\mathcal{H}(i,a) = \emptyset$ (or equivalently $\sum_{(j,b)\in S} x_{jb} < \overline{q}_S$ for all $S \in \mathcal{H}_2$ containing (i,a)), then we let $\nu(i,a) = N \times O$.

We next define the following binary relations on $N \times O$ given X as follows:[35]

$$\begin{aligned} &(j,b) \rhd_1 (i,a) \iff i = j, b \succ_i a, \text{ and } \ x_{ia} > 0, \\ &(j,b) \rhd_2 (i,a) \iff \nu(j,b) \subseteq \nu(i,a). \end{aligned} \tag{F.1}$$

We then say

$$(j,b) \rhd (i,a) \iff (j,b) \rhd_1 (i,a) \text{ or } (j,b) \rhd_2 (i,a).$$

We say a binary relation $\rhd$ is **strongly cyclic** if there exists a finite cycle $(i_0,a_0) \rhd (i_1,a_1) \rhd \cdots \rhd (i_k,a_k) \rhd (i_0,a_0)$ such that $\rhd = \rhd_1$ for at least one relation. We next provide a characterization of ordinal efficiency.

Lemma 5. *Expected assignment X is ordinally efficient if and only if $\rhd$ is not strongly cyclic given X.*[36]

A remark is in order. In their environment, Bogomolnaia and Moulin (2001) define the binary relation $\rhd$ over the set of objects where $b \rhd a$ if there is an agent i such that $b \succ_i a$ and $x_{ia} > 0$. Bogomolnaia and Moulin show that in their environment a random assignment is ordinally efficient if and only if $\rhd$ is acyclic. Our contribution over their characterization is that we expand the domain over which the binary relation is defined to the set of agent-object pairs, in order to capture the complexity that results from a more general environment than that of BM.

Proof of Lemma 5. **"Only if" part.** First note that the following property holds.

[35] Given that $\mathcal{H}_2$ has a hierarchical structure,

$$(j,b) \rhd_2 (i,a) \iff (j,b) \in S \text{ for any } S \in \mathcal{H}_2 \text{ such that } (i,a) \in S, x_S = \overline{q}_S.$$

[36] In Kojima and Manea (2008), ordinal efficiency is characterized by two conditions, acyclicity and non-wastefulness. We do not need non-wastefulness as a separate axiom in our current formulation since a "wasteful" random assignment (in their sense) contains a strong cycle as defined here.

Claim 1. *$\rhd_1$ and $\rhd_2$ are transitive, that is,*

$$(k,c) \rhd_1 (j,b), (j,b) \rhd_1 (i,a) \Rightarrow (k,c) \rhd_1 (i,a),$$
$$(k,c) \rhd_2 (j,b), (j,b) \rhd_2 (i,a) \Rightarrow (k,c) \rhd_2 (i,a).$$

Proof. Suppose $(k,c) \rhd_1 (j,b)$ and $(j,b) \rhd_1 (i,a)$. Then, by definition of $\rhd_1$, we have $i = j = k$ and (i) $c \succ_i b$ since $(k,c) \rhd_1 (i,b)$ and (ii) $b \succ_i a$ since $(j,b) \rhd_1 (i,a)$. Thus $c \succ_i a$. Since $(j,b) \rhd_1 (i,a)$, we have $x_{ia} > 0$. Therefore $(k,c) \rhd_1 (i,a)$ by definition of $\rhd_1$.

Suppose $(k,c) \rhd_2 (j,b)$ and $(j,b) \rhd_2 (i,a)$. Then $\nu(k,c) \subseteq \nu(j,b)$ and $\nu(j,b) \subseteq \nu(i,a)$ by property (F.1). Hence $\nu(k,c) \subseteq \nu(i,a)$ which is equivalent to $(k,c) \rhd_2 (i,a)$, completing the proof by property (F.1). □

