IMPLEMENTING RANDOM ASSIGNMENTS: A GENERALIZATION OF THE BIRKHOFF-VON NEUMANN THEOREM

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

ABSTRACT. The literature on random mechanisms often describes outcomes incompletely as "random assignments" — expressing the expected number of objects of each type assigned to different agents — and a set of feasibility constraints that a pure assignment must satisfy. We provide a necessary and sufficient condition (the "bihierarchy" condition) for the set of constraints to have the property that if the random assignment satisfies them, then it is implementable by a lottery over feasible pure assignments. Our theorem maximally generalizes the celebrated Birkhoff-von Neumann theorem. We also provide an algorithm to implement any such random assignment. Several applications are described, including (i) single-unit random assignment, such as school choice; (ii) multi-unit random assignment, such as course allocation and fair division; and (iii) two-sided matching problems, such as the scheduling of inter-league sports matchups. The same method also finds applications outside economics, generalizing previous results on the minimize makespan problem in the computer science literature.

KEYWORDS: Birkhoff-von Neumann Theorem, Market Design, Random Assignment, Probabilistic Serial, Utility Guarantee, Makespan, Maximal Domain, Fair Allocation, Santa Claus Problem, Optimal Assignment, Assignment Message.

Date: February 15, 2010.

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 who pointed out close connections between our analysis and the integer programming literature. 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.

1. Introduction

Randomness plays an important role in mechanism design. In some cases, randomness is required for fairness reasons, as with the allocation of slots in public schools, course seats or dormitory rooms at universities, and organs amongst patients needing transplants. In these problems the objects assigned are indivisible and monetary transfers cannot be used to equalize outcomes, so deterministic procedures can be very unfair. Randomizing over outcomes can restore *ex ante* fairness. Random mechanisms may also be recommended by efficiency or revenue considerations. For instance, the optimal auction mechanism involves random allocations when a seller faces a non-monotonic marginal revenue function (Myerson, 1981) or when agents are budget constrained (Che and Gale, 2000; Maskin, 2000; Pai and Vohra, 2009).

A complete description of a random outcome consists of a probability distribution over feasible pure outcomes. Yet, the literature often proceeds instead with a "random assignment" — a profile describing the expected number of each object type assigned to each agent. Random assignments describe outcomes concisely; a random assignment is represented by a matrix $P = [P_{ia}]$ where its generic entry P_{ia} is the expected number of object a assigned to agent i. Further, concepts of efficiency, incentive compatibility, and fairness can sometimes be defined and analyzed more naturally by working directly with the random assignment matrix. For instance, Hylland and Zeckhauser (1979) and Bogomolnaia and Moulin (2001) develop mechanisms that directly produce random assignments and analyze their mechanisms' properties based on random assignments.

Clearly, every random outcome corresponds to some random assignment, but is the converse also true? That is, if a mechanism prescribes an arbitrary random assignment, will we be able to *implement* it by conducting some lottery over pure assignments? And if the answer is affirmative, then is there a fast algorithm to realize that lottery?

The celebrated Birkhoff-von Neumann theorem (Birkhoff, 1946; von Neumann, 1953) supplies a partial answer to these questions. It asserts that any bistochastic matrix — a square non-negative matrix whose rows and columns sum to one — can be expressed as a convex combination of permutation matrices — bistochastic matrices whose elements are all zero or one. The theorem ensures that any random assignment of n objects to n agents, with one object per agent, can be implemented.[1] In this problem, the expected allocation is a vector of probabilities and the relevant constraints are that each item and each agent are assigned exactly once.

[1]Birkhoff's proof provides a constructive algorithm for the implementation. For a nice description, see Lovász and Plummer (1986, Corollary 1.4.15).

Bistochastic matrices are characterized by a set of inequalities (each element is non-negative) and equations fixing the sum for each row and column. Real-world assignment problems sometimes omit or modify some of the row and column constraints or add constraints with other forms. For example, in school choice problems each school has capacity for multiple students, allowing a column sum greater than one. And, some students may be unassigned to a public school, opting for a private school outside the matching system, allowing a row sum of zero. In course-allocation problems students seek schedules with multiple courses, and courses have openings for multiple students, allowing for both row and column sums greater than one. In a house exchange problem, some agent may supply a house, so its corresponding matrix entry can be negative. When a school district imposes minimum or maximum quotas on the number of students in a school from a particular group defined by gender, race, income, ethnicity or residence, entries are summed over subsets of a single column. When a public school authority wishes to install multiple school programs in one building, allowing the enrollments to respond to demand, but subject to an overall constraint imposed by the building size, entries are summed over multiple columns. In a course allocation problem, a student may wish to enroll in no more than a certain number of courses in a given subject, constraining a sum of elements in a subset of one row.

Are random assignments implementable when the set of allowable constraints is broadened in these ways? If so, can we accommodate an even wider set of constraints? To explore these questions, we introduce a model of indivisible good assignment in which the expected quantity of any good assigned to an individual can be any real number. Feasibility of a pure allocation is described by integer upper and lower bounds on the sums of entries in subsets of entries of the assignment matrix. We call a collection of such subsets a **constraint structure** and say that a constraint structure is "decomposable" if *every* random assignment satisfying constraints with that structure is implementable.

Our first theorem identifies a sufficient condition — the bihierarchy condition — for a constraint structure to be decomposable. A constraint structure is a "hierarchy" if, for any two sets in the collection, either they are disjoint or one set is a subset of the other, and it is a "bihierarchy" if it is the union of two hierarchies. We show that if a constraint structure is a bihierarchy, then it is decomposable, that is, any random assignment satisfying constraints on that structure can be implemented.

This condition informs when a mechanism dealing with random assignments can be useful. To actually use such a mechanism, one must be able to implement random assignments sufficiently fast. We provide a polynomial-time algorithm for implementing random assignments under bihierarchical constraints. Our algorithm is random: it does not explicitly produce the entire convex combination of integer-valued matrices. Rather, it reaches each element of the convex combination with the right probability.

The bihierarchy structure includes the Birkhoff-von Neumann constraint structure as a special case, for the row-and-column constraint structure of that theorem is described by a row hierarchy and a column hierarchy. Importantly, the bihierarchy structure allows many other sorts of constraints of potential interest for market design, including all of the sample constraints described above.

Yet there are constraint structures that are not bihierarchies. Is further generalization possible? Our second theorem shows that the answer is negative: Our result is the maximal generalization of the Birkhoff-von Neumann Theorem. More precisely, if the constraint structure include the rows and columns but is not a bihierarchy, then there exists a random assignment that satisfies all the constraints but is not implementable. Constraints on rows and columns are commonly present in bilateral matching, because they reflect overall limits on the supply of an item and on the number of items assigned to an individual. What does this result say about multilateral matching, as for tasks that require more than two specialists? Or about bilateral matching with just one type of agent (the "roommate" problem)? With minor exceptions (which we will identify), the constraint structures associated with these problems fail the bihierarchy condition, and the expected allocations in these cases are generally not decomposable.

Our extension of the Birkhoff-von Neumann theorem can be used to create new mechanisms in a number of market design problems, and our maximal domain result tells what kinds of random mechanisms are *not* possible. In this paper we discuss two kinds of applications in which we exploit the bihierarchical structure to create new mechanisms for market design.

The first kind of application extends existing random assignment mechanisms to accommodate important new kinds of constraints. We illustrate how Bogomolnaia and Moulin's (2001) probabilistic serial mechanism can be extended to accommodate group-specific quota and endogenous capacity constraints, for possible application to the school-choice problem. We also suggest how Hylland and Zeckhauser's (1979) pseudo-market mechanism can be extended to accommodate multi-unit demand with various constraints, for possible application to the course-allocation problem. These mechanisms are known to have desirable welfare properties. Yet, we are not aware of any real-life assignment problems in which these mechanisms have been regularly employed. We believe that our extensions will enhance the practical applicability of these mechanisms.

We call our second kind of application the "utility guarantee" procedure: it aims to reduce the randomness of *ex post* utility outcomes and so to increase perceived fairness in certain multi-unit assignment problems, such as course allocation, task assignment, and the fair division of estates. One attractive method for achieving efficiency and *ex ante* fairness in such problems is to treat objects as "divisible," solve for an efficient and fair random assignment, and then implement it by a lottery over pure assignments. A potential difficulty is that there may be many ways to implement a given random assignment, some of which entail pure assignments that are quite unfair *ex post*. To mitigate that, we introduce "utility proximity" constraints on a random assignment (in addition to original constraints), so that the whole set of constraints still respects the bihierarchy constraint structure. These constraints allow us to limit the distance between the agents' realized utilities and their expected utilities from the random assignment. Specifically, the difference between any agent's highest and lowest realized utilities is at most the difference in value between his most- and least-preferred fractionally assigned objects. This utility-guarantee technique, together with a fair random assignment, produces a pure outcome that has desirable *ex post* fairness features.

We show that this utility-guarantee approach can also be adapted to two-sided matching problems, in which both sides of the market are agents. Starting with any random 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 matchups in sports scheduling or a fair speed-dating mechanism.

Related Literature. Extensions of the Birkhoff-von Neumann theorem similar to ours have been obtained by other researchers. Watkins and Merris (1974), Lewandowski, Liu, and Liu (1986) and de Werra (1984) generalized the theorem with linear programming and network flow applications in mind. They formulate and prove a version that applies to bihierarchical constraint structures that include only supersets of rows and columns of the matrix - a restriction that excludes some of our important applications. In another precursor, Milgrom (2008) extends the assignment game (Shapley and Shubik, 1972) to allow preferences over arbitrary bundles of goods. He represents those preferences with a particular linear model that incorporates a bihierarchy of constraints and shows that a competitive equilibrium with integer assignments exists, that the bihierarchy constraints describe substitutable goods, and that certain of the constraints must form a hierarchy for the substitutes property to be satisfied generally.