To show the "only if" part of the Theorem, suppose $\rhd$ is strongly cyclic. By Claim 1, there exists a cycle of the form

$$(i_0,b_0) \rhd_1 (i_0,a_0) \rhd_2 (i_1,b_1) \rhd_1 (i_1,a_1) \rhd_2 (i_2,b_2) \rhd_1 (i_2,a_2) \rhd_2 \cdots \rhd_1 (i_k,a_k) \rhd_2 (i_0,b_0),$$

in which every pair (i,a) in the cycle appears exactly once except for (i_0,b_0) which appears exactly twice, namely in the beginning and in the end of the cycle. Then there exists $\delta > 0$ such that a matrix Y defined by

$$y_{ia} = \begin{cases} x_{ia} + \delta & \text{if} \quad (i,a) \in \{(i_0,b_0),(i_1,b_1),\dots,(i_k,b_k)\}, \\ x_{ia} - \delta & \text{if} \quad (i,a) \in \{(i_0,a_0),(i_1,a_1),\dots,(i_k,a_k)\}, \\ x_{ia} & \text{otherwise}, \end{cases}$$

satisfies quotas. Since $\delta > 0$ and $b_l \succ_{i_l} a_l$ for every $l \in \{0,1,\dots,k\}$, Y ordinally dominates X. Therefore X is not ordinally efficient.

"If" part. Suppose X is ordinally inefficient. Then, there exists an expected assignment Y which ordinally dominates X. We then prove that $\rhd$, given X, must be strongly cyclic.

(1) **Step 1: Initiate a cycle.**
   (a)

Claim 2. *There exist $(i_0,a_0),(i_1,a_1) \in N \times O$ such that $i_0 = i_1$, $x_{i_1a_1} < y_{i_1a_1}$ and $(i_1,a_1) \rhd_1 (i_0,a_0)$ given X.*

Proof. Since Y ordinally dominates X, there exists $(i_1,a_1) \in N \times O$ such that $y_{i_1a_1} > x_{i_1a_1}$ and $y_{i_1a} = x_{i_1a}$ for all $a \succ_{i_1} a_1$. So there exists $a_0 \prec_{i_1} a_1$ with $x_{i_1a_0} > y_{i_1a_0} \geq 0$ since $x_{\{i_1\}\times N} = y_{\{i_1\}\times N}$ by assumption. Hence, we have $(i_1,a_1) \rhd_1 (i_1,a_0) = (i_0,a_0)$ given X. □

   (b) If $(i_0, a_0) \in \nu(i_1, a_1)$, then $(i_0, a_0) \rhd_2 (i_1, a_1) \rhd_1 (i_0, a_0)$, so we have a strong cycle.
   (c) Else, circle (i_1, a_1) and go to Step 2.

(2) **Step $t+1$ ($t \in \{1, 2 \ldots\}$): Consider the following cases.**
   (a) Suppose (i_t, a_t) is circled.
      (i)

Claim 3. *There exists $(i_{t+1}, a_{t+1}) \in \nu(i_t, a_t)$ such that $x_{i_{t+1}a_{t+1}} > y_{i_{t+1}a_{t+1}}$. Hence, $(i_{t+1}, a_{t+1}) \rhd_2 \nu(i_t, a_t)$.*

Proof. Note that $\nu(i_t, a_t) \subsetneqq N \times O$ since if $\nu(i_t, a_t) = N \times O$, then there exists $(i_{t'}, a_{t'})$ with $t' < t$ and $(i_{t'}, a_{t'}) \in \nu(i_t, a_t)$, so we have terminated the algorithm. Thus we have $\sum_{(i,a)\in\nu(i_t,a_t)} x_{ia} = \overline{q}_{\nu(i_t,a_t)}$. Since $x_{i_ta_t} < y_{i_ta_t}$, there exists $(i_{t+1}, a_{t+1}) \in \nu(i_t, a_t)$ such that $x_{i_{t+1}a_{t+1}} > y_{i_{t+1}a_{t+1}}$. □

      (ii) If $(i_{t'}, a_{t'}) \in \nu(i_{t+1}, a_{t+1})$ for $t' < t$, then we have a strong cycle, $(i_{t'}, a_{t'}) \rhd (i_{t+1}, a_{t+1}) \rhd \ldots \rhd (i_{t'}, a_{t'})$, and at least one $\rhd$ is $\rhd_1$.
      (iii) Else, square (i_{t+1}, a_{t+1}) and move to the next step.
   (b) Case 2: Suppose (i_t, a_t) is squared.
      (i)

Claim 4. *There exists $(i_{t+1}, a_{t+1}) \in \nu(i_t, a_t)$ such that $i_{t+1} = i_t$, $x_{i_{t+1}a_{t+1}} < y_{i_{t+1}a_{t+1}}$, and $(i_{t+1}, a_{t+1}) \rhd_1 \nu(i_t, a_t)$.*

Proof. Since (i_t, a_t) is squared, by Claim 3, $x_{i_ta_t} > y_{i_ta_t}$. Since Y ordinally dominates X, there must be $(i_{t+1}, a_{t+1}) \in \nu(i_t, a_t)$ with $i_{t+1} = i_t$ such that $x_{i_{t+1}a_{t+1}} < y_{i_{t+1}a_{t+1}}$, and $a_{t+1} \succ_{i_t} a_t$. Since $x_{i_ta_t} > y_{i_ta_t} \geq 0$, we thus have $(i_{t+1}, a_{t+1}) \rhd_1 \nu(i_t, a_t)$. □

      (ii) If $(i_{t'}, a_{t'}) \in \nu(i_{t+1}, a_{t+1})$ for $t' \leq t$, then we have a strong cycle, $(i_{t'}, a_{t'}) \rhd (i_{t+1}, a_{t+1}) \rhd \ldots \rhd (i_{t'}, a_{t'})$, and at least one $\rhd$ is $\rhd_1$.
      (iii) Else, circle (i_{t+1}, a_{t+1}) and move to the next step.

The process must end in finite steps and, at the end we must have a strong cycle. □

Given the above lemma, we are ready to proceed to the proofs of Theorems 3 and 4.

Proof of Theorem 3. Although the proof is a relatively simple modification of Theorem 1 of Bogomolnaia and Moulin (2001), we present the proof for completeness. We prove the claim by contradiction. Suppose that $PS(\succ)$ is ordinally inefficient for some $\succ$. Then, by Theorem 5 and Claim 1 there exists a strong cycle

$$(i_0, b_0) \rhd_1 (i_0, a_0) \rhd_2 (i_1, b_1) \rhd_1 (i_1, a_1) \rhd_2 (i_2, b_2) \rhd_1 (i_2, a_2) \rhd_2 \cdots \rhd_1 (i_k, a_k) \rhd_2 (i_0, b_0),$$

in which every pair (i,a) appears exactly once except for (i_0,b_0) which appears exactly twice, namely in the beginning and the end of the cycle. Let v^l and w^l be the steps of the symmetric simultaneous eating algorithm at which (i_l,a_l) and (i_l,b_l) become unavailable, respectively (that is, $(i_l,a_l) \in S^{v_l-1} \backslash S^{v_l}$ and $(i_l,a_l) \in S^{w_l-1} \backslash S^{w_l}$.) Since $(i_l,b_l) \rhd_1 (i_l,a_l)$, by the definition of the algorithm we have $w^l < v^l$ for each $l \in \{0,1,\dots,k\}$. Also, by $(i_l,a_l) \rhd_2 (i_{l+1},b_{l+1})$, we have $v^l \leq w^{l+1}$ for any $l = \{0,1,\dots,k\}$ (with notational convention $(i_{k+1},a_{k+1}) = (i_0,a_0)$.) Combining these inequalities we obtain $w^0 < v^0 \leq w^1 < v^1 \leq \cdots \leq w^k < v^k \leq w^{k+1} = w^0$, a contradiction. □

Proof of Theorem 4. Let $X = PS(\succ)$. Fix $i \in N$ and let O be ordered in the decreasing order of $\succ_i$, that is, $a_1 \succ_i a_2 \succ_i \cdots \succ_i a_{|N|}$. Let v_1 be the step in which i stops receiving probability share of a_1. In that step we have $x_{ia_1} = x^{v_1}_{ia_1} = t^{v_1}$ and there is $S_1 \in \mathcal{H}_2$ such that $(i,a_1) \in S_1$ and $X^{v_1}_{S_1} = \overline{q}_{S_1}$. Suppose $x_{ja_1} > x_{ia_1}$ for some $j \in N$. Then we have $(j,a_1) \notin S_1$ since $x_{ja_1} \leq t^{v_1} = x_{ia_1}$ if $(j,a_1) \in S_1$ by definition of the algorithm. Also $S_1 = N_1 \times \{a_1\}$ for some $N_1 \subseteq N$ with $i \in N_1$ and $j \notin N_1$ since $(i,a_1) \in S_1$ and $(j,a_1) \notin S_1$. Let Y be defined as in the definition of no feasible envy in the main text. Then, since $i \in N_1$ and $j \notin N_1$,