This paper extends this literature in two ways. First, as described above, our extension of the Birkhoff-von Neumann theorem is maximal, and the extra generality in our formulation is needed for some of our suggested applications. Second, we provide a polynomial time random algorithm which implements a random assignment as a lottery over pure assignments. The problem of randomizing over vertices in this way does not arise in linear programming, and this problem appears not to have been studied previously.

Also related to our work is the problem of *reduced-form* implementation in the auction literature (Maskin and Riley, 1984; Matthews, 1984; Moore, 1984; Border, 1991). Like many assignment problems, auction mechanism design problems are often usefully approached in terms of the marginal distribution of allocations, which are profiles of interim allocation rules, each expressing the probability of each bidder obtaining the good as a function of his type. The question then is which interim allocations can be implemented by a lottery over feasible pure allocations. This method has been useful in characterizing optimal mechanisms in the presence of bidder risk aversion (Maskin and Riley, 1984), budget constraints (Maskin, 2000; Pai and Vohra, 2009), or cartel mechanisms with limited transfers (Che, Condorelli, and Kim, 2009). This literature has so far confined its attention to single-good auction problems. The current paper addresses a similar problem for the assignment of multiple goods to multiple agents.

Finally, the paper is related to the literature on market design. Our generalization of the Birkhoff-von Neumann Theorem can be applied to generalize random mechanisms such as the pseudo-market mechanism (Hylland and Zeckhauser, 1979) and the probabilistic serial mechanism (Bogomolnaia and Moulin, 2001). The utility-guarantee result can be applied to the course allocation problem (Budish, 2009; Budish and Cantillon, 2009; Sönmez and Ünver, 2008) and two-sided matching (Roth and Sotomayor, 1990).[2]

The rest of the paper is organized as follows. Section 2 presents the model. Section 3 offers the main theorems. Section 4 illustrates the application of the theorem to single-unit assignment. In section 5 we provide the utility-guarantee theorem and apply it to multi-unit assignment. Section 6 offers the utility-guarantee theorem in the two-sided matching setting. In Section 7 we generalize to applications besides two-sided matching. Section 8 describes a polynomial time algorithm for implementing a random assignment. Section 9 concludes.

[2]The utility-guarantee method can also be applied to the minimize makespan problem (Lenstra, Shmoys, and Tardos, 1990) of the computer science literature. See below.

2. Setup

An **environment** is a tuple $\mathcal{E} = \langle N, O, \mathcal{H}, q\rangle$ where N and O are sets of agents and objects where $|N|, |O| \geq 2$; $\mathcal{H}$ is a set of subsets of $N \times O$ that includes all singletons (we call $\mathcal{H}$ a **constraint structure**); and $q = (\underline{q}_S, \overline{q}_S)_{S\in\mathcal{H}}$ is the set of quotas associated with each set in $\mathcal{H}$. We call $\underline{q}_S$ the **floor constraint** and $\overline{q}_S$ the **ceiling constraint** for S. For each $S \in \mathcal{H}$, we assume $\underline{q}_S \in \mathbb{Z} \cup \{-\infty\}$ and $\overline{q}_S \in \mathbb{Z} \cup \{\infty\}$, where $\mathbb{Z}$ is the set of integers.

A (generalized) **random assignment** is a $|N| \times |O|$ matrix $P = [P_{ia}]$ where $P_{ia} \in (-\infty, \infty)$ for all $i \in N, a \in O$. A **deterministic assignment** is a random assignment P each of whose entries is an integer. Note that we allow for assigning more than one unit of a good and even for assigning a negative amount of a good. One interpretation of receiving a negative amount of a good is supplying the good. Given environment $\mathcal{E} = \langle N, O, \mathcal{H}, q\rangle$, P is said to be **feasible in $\mathcal{E}$** if

$$\underline{q}_S \leq P_S \leq \overline{q}_S, \text{ for all } \quad S \in \mathcal{H},$$

where we define

$$P_S := \sum_{(i,a)\in S} P_{ia},$$

for any random assignment P and $S \in \mathcal{H}$.

Definition 1. *The constraint structure $\mathcal{H}$ is **decomposable** if, for any random assignment P and any quotas $(\underline{q}_S, \overline{q}_S)_{S\in\mathcal{H}}$, such that $\underline{q}_S \leq P_S \leq \overline{q}_S$ for all $S \in \mathcal{H}$, there exist $\lambda^1, \dots, \lambda^K$ and $P^1, \dots, P^K$ such that*

(1) $P = \sum_{k=1}^K \lambda^k P^k$,
(2) $\lambda^k > 0, k = 1, \dots, K,$ *and* $\sum_{k=1}^K \lambda^k = 1$,
(3) $\underline{q}_S \leq P^k_S \leq \overline{q}_S$ *for each* $k = 1, \dots, K$ *and* $S \in \mathcal{H}$,
(4) P^k_{ia} *is an integer for each* (i, a).

If $\mathcal{H}$ is decomposable, then every P satisfying all the given constraints in $\mathcal{H}$ can be expressed as a convex combination of deterministic assignments satisfying the constraints. In other words, any random assignment satisfying the constraints in $\mathcal{H}$ can be implemented as a lottery over deterministic outcomes each of which respects constraints in $\mathcal{H}$.

Decomposability of a constraint structure has another, more convenient, formulation. Since $\mathcal{H}$ is decomposable only if conditions (1)-(4) hold for *any* matrix P and quotas $\{q_S\}$ satisfying the feasibility inequalities, the conditions must hold in particular for quotas

$\underline{q}_S = \lfloor P \rfloor$ and $\overline{q}_S = \lfloor P \rceil$.[3] Conversely, for any given P, a decomposition that satisfies (1)-(4) with quotas $\underline{q}_S = \lfloor P \rfloor$ and $\overline{q}_S = \lfloor P \rceil$ also satisfies the conditions for every other vector of quotas such that P is feasible. Consequently, $\mathcal{H}$ is decomposable if and only if for every random assignment P there exist $\lambda^1, \ldots, \lambda^K$ and $P^1, \ldots, P^K$ such that

(1) $P = \sum_{k=1}^{K} \lambda^k P^k$,
(2) $\lambda^k > 0, k = 1, \ldots, K$, and $\sum_{k=1}^{K} \lambda^k = 1$,
(3) $P_S^k \in \{\lfloor P_S \rfloor, \lceil P_S \rceil\}$ for all $k \in \{1, \ldots, K\}$ and $S \in \mathcal{H}$.

This alternative formulation requires that each assignment in the decomposition rounds the random assignment either up or down to the nearest integer, with respect to every constraint set.

Note that our notion of decomposability is defined for a constraint structure without reference to specific quotas. This approach has at least two advantages. First, our formulation produces a sharp result: A necessary and sufficient condition for decomposability will be obtained in terms of constraint structures. Second, and more importantly, specific quotas may vary over time or with market conditions, so it is desirable to characterize a robust condition under which a random mechanism is guaranteed to work for any possible quota.

3. The Rounding Theorem

A constraint structure $\mathcal{H}$ is a **hierarchy** if $S \subset S'$ or $S' \subset S$ or $S \cap S' = \emptyset$ for every $S, S' \in \mathcal{H}$.[4] The following concept plays a central role in the rest of this paper.

Definition 2. *A set $\mathcal{H} \subset 2^{N \times O}$ is a* ***bihierarchy*** *if there exist $\mathcal{H}_N$ and $\mathcal{H}_O$ such that*

(1) *$\mathcal{H} = \mathcal{H}_N \cup \mathcal{H}_O$ and $\mathcal{H}_N \cap \mathcal{H}_O = \emptyset$, that is, $\mathcal{H}_N$ and $\mathcal{H}_O$ partition $\mathcal{H}$, and*
(2) *$\mathcal{H}_N$ and $\mathcal{H}_O$ are hierarchies.*

In many applications $\mathcal{H}_N$ and $\mathcal{H}_O$ include, respectively, sets of the form $\{i\} \times O$ and $N \times \{a\}$ where $i \in N$, $a \in O$. These sets represent constraints imposed on each agent and object, thus the mnemonic notation $\mathcal{H}_N$ and $\mathcal{H}_O$. However, at this point we do not impose such a restriction, and we will state the restriction whenever applicable. (Partitions of the sets $\mathcal{H}$ into $\mathcal{H}_N$ and $\mathcal{H}_O$ need not be unique, either. For instance, singleton sets can be partitioned into the two families in any arbitrary fashion.)

Theorem 1. (Rounding Theorem) *If $\mathcal{H}$ is a bihierarchy, then it is decomposable.*

[3]For any $x \in \mathbb{R}$, $\lfloor x \rfloor$ and $\lceil x \rceil$ are the largest integer no larger than x and the smallest integer no smaller than x, respectively.

[4]A hierarchy is called a laminar family in the combinatorial optimization literature.

Theorem 1 shows that any random assignment P can be decomposed into matrices where the sum of the entries within each element of the bihierarchy is rounded up or down to the nearest integer. Its proof — in fact, a proof for a more general result — appears in Appendix A.