$$\begin{aligned} y_{S_1} &\geq \sum_{k \in N_1} x_{ka_1} - x_{ia_1} + x_{ja_1} \\ &> \sum_{k \in N_1} x_{ka_1} \\ &\geq \sum_{k \in N_1} x^{v_1}_{ka_1} \\ &= x^{v_1}_{S_1} = \overline{y}_{S_1}, \end{aligned}$$

which implies that Y is not feasible.

Let $l \geq 2$ and v_l be the step in which i stops receiving probability share of a_l. In that step we have $\sum_{m=1}^{l} x_{ia_m} = \sum_{m=1}^{l} x^{v_l}_{ia_m} = t^{v_l}$ and there is $S_l \in \mathcal{H}_2$ such that $(i,a_l) \in S_l$ and $x^{v_l}_{S_l} = \overline{q}_{S_l}$. Suppose $\sum_{m=1}^{m'} x_{ja_m} \leq \sum_{m=1}^{m'} x_{ia_m}$ for all $m' \leq l-1$ and $\sum_{m=1}^{l} x_{ja_m} > \sum_{m=1}^{l} x_{ia_m}$ for some $j \in N$. Then we have $(j,a_l) \notin S_l$ since $\sum_{m=1}^{l} x_{ja_m} \leq t^{v_l} = \sum_{m=1}^{l} x_{ia_m}$ if $(j,a_l) \in S_l$ by definition of the algorithm. Also $S_l = N_l \times \{a_l\}$ for some $N_l \subseteq N$ with $i \in N_l$ and $j \notin N_l$ since $(i,a_l) \in S_l$ and $(j,a_l) \notin S_l$. Let Y be defined as in the definition of no feasible envy in the main text. Then, since $i \in N_l$ and $j \notin N_l$,

$$\begin{aligned} y_{S_l} &\geq \sum_{k \in N_l} x_{ka_l} - x_{ia_l} + x_{ja_l} \\ &> \sum_{k \in N_l} x_{ka_l} \\ &\geq \sum_{k \in N_l} x^{v_l}_{ka_l} \\ &= x^{v_l}_{S_l} = \overline{q}_{S_l}, \end{aligned}$$

which implies that Y is not feasible. By induction, we complete the proof. □

Appendix G. Proof of Theorem 6

First, define a price space $\mathcal{P} = [0, |N|b^*]^{|O|}$. We then define for each agent i his demand correspondence $d^*_i(\cdot)$ in the usual manner. That is, for each $\mathbf{p} \in \mathcal{P}$, $d^*_i(\mathbf{p})$ is the set of fractional consumption bundles that maximize the utility of agent i subject to the constraint that the total expenditure is at most b^* under prices $\mathbf{p}$. By standard arguments, for each i, his demand correspondence $d^*_i(\mathbf{p})$ is nonempty and convex-valued for all $\mathbf{p} \in \mathbb{R}^{|O|}_+$, and upper hemicontinuous in $\mathbf{p}$.

Define the excess demand correspondence $z(\cdot)$ by $z(\mathbf{p}) = \sum_i d^*_i(\mathbf{p}) - \mathbf{q}$ for each $\mathbf{p} \in \mathcal{P}$. Note that this correspondence is also upper hemicontinuous and convex-valued because it is a linear sum of upper hemicontinuous and convex-valued correspondences. Introduce the following objects:[37]

(1) Let $\overline{q} = \max\{\max_{a \in O} q_a, \max_{i \in N, a \in O} q_{\{(i,a)\}}\}$.
(2) Define an auxiliary enlargement of the price space, $\tilde{\mathcal{P}} = [-\overline{q}, |N|b^* + |N|\overline{q}]^{|O|}$.
(3) Define a truncation function $t : \tilde{\mathcal{P}} \to \mathcal{P}$ by $t(\mathbf{p}) = \left(\max\{0, \min\{p_a, |N|b^*\}\}\right)_{a \in O}$.