3.1. **Corollary: the Birkhoff-von Neumann Theorem.** We denote the environment studied by Birkhoff and von Neumann by $\mathcal{E}^{BvN} \equiv \langle N, O, \mathcal{H}^{BvN}, q\rangle$ where

$$\mathcal{H}^{BvN} = \{\{(i,a)\}|(i,a) \in N \times O\} \cup \{\{i\} \times O|i \in N\} \cup \{N \times \{a\}|a \in O\}$$
$$\underline{q}_{\{(i,a)\}} = 0, \overline{q}_{\{(i,a)\}} = 1, \quad \text{for all } (i,a) \in N \times O,$$
$$\underline{q}_S = \overline{q}_S = 1, \quad \text{for all } S \in \mathcal{H} \setminus \{\{(i,a)\}|(i,a) \in N \times O\}.$$

This is an environment in which each agent receives exactly one object and each object is allocated to exactly one agent, and no other constraints are imposed. $\mathcal{H}^{BvN}$ is a bihierarchy since it can be partitioned, for example, into $\mathcal{H}^{BvN}_N$ and $\mathcal{H}^{BvN}_O$ where

$$\mathcal{H}^{BvN}_N := \{\{(i,a)\}|(i,a) \in N \times O\} \cup \{\{i\} \times O|i \in N\},$$
$$\mathcal{H}^{BvN}_O := \{N \times \{a\}|a \in O\},$$

and clearly $\mathcal{H}^{BvN}_N$ and $\mathcal{H}^{BvN}_O$ are hierarchies.

A random assignment feasible in $\mathcal{E}^{BvN}$ is called a **bistochastic matrix** or a doubly stochastic matrix. Equivalently, P is a bistochastic matrix if

(1) $P_{ia} \geq 0$ for all $i \in N$ and $a \in O$,
(2) $\sum_{a \in O} P_{ia} = 1$ for all $i \in N$, and
(3) $\sum_{i \in N} P_{ia} = 1$ for all $a \in O$.

An integer-valued bistochastic matrix is called a permutation matrix. Since $\mathcal{H}^{BvN}$ is a bihierarchy, the following Birkhoff-von Neumann Theorem is an immediate corollary of the Rounding Theorem.

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

3.2. **Necessity of a bihierarchical constraint structure.** Theorem 1 shows that bihierarchy is sufficient for decomposition. This section examines the sense in which it is necessary. Doing so also provides an intuition about the role bihierarchy plays for implementation of random assignments. We begin with an example of a non-bihierarchical constraint structure that is not decomposable.

Example 1. *Consider the following environment with 2 goods and 2 agents and the constraint structure*

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

Clearly, $\mathcal{H}$ is not a bihierarchy. Suppose each set in $\mathcal{H}$ has a common floor and ceiling quota of one. The following random assignment

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

cannot be decomposed into feasible deterministic assignments. To see this first observe that, for any convex decomposition of P, there exists P^k that is part of the decomposition of P with $P^k_{1a} = 1$. Since the constraint set $\{(1,a),(1,b)\}$ has a quota of one, it follows that $P^k_{1b} = 0$. Since the quota of one binds for $\{(1,b),(2,a)\}$, it follows that $P^k_{2a} = 1$. This is a contradiction because $P^k_{\{(1,a),(2,a)\}} = P^k_{1a} + P^k_{2a} = 2$ violates the quota for $\{(1,a),(2,a)\}$, which is one.

Example 1 suggests that the failure of decomposability is caused by a "cycle" of an odd number formed by constraint sets. In the above example, for instance, a cycle formed by three constraint sets $\{(1,a),(1,b)\},\{(1,a),(2,a)\},\{(1,b),(2,a)\}$ leads to a situation where at least one of the constraints is violated. Generalizing this idea, we say that a sequence $(S_1,\dots,S_l) \in \mathcal{H}^l$ is an **odd cycle** if $S_i \neq S_j$ for all $i \neq j$, l is odd, and there exists a sequence $(x_1,\dots,x_l) \in (N \times O)^l$ such that for each $i = 1,\dots,l$, $x_i \in S_i \cap S_{i+1}$ 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 $\mathcal{H}$ contains an odd cycle, then $\mathcal{H}$ is not decomposable.*

An important role of the bihierarchy is to rule out odd cycles. To see this, suppose that $\mathcal{H} = \mathcal{H}_N \cup \mathcal{H}_O$ is a bihierarchy that contains an odd cycle, $\{S_1,\dots,S_l\}$. Assume without loss $S_1 \in \mathcal{H}_N$. Then, S_2 must belong to $\mathcal{H}_O$, since $S_1 \cap S_2 \neq \emptyset$ and neither is a subset of the other (since $x_2 \in S_2 \setminus S_1$ and $x_l \in S_1 \setminus S_2$). Arguing in the same fashion, S_3 must be in $\mathcal{H}_N$, S_4 in $\mathcal{H}_O$, ..., and S_l must be in $\mathcal{H}_N$ since l is an odd number. But $S_l \cap S_1 \neq \emptyset$ and neither is a subset of the other. So $\mathcal{H}_N$ cannot be a hierarchy, and $\mathcal{H}$ cannot be a bihierarchy, a contradiction.

Is bihierarchy necessary for decomposition? The next example suggests that this is not the case.[5]

[5]The argument presented in the example does not *show* that the constraint structure is decomposable since it limits attention to just a single set of quota constraints. It turns out that the dual of the constraint

Example 2. Consider an environment with 2 goods 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. Any feasible random assignment

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

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

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

Note that the constraint structure does not allow for an odd cycle although it is not a bihierarchy.

Notice, however, that Example 2 is somewhat non-standard in that some row and column constraints are not present. Decomposability would fail if all row and column constraints are added to the constraint structure of Example 2 as the new constraint structure has an odd cycle. This observation turns out to be true more generally. We show that *bihierarchy is in fact necessary for decomposability in an important sense* — namely, whenever all the "standard" constraint sets are present.

Theorem 2. (NECESSITY) *Suppose $\mathcal{H}^{BvN} \subset \mathcal{H}$. If $\mathcal{H}$ is not a bihierarchy, then it is not decomposable.*

Recall that the condition $\mathcal{H}^{BvN} \subset \mathcal{H}$ is natural in bilateral matching settings and is imposed in all applications in this paper.

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 $\mathcal{H} \supset \mathcal{H}^{BvN}$ is not a bihierarchy.

Remark 1. *In light of Examples 1 and 2, one might wonder whether an absence of odd cycles is sufficient for decomposability. This turns out to be false. 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. One can check that the random assignment P above cannot be feasibly decomposed.*

---

structure (defined in Section 7) forms a bihierarchy, a condition that Theorem 5 shows is sufficient for decomposability.

In Section 7 we consider a more general environment than the bilateral setting, and present a further generalization of the Rounding Theorem to that environment. We also use Lemma 1 to show that the decomposition result cannot be extended to any multilateral matching of more than two kinds of agents or to roommate matching.

3.3. **Examples of Bihierarchy.** As discussed above, $\mathcal{H}^{BvN}$ is a bihierarchy. This section discusses more examples.

3.3.1. *Flexible Capacity.* Consider a school choice problem in which the school authority wishes to run several education programs within one building. Several capacity constraints can be represented using a hierarchy $\mathcal{H}_O$ containing sets of the form $S = N \times O'$. The ceiling $\overline{q}_S$ then describes the total capacity that can be allocated within O', which can apply to a program or a set of programs. Notice that the hierarchical structure $\mathcal{H}_O$ allows for the nested constraints on program sizes.

3.3.2. *Group-specific Quotas.* Affirmative action policies are sometimes implemented as quotas on students fitting specific gender, racial, or economic profiles.[6] 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 set to begin in 2010 in Seoul, Korea, limits the percentage of seats allocated to the applicants from outside the district.[7] and a number of school choice programs in Japan have similar quotas based on residential areas as well.

Such quotas can be incorporated by $\mathcal{H}_O$ containing sets of the form $N' \times \{a\}$ for $a \in O$ and $N' \subsetneq N$. 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 a hierarchical structure of $\mathcal{H}_O$ as long as they do not overlap with each other. Moreover, a nested series of constraints can be accommodated. 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 accomodate both flexible-capacity constraints and group-specific quota constraints within the same hierarchy $\mathcal{H}_O$. Flexible-capacity constraints are defined on multiple columns of a random assignment matrix P, whereas group-specific quota

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

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

constraints are defined on subsets of single columns of P. Any subset of a single column will be a subset of or disjoint from any set of multiple columns.

3.3.3. *Course Allocation.* The course allocation problem begins with a set of students and courses. 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 such that $\mathcal{H}_N \supset \mathcal{H}_N^{BvN}$. 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 finance courses and marketing courses, if $\mathcal{H}_N$ 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 3.3.1-3.3.2 can also be incorporated into the course allocation problem without jeopardizing the bihierarchical structure. These constraints pertain to $\mathcal{H}_O$, while the preference and scheduling constraints described above pertain to $\mathcal{H}_N$. So long as $\mathcal{H}_N$ and $\mathcal{H}_O$ are each hierarchies, $\mathcal{H} = \mathcal{H}_N \cup \mathcal{H}_O$ is a bihierarchy.

3.3.4. *Interleague Play.* 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, but play across the AL and NL, called interleague play, was introduced in 1997.[8] 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. There are some matchup

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

constraints: Each team in N has a geographic rival in O, and they must play twice. For fairness reasons, 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 form a bihierarchy.

4. Application: Single-unit assignment

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 to allocate objects in such a setting is the **random priority** mechanism. *In this mechanism, every agent reports preference rankings of the objects. The designer then orders the agents at random, with each ordered list chosen with equal probability. Given a realized list, the first agent in the list 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 deterministic 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.[9] 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 resulting random assignment is

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

[9]The null object is allowed in many subsequent works although it is not present in the original setting of Bogomolnaia and Moulin.

This assignment entails an unambiguous efficiency loss. For instance, every agent prefers an alternative random assignment,

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

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

The probabilistic serial (PS) mechanism, introduced by Bogomolnaia and Moulin (2001) in $\mathcal{E}^{BvN}$, 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 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 PS assignment is defined as the resulting probability shares.* In the current 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 PS assignment is given by P'. In particular, the probabilistic serial mechanism eliminates the inefficiency that was present under RP. More generally, Bogomolnaia and Moulin (2001) show that the probabilistic serial random assignment is ordinally efficient with respect to any reported preferences.[10]

We generalize the PS mechanism to accomodate constraints such as those discussed in Sections 3.3.1-3.3.2. Specifically, let $\mathcal{H} \supset \mathcal{H}^{BvN}$ be a bihierarchy in which $\underline{q}_S = 0$ for

[10]The contribution of Bogomolnaia and Moulin has led to much subsequent work on random assignment mechanisms for single-unit assignment problems. The PS 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 PS mechanism. Ordinal efficiency is characterized by Abdulkadiroğlu and Sönmez (2003a), McLennan (2002) and Manea (2006). Behavior of the random priority and PS 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 PS mechanism is group strategy-proof and first-order stochastically dominates the random priority mechanism, and Bogomolnaia and Moulin (2002) give two characterizations of the PS mechanism.

$S \in \mathcal{H}_O$, and $\mathcal{H}_N = \mathcal{H}_N^{BvN}$. Consider the following generalized eating algorithm. *Time runs continuously from 0 to 1. At every point in time, each agent "eats" her favorite object with speed one among those that are "available" at that instance, and the PS assignment is defined as the probability shares eaten by each agent at time* 1. In order to obtain a feasible random 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 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 the quota $\overline{q}_S$ for every constraint set $S \ni (i, a)$.*

By construction, the generalized PS mechanism produces a random assignment that satisfies the quotas defined over all constraint sets. Since the constraint structure form a bihierarchy, the Rounding Theorem ensures that the random assignment can be implemented by a lottery over deterministic assignments each satisfying the constraints. In this sense, our generalization of the Birkhoff-von Neumann theorem enables the PS mechanism to be applicable to a broader set of real-life market design environments than has previously been possible.[11]

We note one limitation of our generalization, which is that the algorithm works only 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. Often, administrative goals can 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.

An important remaining question is whether our adaptation of PS will continue to have the original's desirable properties of efficiency, fairness and incentives. It turns out that these questions are answered in the affirmative. Yet, full treatment of these questions will take us far afield, so we postpone them to our companion paper (Budish, Che, Kojima, and Milgrom, 2009).

[11]For instance, the upcoming Korean school choice program involves quotas at each school for applicants from outside districts, and unlike in some other school-choice contexts the schools do not have preferences over the students (e.g. based on test scores or sibling priority). Hence the problem is essentially random assignment with group specific quotas. Currently, the algorithm appears to be that of the so-called Boston mechanism (see Abdulkadiroğlu and Sönmez (2003b) for a description). One could use the modified PS instead.

5. Application: Multi-Unit Assignment

This section considers multi-unit resource allocation problems in which monetary transfers are prohibited. Examples include the assignment of course schedules to students, 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. For concreteness consider the course allocation problem, where the course administrator in a school assigns seats in courses. Sönmez and Ünver (2008) point out that mechanisms used in many business schools suffer from incentive issues and inefficiency. Other schools use generalizations of random priority mechanisms. For example, Harvard Business School uses a mechanism similar to those used in some professional sports drafts. The mechanism is manipulable unlike the random priority mechanism for single-unit assignment and, like the random priority mechanism, suffers from efficiency loss (Budish and Cantillon, 2009). Designing good course allocation mechanisms remains a challenging market design problem.[12]

An alternative approach is to find a desirable random assignment directly. For the single-unit assignment problem, discussed above, the PS mechanism adopts this approach. The same approach is taken by Hylland and Zeckhauser (1979). For the single-unit assignment environment, they propose a mechanism based on the idea of Competitive Equilibrium from Equal Incomes (CEEI). In that mechanism, agents are given an equal budget of artificial currency. Based on agents' reported preferences, the mechanism computes a competitive equilibrium where commodities are probabilities of objects and each agent is allocated the random assignment corresponding to her consumption bundle in equilibrium. The first welfare theorem guarantees that a random assignment constructed in this way is (*ex ante*) Pareto efficient. (The use of equal incomes ensures that the random assignment is envy free.) Since this mechanism produces a random assignment directly, the Birkhoff-von Neumann theorem is again crucial for implementing the mechanism. Adapting their idea, we can consider a mechanism based on competitive equilibrium in multi-unit assignment environments (see Pratt and Zeckhauser (1990), Pratt (2007)). As with Hylland and Zeckhauser's mechanism, the resulting assignment promises to have desirable ex-ante efficiency and fairness properties. This simple multi-unit extension of the Hylland-Zeckhauser model conforms to a bihierarchy, so the Rounding Theorem ensures

[12]Axiomatic analysis on this problem has also obtained negative conclusions. 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.

that we can implement the random assignments. And, we can accommodate additional constraints within the bihierarchy framework.[13]

In multi-unit assignment, a given random assignment may be implementable in several different ways, and the choice among them may be important. For example, suppose that two agents are to divide n objects (where n is even), that preferences are additively separable, and that agent's ordinal rankings of the items are the same. Suppose the "fair" random assignment specifies that each agent receive half of each object. One way to implement this is to chose $n/2$ objects randomly to assign to one agent and give the remaining $n/2$ to the other agent. This method, however, could entail a highly "unfair" outcome *ex post*, in which one agent gets the $n/2$ best objects and the other gets the $n/2$ worst ones.

Based on the Rounding Theorem, we provide a method to reduce the variation in utility outcomes resulting from randomization. Formally, consider an input $\langle N, O, P, \mathcal{H} \rangle$ where $\mathcal{H}$ is a bihierarchy partitioned into hierarchies $\mathcal{H}_O$ and $\mathcal{H}_N = \mathcal{H}_N^{BvN}$. Assume that P satisfies $\sum_a P_{ia} \in \mathbb{Z}$ for each $i \in N$. This assumption is without loss of generality because any random assignment with non-integral row sums is equivalent to a random assignment with an additional column representing a null object, the sole purpose of which is to ensure that rows sum to integer amounts.

Theorem 3. (Utility Guarantee) *Suppose that there is a set of values $(v_{ia})_{(i,a) \in N \times O}$ such that, for each i, agent i's expected utility from a random assignment P is $\sum_{a \in O} P_{ia} v_{ia}$. Then, for any P, there exists a decomposition of P that satisfies all of the conditions of the Rounding Theorem, and also:*

$$\sum_a P'_{ia} v_{ia} - \sum_a P''_{ia} v_{ia} \in [-\Delta_i, \Delta_i], \tag{5.1}$$

$$\sum_a P'_{ia} v_{ia} \in \left[\sum_a P_{ia} v_{ia} - \Delta_i, \sum_a P_{ia} v_{ia} + \Delta_i \right], \tag{5.2}$$

for each i and each P' and P'' in the convex combination, where $\Delta_i := \max\{v_{ia} - v_{ib} | a, b \in O, P_{ia}, P_{ib} \notin \mathbb{Z}\}$.

[13] Budish (2009) proposes a mechanism that adapts CEEI to multi-unit assignment in a different way. Whereas here agents consume their most preferred affordable random assignment of objects, in Budish's mechanism agents consume their most preferred affordable sure bundle of objects. Randomness enters through the budgets, which are approximately equal rather than exactly equal. The tradeoffs between these two approaches are described in Section 7.2 of Budish (2009).

A proof sketch can be given based on the Rounding Theorem. The idea is to supplement the actual constraints of the problem with a set of "utility proximity" constraints, as follows. If there are n objects, for each agent we create n additional constraints. The j^{th} constraint set of agent i, S_{ij}, consists of his $1^{\text{st}}, 2^{\text{nd}}, \ldots, j^{\text{th}}$ most preferred objects; its floor and ceiling constraints are $\left\lfloor \sum_{a \in S_{ij}} P_{ia} \right\rfloor$ and $\left\lceil \sum_{a \in S_{ij}} P_{ia} \right\rceil$, respectively. The resulting constraint structure is still a bihierarchy after this addition, so the Rounding Theorem guarantees that the random assignment can be implemented with all of the constraints satisfied. Satisfying the constraints on all of the artificial "upper contour" sets means that in each realized assignment, each agent receives her j most preferred objects, for each j, with approximately the same probability as in the original random assignment. Our method thus ensures that each realized assignment inherits roughly the same fairness properties as the original random assignment; more precisely, the approximation error is at most the utility difference between the most valuable and the least valuable objects. The proposed method is probably useful when that utility difference is small compared to the total utility of the agent's bundle of goods.

5.1. **Further Applications of Theorem 3.** Theorem 3 can be used to approach certain solved problems in integer programming, leading to slightly improved versions of well-known results.

5.1.1. *The Maximin Approach to Fair Division.* Suppose that there are a number of indivisible objects O to be allocated to agents N. Agents' utilities are additive subject to a fixed quota on the number of items assigned (the quota can be infinite), so utility of agent i from random assignment P is $\sum_{a \in O} v_{ia} P_{ia}$ if her quota constraint is satisfied. The problem is to maximize the utility of the worst-off agent. This is sometimes called the Santa Claus problem: Santa Claus wants to give presents to children in such a way that the least fortunate child is as happy as possible given the fixed set of available presents. Formally, consider the social planner's problem:

$$\text{maximize } \omega \text{ subject to} \tag{5.3}$$

$$\begin{aligned}
P_{ia} &\in \mathbb{N} && \text{for all } i \in N, a \in O, \\
P_S &\leq \overline{q}_S, && \text{for all } S \in \mathcal{H}_O, \\
\sum_{a \in O} P_{ia} &\leq \overline{q}_{\{i\} \times O}, && \text{for all } i \in N, \\
\omega &\leq \sum_{a \in O} P_{ia} v_{ia} && \text{for all } i \in N,
\end{aligned}$$

where $\mathbb{N} = \{0, 1, 2, \dots\}$ is the set of natural numbers (nonnegative integers). This problem is known to be computationally difficult. Thus in practice, the social planner may need to use a mechanism that is easier to implement. On the other hand, she wants to attain the objective at least approximately.