Let a correspondence $f : \tilde{\mathcal{P}} \to \tilde{\mathcal{P}}$ be defined by $f(\mathbf{p}) = t(\mathbf{p}) + z(t(\mathbf{p}))$. We will show that we can apply Kakutani's fixed point theorem. To do so, first note that $z(t(\mathbf{p}))$ is upper hemicontinuous and convex-valued on $\tilde{\mathcal{P}}$ because $t(\cdot)$ is a continuous function and $z(\cdot)$ is an upper hemicontinuous and convex-valued correspondence. This implies that $f(\mathbf{p})$ is upper hemicontinuous and convex-valued as well. Second, note that the range of $t(\mathbf{p}) + z(t(\mathbf{p}))$ lies in $\tilde{\mathcal{P}}$ as required because, for any $\mathbf{p} \in \tilde{\mathcal{P}}$ and $a \in O$, the excess demand $z_a(\mathbf{p})$ is at least $-\overline{q}$ (because the supply of object a is $q_a \leq \overline{q}$) and at most $|N|\overline{q}$ (because the demand of object

[37] These objects prove useful in handling some boundary issues that arise because we allow objects to be in excess supply at price zero (and preferences are satiable, so prices of zero may actually arise).

a by any agent i is at most $q_{\{(i,a)\}} \le \bar{q}$). Thus $f(\mathbf{p})$ is an upper hemicontinuous and convex-valued correspondence defined on the compact and convex set $\tilde{\mathcal{P}}$. Thus by Kakutani's fixed point theorem, there exists a fixed point $\mathbf{p}^* \in f(\mathbf{p}^*)$.

To complete the proof, we will show that any fixed point $\mathbf{p}^*$ of $f(\cdot)$ induces a competitive equilibrium; specifically, $t(\mathbf{p}^*)$ is a competitive equilibrium price vector. To show this claim, suppose that $\mathbf{p}^*$ is a fixed point of $f(\cdot)$. By the definition of a fixed point and correspondence $f(\cdot)$, this means that there exists $\mathbf{z}^* = [z^*_a]_{a\in O} \in z(t(\mathbf{p}^*))$ such that $\mathbf{p}^* = t(\mathbf{p}^*) + \mathbf{z}^*$, or equivalently $p^*_a = t_a(\mathbf{p}^*) + z^*_a$ for all $a \in O$. First suppose that $p^*_a \in [0, |N|b^*]$. The truncation does not bite for such an object a, that is, $t_a(\mathbf{p}^*) = p^*_a$. Then $p^*_a = t_a(\mathbf{p}^*) + z^*_a$ implies $z^*_a = 0$ (i.e., the demand and supply for object a exactly clear at $t(\mathbf{p}^*)$). Second, suppose that $p^*_a < 0$. Then $t_a(\mathbf{p}^*) = 0$ and hence $p^*_a = t_a(\mathbf{p}^*) + z^*_a$ implies $z^*_a = p^*_a < 0$.[38] Lastly, suppose that $p^*_a > |N|b^*$. Then $t_a(\mathbf{p}^*) = |N|b^*$ and hence $p^*_a = t_a(\mathbf{p}^*) + z^*_a$ implies that $z^*_a = p^*_a - |N|b^* > 0$ (i.e., object a is in excess demand at $t(\mathbf{p}^*)$). But this is impossible because $t_a(\mathbf{p}^*) = |N|b^*$, so even if all agents spend their entire budget on object a at price vector $t(\mathbf{p}^*)$, total demand is less than or equal to one (which is weakly less than supply by assumption). These arguments complete the proof.