To attain these conflicting goals, consider the following two-stage algorithm. In the first stage, solve the **associated linear programming problem**, that is, a problem identical to (5.3) except that the constraint $P_{ia} \in \mathbb{N}$ is replaced by $P_{ia} \in [0, \infty)$. Since the problem relaxes the integrality of the first constraint, the solution may be infeasible. On the other hand the optimal solution of this problem is easy to compute since it is a simple linear programming problem. In the second stage, given the optimal solution of the associated linear programming problem, round the solution into an integer-valued solution, making the assignment a feasible solution in problem (5.3). The cost of doing so is that the social welfare typically decreases when the social planner modifies the optimal fractional solution into an integral one. However, the following claim guarantees that the loss of efficiency can be bounded. Let P^* and ω^* be a solution and the optimal value of the linear programming problem associated with (5.3).

Corollary 2. *There exists a solution of the problem (5.3) with value* $\omega' \geq \omega^* - \max_{i \in N} \overline{v}_i$*, where* $\overline{v}_i = \max\{v_{ia} | a \in O, P^*_{ia} \notin \mathbb{N}\}$*. In particular,* $\omega' \geq \omega^{**} - \max_{i \in N} \overline{v}_i$ *where* w^{**} *is the optimal value of the problem (5.3).*

The proof is a direct application of Theorem 3. The theorem gives a bound on the utility loss for each agent in an integer solution associated with the optimal fractional solution of the linear programming problem.

Corollary 2 generalizes Bezáková and Dani (2005), who proposed a similar two-stage algorithm for $\mathcal{E}^{BvN}$. While Corollary 2 is a small extension, its derivation from Theorem 3, which in turn is a direct consequence of the Rounding Theorem, highlights a common foundation among apparently dissimilar results.

5.1.2. *Scheduling Jobs on Parallel Machines: Minimize Makespan Problem.* Our approach can also be applied to the so-called "minimize makespan problem" studied widely in computer science. Consider the problem, slightly generalizing Lenstra, Shmoys, and Tardos (1990), in which a set N of parallel machines must be assigned to perform a set O of independent jobs. The jobs are indivisible, that is, each job requires one machine in its entirety (or equivalently, it is prohibitively costly to process part of a job in one machine and process remaining parts in others). The processing of job a on machine i takes time c_{ia}. The machines are parallel and jobs are independent, that is, more than one machine can process jobs simultaneously and any job can be processed irrespective of whether other jobs are already completed. The **makespan** of the assignment of jobs to machines is the time needed to finish all jobs. The objective is to find a schedule that minimizes the makespan.

Let $J_i(t)$ denote the set of jobs that require at most time t when processed by machine i, and let $M_a(t)$ denote the set of machines that can process job a in no more than time t. Consider the relaxed problem in which random assignments are allowed, and let P be a fractional assignment of jobs to machines where each machine i finishes processing jobs by deadline d_i. We will show the following slight generalization of the rounding theorem of Lenstra, Shmoys, and Tardos (1990).

Corollary 3 (Theorem 1 of Lenstra, Shmoys, and Tardos (1990))**.** *Let* $c = (c_{ia})_{(i,a)\in N\times O} \in \mathbb{R}_+^{|N|\times|O|}$, $d = (d_i)_{i\in N} \in \mathbb{R}_+^{|N|}$ *and* $t \in \mathbb{R}_+$ *be given. If there is a feasible solution* P *to the (in)equalities,*

$$\sum_{i\in M_a(t)} P_{ia} = 1, \qquad \text{for } a \in O,$$

$$\sum_{a\in J_i(t)} P_{ia}c_{ia} \leq d_i, \qquad \text{for } i \in N,$$

$$P_{ia} \geq 0, \qquad \text{for } a \in J_i(t), i \in N,$$

then there is an integer solution P' *to the following set of conditions,*

$$\sum_{i\in M_a(t)} P'_{ia} = 1, \qquad \text{for } a \in O, \tag{5.4}$$

$$\sum_{a\in J_i(t)} P'_{ia}c_{ia} \leq d_i + t, \qquad \text{for } i \in N,$$

$$P'_{ia} \in \{0,1\}, \qquad \text{for } a \in J_i(t), i \in N.$$

Proof. By Theorem 3, there exists P' that is integer-valued and satisfies (5.4) and

$$\sum_a P'_{ia}c_{ia} \leq \sum_a P_{ia}c_{ia} + \max_{a,b\in O: P_{ia},P_{ib}\in(0,1)} (c_{ia} - c_{ib}).$$

Since $\sum_{i\in M_a(t)} P_{ia} = 1$, $P_{ia} > 0$ means $i \in M_a(t)$, which in turn implies that $t \geq c_{ia} \geq c_{ia} - c_{ib}$ for any $b \in O$. This completes the proof. □

The result implies that there exists a feasible integer solution whose makespan is within time t of the optimal (infeasible) fractional solution, where t is the time of the single slowest job processed in the fractional solution. Since the optimal fractional solution is weakly better than the optimal feasible integer solution, we have a method that finds an integer solution that is "close" to the true optimum. Some generalizations of the minimize makespan problem, such as Theorem 2.1 of Shmoys and Tardos (1993), are also corollaries of Theorem 3, via a logic similar to the proof of Corollary 3.

6. Application: Two-Sided Matching

Our approach can be applied to the two-sided matching environment. In this section, both N and O are sets of agents. We allow for many-to-many matching, that is, each agent in N can be matched with multiple agents in O, and vice versa.

Theorem 4. *Consider a problem where the constraint structure is $\mathcal{H}^{BvN}$. Suppose that there are sets of values $(v_{ia})_{(i,a)\in N\times O}$ and $(w_{ia})_{(i,a)\in N\times O}$ such that, for each agent $i \in N$ (respectively agent $a \in O$), her expected utility from any random assignment P is $\sum_{a\in O} P_{ia}v_{ia}$ (respectively $\sum_{i\in N} P_{ia}w_{ia}$). Then, for any P, there exists a decomposition of P that satisfies the conditions of the Rounding Theorem, and also:*

$$\sum_a P'_{ia}v_{ia} - \sum_a P''_{ia}v_{ia} \in [-\Delta_i, \Delta_i]$$

$$\sum_a P'_{ia}v_{ia} \in \left[\sum_a P_{ia}v_{ia} - \Delta_i, \sum_a P_{ia}v_{ia} + \Delta_i\right],$$

$$\sum_i P'_{ia}w_{ia} - \sum_i P''_{ia}w_{ia} \in [-\Delta_a, \Delta_a]$$

$$\sum_i P'_{ia}w_{ia} \in \left[\sum_i P_{ia}w_{ia} - \Delta_a, \sum_i P_{ia}w_{ia} + \Delta_a\right],$$

for each i, a and each P' and P'' being part of the convex decomposition, where $\Delta_i = \max\{v_{ia} - v_{ib} | a, b \in O, P_{ia}, P_{ib} \notin \mathbb{N}\}$ and $\Delta_a = \max\{w_{ia} - w_{ja} | i, j \in N, P_{ia}, P_{ja} \notin \mathbb{N}\}$.

Proof. The proof is a straightforward adaptation of the proof of Theorem 3 and hence is omitted. □

Let us suggest one possible application. There are two leagues of sports teams N and O, say the American League and National League in professional baseball, and the planner wants to schedule interleague play. 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 planner could first give a uniform probability for each match: That will give one specific random assignment in which any pair of teams in the same league is treated equally. Then, using Theorem 4, the planner finds a deterministic 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.

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.

7. Beyond Bilateral Assignment

Throughout the paper we have focused on a random assignment of objects (or agents) to agents. However, some of our results can be extended beyond pairwise assignment, as described below.

Let X be a finite set and $\mathcal{H}$ be a collection of subsets of X. We call the pair $\mathcal{X} = (X, \mathcal{H})$ a **hypergraph**. A (generalized) random assignment is a vector $P = [P_x]$ where $P_x \in (-\infty, \infty)$ for all $x \in X$. For each $S \in \mathcal{H}$, $P_S = \sum_{x \in S} P_x$. A deterministic assignment is a random assignment each of whose entries is an integer. As before, the constraint structure $\mathcal{H}$ is decomposable if, for each $(\underline{q}_S, \overline{q}_S)_{S\in\mathcal{H}}$ and P with $\underline{q}_S \le P_S \le \overline{q}_S$ for all $S \in \mathcal{H}$, there exist $\lambda^1, \dots, \lambda^K$ and $P^1, \dots, P^K$ such that

(1) $P = \sum_{k=1}^{K} \lambda^k P^k$,
(2) $\lambda^k > 0, k = 1, \dots, K$, and $\sum_{k=1}^{K} \lambda^k = 1$,
(3) P_x^k is an integer for each x,
(4) $\underline{q}_S \le P_S^k \le \overline{q}_S$ for each $k = 1, \dots, K$ and $S \in \mathcal{H}$.

Equivalently, $\mathcal{H}$ is decomposable if and only if for every random assignment P there exist $\lambda^1, \dots, \lambda^K$ and $P^1, \dots, P^K$ such that

(1) $P = \sum_{k=1}^{K} \lambda^k P^k$,
(2) $\lambda^k > 0, k = 1, \dots, K$, and $\sum_{k=1}^{K} \lambda^k = 1$,
(3) $P_S^k \in \{\lfloor P_S \rfloor, \lceil P_S \rceil\}$ for all $k \in \{1, \dots, K\}$ and $S \in \mathcal{H}$.

We say that $\mathcal{X}$ forms a bihierarchy if there exist $\mathcal{H}_1$ and $\mathcal{H}_2$ such that $\mathcal{H}_1 \cup \mathcal{H}_2 = \mathcal{H}$, $\mathcal{H}_1 \cap \mathcal{H}_2 = \emptyset$, and $\mathcal{H}_i$ is a hierarchy for each $i = \{1, 2\}$: if $S, S' \in \mathcal{H}_i$, then $S \cap S' = \emptyset$ or $S \subset S'$ or $S' \subset S$.

It is useful to define the dual of a hypergraph. Given a hypergraph $\mathcal{X} = (X, \mathcal{H})$, its dual is $\mathcal{X}^T = (\mathcal{H}, X)$. A bihierarchy can be defined for its dual. To this end, for each $x \in X$, let $\mathcal{S}(x) := \{S \in \mathcal{H} | x \in S\}$ be the collection of sets in $\mathcal{H}$ each containing x. We say **the dual of $\mathcal{X}$ forms a bihierarchy** if there are X_1 and X_2 such that $X_1 \cup X_2 = X$,

$X_1 \cap X_2 = \emptyset$ and X_i, $i = 1, 2$, is a **dual hierarchy**: if $x, x' \in X_i$, then $\mathcal{S}(x) \cap \mathcal{S}(x') = \emptyset$ or $\mathcal{S}(x) \subset \mathcal{S}(x')$ or $\mathcal{S}(x') \subset \mathcal{S}(x)$.

A hypergraph $\mathcal{X} = (X, \mathcal{H})$ can be represented by an incidence matrix $A = [a_{xS}]$ such that $a_{xS} = 1_{\{x \in S\}}$. The incidence matrix of the dual $\mathcal{X}^T$ is A^T, the transpose of A.

Theorem 5. *A hypergraph is decomposable if either it forms a bihiearchy or its dual forms a bihierarchy.*

The proof of Theorem 5 is in Appendix A. A key step in the proof, due to Edmonds (1970), is to show that the incidence matrix A of a bihierarchical constraint structure satisfies a condition called total unimodularity. This in turn enables one to show that the set of random assignments satisfying a bihierarchical constraint structure has integral extreme points, which enables decomposition because the set is convex.[14]

We present two examples of bihierarchies whose dual satisfies the bihierarchy condition.

Example 3. *Consider* $X = \{a, b, c, d, e, f\}$, *and*

$$\mathcal{H} = \{\{a, d\}, \{a, e\}, \{a, f\}, \{b, d\}, \{b, e\}, \{b, f\}, \{c, d\}, \{c, e\}, \{c, f\}\}.$$

The hypergraph $\mathcal{X} = (X, \mathcal{H})$ *is in fact a bipartite graph in this case. Even though it does not form a bihierarchy, its dual forms a bihierarchy. Its dual is an assignment between three agents and three objects, with only row and column constraints.*

Example 4. *Consider* $X = \{a, b, c, d, e, f, \alpha, \beta, \delta, \epsilon\}$, *and*

$$\mathcal{H} = \{\{a, d, \alpha, \delta\}, \{a, e, \alpha, \epsilon\}, \{a, f\}, \{b, d, \beta, \delta\}, \{b, e, \beta, \epsilon\}, \{b, f\}, \{c, d\}, \{c, e\}, \{c, f\}\}.$$

The hypergraph $\mathcal{X} = (X, \mathcal{H})$ *again does not form a bihierarchy, but its dual forms a bihierarchy. Its dual is the 3 by 3 matching, with row and column constraints, and two subrow and two subcolumn constraints.*

We note that Lemma 1 clearly holds in this general environment (with an identical proof), providing a necessary condition for decomposability. We can apply this lemma to show the difficulty one faces in implementing random assignments in multilateral matching and roommate matching.

[14]We thank Tomomi Matsui and Akihisa Tamura for informing us of the connection with the mathematics literature on matroids and Edmonds (1970).

7.1. **Multilateral Matching.** Thus far, we have focused on bilateral matching in which agents on one side are assigned to objects (or agents) on the other side. As noted, many important market design problems fall into the bilateral matching 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 trilateral, 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 trilateral matching problem in which we introduce another finite set M of say agents, in addition to N and O. A matching then consists of a triple $(i, a, m) \in N \times O \times M$, and a random assignment is defined by a profile $[P_{(i,a,m)}]_{(i,a,m)\in N\times O\times M}$ that assigns a real number to each triple (i, a, m). Constraints on the random assignment can be described as before via the constraint structure, i.e., the sets of (i, a, m)'s whose entries are subject to a ceiling or a floor. That is, the constraint structure $\mathcal{H} \subset 2^{N\times O\times M}$ is a collection of subsets of $N \times O \times M$. 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 M must be assigned to some pair in the other two sides (which may include a null object or null agent, by including such entities to the sets). Hence, it is natural to assume that $\mathcal{H}$ contains the sets $\overline{\mathcal{H}}^{BvN} := \{\{i\} \times O \times M | i \in N\} \cup \{N \times \{a\} \times M | a \in O\} \cup \{N \times O \times \{m\} | m \in M\}$.

Notice that the problem reduces to that of bilateral matching if the cardinality of N or O or M is one. It turns out that, except for such cases, no analogue of the Birkhoff-von Neumann theorem holds with a trilateral matching.

Theorem 6. (Impossibility with Trilateral Matching) *In trilateral matching with $N \times O \times M$ where $|N|, |O|, |M| \geq 2$, any $\mathcal{H} \supset \overline{\mathcal{H}}^{BvN}$ is not decomposable.*

Proof. We prove the result by showing that any $\mathcal{H} \supset \overline{\mathcal{H}}^{BvN}$ contains an odd cycle. By Lemma 1, this is sufficient for failure of decomposability. (Even though the proof of Lemma 1 formally deals with the bilateral matching setup, its proof does not depend on it.)

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

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

Remark 2. (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 at what contract terms. At first glance, introducing contract terms may appear to transform the environment into a trilateral matching setting. This is in fact not the case. If we let M denote the set of possible contract terms, there is no sense in which the constraint structure contains sets of the form $N \times O \times \{m\}$. In words, there is no reason that each contract term should be chosen by some worker-firm pair. Rather, the matching with contracts can be subsumed into our bilateral matching setup by redefining the object set as $O' := O \times M$.*

7.2. **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 $X_N := \{\{i,j\}|i,j \in N\}$ 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. Let $\mathcal{H}_N$ be a collection of subsets of X_N. A hypergraph $\mathcal{X}_N = (X_N, \mathcal{H}_N)$ allows us to capture the basic environment arising in roommate matching, and we can describe a (generalized) random assignment as a vector $P = [P_x]_{x \in X_N}$ where $P_x \in [0,1]$ for all $x \in X_N$. We assume that each i must be assigned to some agent (possibly himself), so $\mathcal{H}_N$ must contain set $S_i := \{\{i,j\}|j \in N\}$ for each $i \in N$. We call a hypergraph $\mathcal{X}_N$ satisfying this property a **canonical roommate matching problem with N agents**.

Notice that the problem reduces to that of bilateral matching if $|N| \leq 2$, implying that any canonical roommate matching problem with such N is decomposable. The next result shows that these are the only cases for which decomposability holds.

Theorem 7. (IMPOSSIBILITY WITH ROOMMATE MATCHING) *A canonical roommate matching problem with N agents, $|N| \geq 3$, is not decomposable.*

Proof. We prove the result by showing that $\mathcal{H}_N$ in the canonical roommate matching contains an odd cycle. 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. □

8. Polynomial-Time Algorithm for Implementing Random Assignments

Theorem 5 demonstrates implementability of random assignments without revealing the method. For practical purposes, knowing simply that a random assignment is implementable is not sufficient; implementation must be computable or sufficiently fast. Here we provide an algorithm that implements random assignments in polynomial time.[15] At each step of the algorithm, which is described more fully in Appendix B, a given feasible random assignment P is decomposed into a convex combination $\gamma P' + (1 - \gamma)P''$ of two feasible random assignments, each of which has at least one more integer-valued agent-object pair (or integer-valued constraint set). Then, a random number is generated and with probability γ the algorithm continues by similarly decomposing P', while with probability $1 - \gamma$ the algorithm continues by decomposing P''. The algorithm stops when it reaches an integer assignment.

We explain the decomposition method in the context of the following example. Consider a hypergraph $\mathcal{X} = (X, \mathcal{H})$ where $X = \{x_1, x_2, x_3, x_4\}$, $\mathcal{H} = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, S_1, S_2\}$ and $S_1 := \{x_2, x_3\}$ and $S_2 := \{x_3, x_4\}$. Observe that $\mathcal{H}$ is a bihierarchy consisting of two hierarchies, $\mathcal{H}_1 = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, S_1\}$ and $\mathcal{H}_2 = \{S_2\}$, where $S_1 := \{x_2, x_3\}$ and $S_2 := \{x_3, x_4\}$. Suppose we wish to implement a random assignment P with $P_{\{x_1\}} = 0.3, P_{\{x_2\}} = 0.7, P_{\{x_3\}} = 0.3$ and $P_{\{x_4\}} = 0.7$. We represent the given random assignment P as a network flow. The particular way in which the flow network is constructed is crucial for the algorithm, and we formally describe its construction in Appendix B. Here, we present an informal discussion based on our example. See Figure 1.