Appendix H. Proof of Theorem 9

Proof. For each $i \in N$, let $(a^1_i, a^2_i, \dots, a^{|O|}_i)$ be a sequence of objects in decreasing order of i's preferences so that $v_{ia^1_i} \ge v_{ia^2_i} \ge \dots, v_{ia^{|O|}_i}$. Define the class of sets $\mathcal{H}' = \mathcal{H}'_1 \cup \mathcal{H}'_2$ by

$$\mathcal{H}'_1 = \mathcal{H}_1 \cup \left(\bigcup_{\substack{i\in N,\\ k\in\{1,\dots,|O|\}}} \{i\} \times \{a^1_i, \dots, a^k_i\} \right),$$

$$\mathcal{H}'_2 = \mathcal{H}_2.$$

By inspection, $\mathcal{H}'$ is a bihierarchy. Therefore, by Theorem 1, there exists a convex decomposition such that

$$\sum_{(i,a)\in S} x'_{ia}, \sum_{(i,a)\in S} x''_{ia} \in \left\{ \left\lfloor \sum_{(i,a)\in S} x_{ia} \right\rfloor, \left\lceil \sum_{(i,a)\in S} x_{ia} \right\rceil \right\} \quad \text{for all } S \in \mathcal{H}', \tag{H.1}$$

for any integer-valued matrices X' and X'' that are part of the decomposition. In particular, property (H.1) holds for each $\{(i,a)\} \in \mathcal{H}'_1$ and $\{i\} \times \{a^1_i, \dots, a^k_i\} \in \mathcal{H}'_1$. This means that

[38] This means that object a is in excess supply at $t(\mathbf{p}^*)$. Note that this excess supply does not cause a problem because $t_a(\mathbf{p}^*) = 0$, which is allowed by the "complementary slackness" condition in the definition of the mechanism.

- **Observation 1:** For any i and k, $x'_{ia_i^k} - x''_{ia_i^k} \in \{-1, 0, 1\}$. This follows from the fact that $|x'_{ia_i^k} - x''_{ia_i^k}| \leq \lceil x_{ia_i^k} \rceil - \lfloor x_{ia_i^k} \rfloor \leq 1$ and that $x'_{ia_i^k}$ and $x''_{ia_i^k}$ are integer valued.
- **Observation 2:** By the same logic as for Observation 1, it follows that $\sum_{j=1}^{k}(x'_{ia_i^j} - x''_{ia_i^j}) \in \{-1, 0, 1\}$ for any i and k.
- **Observation 3:** Let $(a_i^{k_l})_{l=1}^{\bar{l}}$ be the (largest) subsequence of $(a_i^1, \ldots, a_i^k)$ such that $x'_{ia_i^{k_l}} \neq x''_{ia_i^{k_l}}$ for all l. Then, (i) $x_{ia_i^{k_l}} \notin \mathbb{Z}$ for all l, and (ii) $x'_{ia_i^{k_{2l'}}} - x''_{ia_i^{k_{2l'}}} = -(x'_{ia_i^{k_{2l'-1}}} - x''_{ia_i^{k_{2l'-1}}})$ for any $l' = 1, \ldots, \bar{l}/2$.

  Observation 3 (ii) can be shown as follows. First, the result must hold for $l' = 1$, or else $\sum_{j=1}^{k_2}(x'_{ia_i^j} - x''_{ia_i^j}) = x'_{ia_i^{k_1}} - x''_{ia_i^{k_1}} + x'_{ia_i^{k_2}} - x''_{ia_i^{k_2}} \in \{-2, 2\}$, which violates Observation 2. Now, working inductively, suppose the statement holds for all $l' = 1, \ldots, m-1$ for $m \leq \bar{l}/2$. Then the statement must hold for $l' = m$, or else

$$
\begin{aligned}
&\sum_{j=1}^{k_{2m}}(x'_{ia_i^j} - x''_{ia_i^j}) \\
= &\sum_{l'=1}^{m-1}\left(x'_{ia_i^{k_{2l'-1}}} - x''_{ia_i^{k_{2l'-1}}} + x'_{ia_i^{k_{2l'}}} - x''_{ia_i^{k_{2l'}}}\right) + x'_{ia_i^{k_{2m-1}}} - x''_{ia_i^{k_{2m-1}}} + x'_{ia_i^{k_{2m}}} - x''_{ia_i^{k_{2m}}} \\
= &\ x'_{ia_i^{k_{2m-1}}} - x''_{ia_i^{k_{2m-1}}} + x'_{ia_i^{k_{2m}}} - x''_{ia_i^{k_{2m}}}
\end{aligned}
$$

  must be either -2 or 2, which again violates Observation 2.