[15]We thank Tomomi Matsui and Akihisa Tamura for suggesting this algorithm. An earlier draft of this paper included an alternative algorithm generalizing the stepping-stones algorithm described by Hylland and Zeckhauser (1979). We have made our old algorithm available in a Web Appendix.

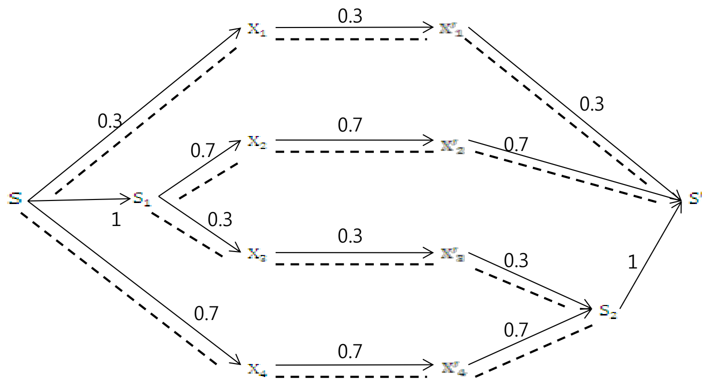

Figure 1: A network flow representation of the example P.

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 X). 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 X, 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 random assignment for the corresponding set. For instance, the flow from x_2 to x_2' is the random assignment $P_{\{x_2\}} = 0.7$ for set x_2, and likewise the flow from x_3 to x_3' is $P_{\{x_3\}} = 0.3$. The flow from s to S_1 represents the random assignment $P_{S_1} = 1$ for set S_1. Naturally, the latter flow must be the sum of the two former flows. More generally, the additive structure of the random assignment is translated into the "law of conservation": *the flow reaching each (intermediate) vertex 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 (x_2, x_2'), we find another edge with a fractional flow that is adjacent to x_2'. Such an edge, (x_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 $x_2 - x_2' - s' - x_1' - x_1 - s - x_4 - x_4' - S_2 - x_3' - x_3 - S_1 - x_2$. 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 random assignment. The resulting network flow then gives rise to a random assignment P' where $P'_{\{x_1\}} = 0, P'_{\{x_2\}} = 1, P'_{\{x_3\}} = 0$, and $P'_{\{x_4\}} = 1$. 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 random assignment P'' where $P''_{\{x_1\}} = 1, P''_{\{x_2\}} = 0, P''_{\{x_3\}} = 1$, and $P''_{\{x_4\}} = 0$. We can now decompose P into these two matrices, i.e., $P = 0.7P' + 0.3P''$.

The random algorithm then selects P' with probability 0.7 and P'' with probability 0.3. Since in this particular example both P' and P'' are integer valued, there is no need to re-iterate 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 $|\mathcal{H}|$.

9. Conclusion

We generalize the Birkhoff-von Neumann theorem by applying it to general real matrices, rather than just bistochastic matrices, and allowing a much larger class of constraints, rather than just row and column constraints. We show that if the constraints specify integer floors and ceilings for sums over sets of elements in the matrix, and if the constraint sets form a bihierarchy, then the matrix is a convex combination of integer matrices each of which also satisfies the constraints. This convex combination is a "decomposition" of the expected allocation described by the originally given matrix. We also establish a converse: if the constraint sets described above include row and column constraints but do not form a bihierarchy, then there are matrices and constraint bounds such that the matrix is not a convex combination of feasible integer matrices. We show that these results are usefully applicable to a range of matching problems, including (i) single-unit assignment, (ii) multi-unit assignment, (iii) fair division, (iv) job scheduling, and (v) two-sided matching, and that the results enable useful generalizations of Bogomolnaia and Moulin's (2001) probabilistic serial mechanism and Hylland and Zeckhauser's (1979) pseudo-market mechanism. We also introduce a polynomial algorithm that implements the random allocation, selecting each pure outcome with the appropriate probability.

We investigate other kinds of constraints as well, mostly with negative results. There is no similar decomposability of expected allocations for matching with three sides or more, nor for roommate problems.

For the future, one may expect to find closer connections to similar decomposability problems arising in optimal mechanism theory. But the goal of research in market design is to facilitate applications, and we are most hopeful that the examples of implementable random assignments described here herald still further applications to come.

Appendix A. Proofs of Theorems 1 and 5

Since Theorem 1 is a special case of Theorem 5, we prove the latter.

A matrix is **totally unimodular** if the determinant of every square submatrix is 0, -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 5 proceeds in two steps. First we show that if either a hypergraph or its dual forms a bihierarchy, then the incidence matrix of the hypergraph is totally unimodular. Second we apply Lemma 2 to show that the hypergraph is decomposable.

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 own proof for completeness below. The case in which the dual of a hypergraph forms a bihierarchy is not in Edmonds (1970).

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 5. Suppose first $\mathcal{X}$ 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 X 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 X 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 $x \in X$. If x 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 x either equals or exceeds by one the number of red sets in $E \cap \mathcal{H}_1$ containing x. By the same reasoning, if x 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 x is less by one than or equal to the number of red sets in $E \cap \mathcal{H}_2$ containing x. In sum, the number of blue sets in E containing x differs at most by one from the number of red sets in E containing x. Thus A is totally unimodular by Lemma 3.

Choose an arbitrary random assignment P and consider the set

$$\{P' | \lfloor P_S \rfloor \leq P'_S \leq \lceil P_S \rceil, \forall S \in \mathcal{H}\}. \tag{A.1}$$

By Lemma 2, every vertex of the set (A.1) is integer valued. Since (A.1) is a convex polyhedron, any point of it (including P) can be written as a convex combination of its vertices. Since we chose P arbitrarily, the hypergraph $\mathcal{X}$ is decomposable.

We next consider the case where the dual of $\mathcal{X}$ forms a bihierarchy. To this end, consider a hypergraph $\mathcal{X}^* = (X^*, \mathcal{H}^*)$ such that $X^* = \mathcal{H}$ and $\mathcal{H}^* = X$. That is, X^* is a finite ground set whose elements share the same labels as the elements in $\mathcal{H}$, and $\mathcal{H}^*$ is a collection of subsets of $\mathcal{H}^*$ that have the same labels as X. Assume that $S \in X^*$ is an element of $x \in \mathcal{H}^*$ in $\mathcal{X}^*$ if and only if x is an element of S in $\mathcal{X}$. The fact that the dual of $\mathcal{X}$ forms a bihierarchy means that (the primal of) $\mathcal{X}^*$ forms a bihierarchy. The argument made above then implies that the incidence matrix A^* associated with $\mathcal{X}^*$ is totally unimodular. Since this matrix coincides with the incidence matrix of the dual of $\mathcal{X}$, $A^* = A^T$. Since a transpose of a totally unimodular matrix is totally unimodular in general by definition, it follows that the incidence matrix A of $\mathcal{X}$ must be also totally unimodular. Hence, by an analogous argument to that above, the hypergraph $\mathcal{X}$ is decomposable. □

Appendix B. Algorithm for Implementing Random Assignments

This section provides a computable algorithm, which also serves as a constructive proof for Theorem 5 for the bihierarchy case (and hence Theorem 1).

Let $(X, \mathcal{H})$ be a hypergraph 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 $P = [P_x]$ be a random 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_x and v'_x for each element $x \in X$, and v_S for each $S \in \mathcal{H} \setminus [(\bigcup_{x \in X}\{x\}) \cup (N \times O)]$. We place (directed) edges according to the following rule.[16]

(1) For each $x \in X$, an edge $e = (v_x, v'_x)$ is placed from v_x to v'_x.
(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 $S' \subset S'' \subset S$.[17]
(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 $S \subset S'$.

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

[17]For the purpose of placing edges, we regard v_x as a vertex corresonding to a singleton set $\{x\} \in H_1$, and v'_x as a vertex corresonding to a singleton set $\{x\} \in H_2$.

(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 $e = (v_x, v'_x)$, we associate flow $P_e = P_x$. For each e that is not of the form (v_x, v'_x) for some $x \in X$, the flow P_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 P with respect to $\mathcal{H}$:

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

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

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

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

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

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

□ **Decomposition Algorithm**

(1) **Cycle-Finding Procedure**
   - (a) **Step 0:** Since deg$[P(\mathcal{H})] < |\mathcal{H}|$ by assumption, there exists an edge $e_1 = (v_1, v'_1)$ such that its associated flow P_{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.
     - (i) If v'_t is the origin of some edge $f_{t'} \in \{f_1, \dots, f_{t-1}\}$, then stop.[18] The procedure has formed a cycle $(f_{t'}, f_{t'+1}, \dots, f_t)$ composed of edges in $\{f_1, \dots, f_t\}$. 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 $f_{t+1} = (v_{t+1}, v'_{t+1})$.