These observations imply that

$$
\begin{aligned}
\sum_{a \in O} x'_{ia} v_{ia} - \sum_{a \in O} x''_{ia} v_{ia} &= \sum_{k=1}^{|O|}(x'_{ia_i^k} - x''_{ia_i^k}) v_{ia_i^k} \\
&= \sum_{l=1}^{\bar{l}}(x'_{ia_i^{k_l}} - x''_{ia_i^{k_l}}) v_{ia_i^{k_l}} \\
&\leq \sum_{l'=1}^{\bar{l}/2} v_{ia_i^{k_{2l'-1}}} - v_{ia_i^{k_{2l'}}} \\
&\leq v_{ia_i^{k_1}} - v_{ia_i^{k_{\bar{l}}}} \\
&\leq \Delta_i,
\end{aligned}
$$

where the first inequality follows from $v_{ia_i^k} \geq v_{ia_i^{k'}}$ for $k < k'$ and Observations 1 and 3-(ii), the second inequality follows from $v_{ia_i^k} \geq v_{ia_i^{k'}}$ for $k < k'$, and the last inequality follows from the definition of Δ_i and Observation 3-(i). Therefore, we obtain property (1). Property (2) follows immediately from property (1). □

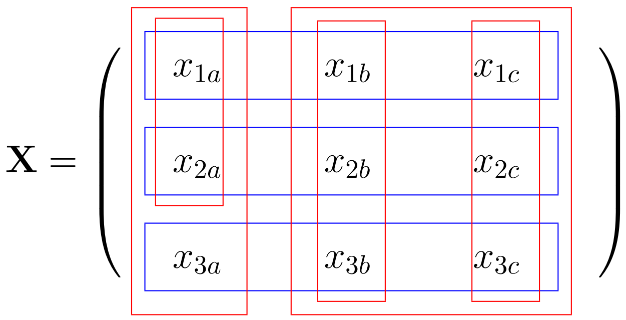

$\mathbf{X} =$
x_{1a} x_{1b} x_{1c}
x_{2a} x_{2b} x_{2c}
x_{3a} x_{3b} x_{3c}

Figure 1. Endogenous Capacities and Group-Specific Quotas

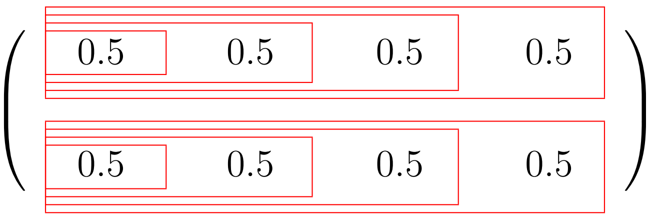
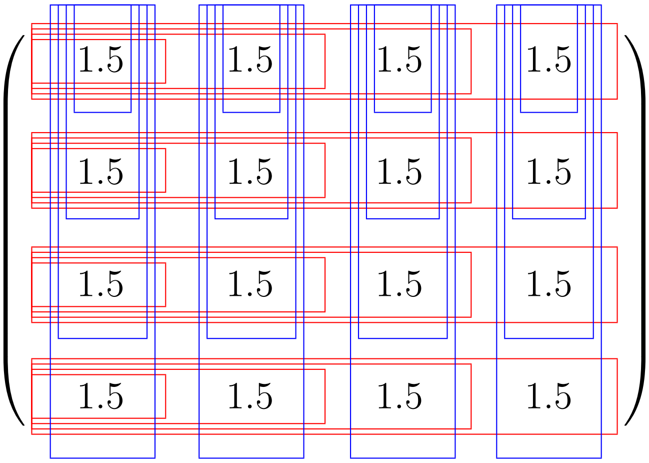


Figure 3. Illustration for Utility Guarantee



$$\begin{pmatrix} 1.5 & 1.5 & 1.5 & 1.5 \\ 1.5 & 1.5 & 1.5 & 1.5 \\ 1.5 & 1.5 & 1.5 & 1.5 \\ 1.5 & 1.5 & 1.5 & 1.5 \end{pmatrix}$$

Figure 4. Interleage Matchup Design

$$\begin{pmatrix} 2 & 1 & 1 & 2 \\ 1 & 2 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 2 & 1 & 1 & 2 \end{pmatrix}$$

Figure 5. Feasible Fair Matchup