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

(2) **Termination - Cycle**
   (a) Construct a set of flows associated with edges (P_e^1) which is the same as (P_e), except for flows $(P_{e_\tau})_{t' \le \tau \le t}$, 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 $P_{e_\tau}^1 = P_{e_\tau} + \alpha$, and each edge e_τ such that $f_\tau \neq e_\tau$, set $P_{e_\tau}^1 = P_{e_\tau} - \alpha$, where $\alpha > 0$ is the largest number such that the induced random assignment $P^1 = (P_x^1)_{x \in X}$ still satisfies all constraints in $\mathcal{H}$. By construction, $P_S^1 = P_S$ if P_S is an integer, and there is at least one constraint set $S \in \mathcal{H}$ such that P_S^1 is an integer while P_S is not. Thus $\deg[P^1(\mathcal{H})] > \deg[P(\mathcal{H})]$.
   (b) Construct a set of flows associated with edges (P_e^2) which is the same as (P_e), except for flows $(P_{e_\tau})_{t' \le \tau \le t}$, 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 $P_{e_\tau}^1 = P_{e_\tau} - \beta$, and each edge e_τ such that $f_\tau \neq e_\tau$, set $P_{e_\tau}^1 = P_{e_\tau} + \beta$, where $\beta > 0$ is the largest number such that the induced random assignment $P^2 = (P_x^2)_{x \in X}$ still satisfies all constraints in $\mathcal{H}$. By construction, $P_S^2 = P_S$ if P_S is an integer, and there is at least one constraint set $S \in \mathcal{H}$ such that P_S^2 is an integer while P_S is not. Thus $\deg[P^2(\mathcal{H})] > \deg[P(\mathcal{H})]$.
   (c) Set γ by $\gamma\alpha + (1-\gamma)(-\beta) = 0$, i.e., $\gamma = \frac{\beta}{\alpha+\beta}$.
   (d) The decomposition of P into $P = \gamma P^1 + (1-\gamma)P^2$ satisfies the requirements of the Lemma by construction.

Appendix C. Proofs of Lemma 1 and Theorem 2

Proof of Lemma 1. Suppose for contradiction that $\mathcal{H}$ is decomposable and contains the odd cycle $S^1, \dots, S^l$, with $x_i \in S_i \cap S_{i+1}$, $i = 1, \dots, l-1$ and $x_l \in S_l \cap S_1$. Consider a random assignment P specified by

$$P_x = \begin{cases} \frac{1}{2} & \text{if } x \in \{x_1, \dots, x_l\}, \\ 0 & \text{otherwise,} \end{cases}$$

where P_x is the entry corresponding to $x \in N \times O$. By definition of an odd cycle, $P_{S_i} = 1$ for all $i \in \{1, \dots, k\}$. Since $\mathcal{H}$ is decomposable, there exist $P^1, P^2, \dots, P^K$ and $\lambda^1, \lambda^2, \dots, \lambda^K$ such that

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

In particular, it follows that $P^k_{S_i} = 1$ for each i and k. Thus there exists k such that $P^k_{x_1} = 1$. Since $P^k_{S_2} = 1$, it follows that $P^k_{x_2} = 0$. The latter equality and the assumption that $P^k_{S_3} = 1$ imply $P^k_{x_3} = 1$. Arguing inductively, it follows that $P^k_{x_i} = 0$ if i is even and $P^k_{x_i} = 1$ if i is odd. In particular, we obtain $P^k_{x_l} = 1$ since l is odd by assumption. Thus $P^k_{S_l} = P^k_{x_l} + P^k_{x_1} >= 2$, contradicting $P^k_{S_l} = 1$. □

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

$$x_1 = (i, a), x_2 = (i, c), x_3 = (k, c), x_4 = (k, b), x_5 = (j, b).$$

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

- 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

$$x_1 = (i, a), x_2 = (i, b), x_3 = (j, b).$$

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

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

$$x_1 = (i,a), x_2 = (i,b), x_3 = (i,c), x_4 = (j,c), x_5 = (j,a).$$

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

(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

$$x_1 = (j,a), x_2 = (k,b), x_3 = (i,b).$$

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

(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 D. Proof of Theorem 3

Proof. For each $i \in N$, let $(a_i^1, a_i^2, \ldots, a_i^{|O|})$ be a sequence of objects in decreasing order of i's preferences so that $v_{ia_i^1} \geq v_{ia_i^2} \geq \ldots, v_{ia_i^{|O|}}$. Define the class of sets $\mathcal{H}' = \mathcal{H}'_N \cup \mathcal{H}'_O$ by

$$\mathcal{H}'_N = \mathcal{H}_N \cup \left(\bigcup_{\substack{i \in N, \\ k \in \{1,\ldots,|O|\}}} \{i\} \times \{a_i^1, \ldots, a_i^k\} \right),$$

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

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

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

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

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

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

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

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

These observations imply that

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

where the first inequality follows from $v_{ia_i^k} \ge v_{ia_i^{k'}}$ for $k < k'$ and Observations 1 and 3-(ii), the second inequality follows from $v_{ia_i^k} \ge v_{ia_i^{k'}}$ for $k < k'$, and the last inequality follows from the definition of Δ_i and Observation 3-(i). Therefore, we obtain property (5.1). Property (5.2) follows immediately from property (5.1). □

Appendix E. Proof of Corollary 2

Proof. Let P^* and ω^* be a solution and the optimal value of the associated linear programming problem. Observe that we can assume $P^*_{ia} = 0$ for every $(i, a) \in N \times O$ with $v_{ia} < 0$ without loss of generality. Introduce a "null object" such that $v_i = 0$ for all i and define $P^*_i = \bar{q}_{\{i\}\times O} - P^*_{\{i\}\times O}$ for all i. With these technical definitions Theorem 3 is applicable to $[P^*_{ia}]_{(i,a)\in N\times(O\cup\{\})}$, and it implies that there exists $P' = [P'_{ia}]_{(i,a)\in N\times O}$ that is integer-valued, satisfies all the constraints in problem (5.3), and

$$\sum_a P'_{ia}v_{ia} \ge \sum_a P^*_{ia}v_{ia} - \overline{v}_i, \tag{E.1}$$

for each i.[19] Since $\sum_a P^*_{ia}v_{ia} \ge \omega^*$ for each i by construction, inequality $(E.1)$ implies that $\sum_a P'_{ia}v_{ia} \ge \omega^* - \overline{v}_i$, implying $\omega' \ge \omega^* - \max_{i\in N}\overline{v}_i$. Finally, $w^* \ge w^{**}$ since w^* is the optimal value of a less constrained problem than problem (5.3). Thus we have $\omega' \ge \omega^{**} - \max_{i\in N}\overline{v}_i$, completing the proof. □

References

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

[19]Note that $\Delta_i = \max\{v_{ia} - v_{ib} | a, b \in O \cup \{\}, P^*_{ia}, P^*_{ib} \notin \mathbb{N}\} \le \max\{v_{ia} | a \in O, P^*_{ia} \notin \mathbb{N}\} = \overline{v}_i$ since $P^*_{ia} = 0$ for all a with $v_{ia} < 0$ by assumption on P^*.

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

Abdulkadiroğlu, A., and T. Sönmez (2003a): "Ordinal Efficiency and Dominated Sets of Assignments," *Journal of Economic Theory*, 112, 157–172.

——— (2003b): "School Choice: A Mechanism Design Approach," *American Economic Review*, 93, 729–747.

Bezáková, I., and V. Dani (2005): "Allocating Indivisible Goods," *ACM SIGecom Exchanges*, 5, 11–18.

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

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

——— (2002): "A Simple Random Assignment Problem with a Unique Solution," *Economic Theory*, 19, 623–635.

Border, K. C. (1991): "Implementation of Reduced Form Auctions: a Geometric Approach," *Econometrica*, 59, 1175–1187.

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

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

Budish, E., Y.-K. Che, F. Kojima, and P. Milgrom (2009): "A Generalization of the Probabilistic Serial Mechanism," In Preparation.

Che, Y.-K., D. Condorelli, and J. Kim (2009): "Weak Cartel and Optimal Auctions," Unpublished mimeo.

Che, Y.-K., and I. Gale (2000): "The Optimal Mechanism for Selling to a Budget-Constrained Buyer," *Journal of Economic Theory*, 92, 198–233.

Che, Y.-K., and F. Kojima (2008): "Asymptotic Equivalence of the Random Priority and Probabilistic Serial Mechanisms," forthcoming, *Econometrica*.

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

de Werra, D. (1984): "Preemptive Scheduling, Linear Programming and Network Flows," *SIAM Journal of Algebraic and Descrite Methods*, 5, 11–20.

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, pp. 69–87.

Ehlers, L., and B. 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, J. W. (2008): "Strategy-Proof, Efficient, and Nonbossy Quota Allocations," forthcoming, *Social Choice and Welfare.*

Hatfield, J. W., and P. Milgrom (2005): "Matching with Contracts," *American Economic Review*, 95, 913–935.

Hoffman, A., and J. 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, A., and R. 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, O. (2007): "Why do popular mechanisms lack efficiency in random environments?," *Journal of Economic Theory*, forthcoming.

Kojima, F. (2008): "Random Assignment of Multiple Indivisible Objects," forthcoming, *Mathematical Social Sciences.*

Kojima, F., and M. Manea (2008): "Incentives in the Probabilistic Serial Mechanism," forthcoming, *Journal of Economic Theory.*

Lenstra, J. K., D. B. Shmoys, and E. Tardos (1990): "Approximation Algorithms for Scheduling Unrelated Parallel Machines," *Mathematical Programming*, 46, 259–271.

Lewandowski, J. L., C. L. Liu, and J. W. S. Liu (1986): "An Algorithmic Proof of a Generalization of the Birkhoff-von Neumann Theorem," *Journal of Algorithms*, 7, 323–330.

Lovász, L., and M. Plummer (1986): "Matching Theory.," *Ann. Discrete Math.*

Manea, M. (2006): "A Constructive Proof of The Ordinal Efficiency Welfare Theorem," forthcoming, *Journal of Economic Theory.*

——— (2009): "Asymptotic Ordinal Inefficiency of Random Serial Dictatorship," forthcoming, *Theoretical Economics.*

Maskin, E. (2000): "Auctions, Development, and Privatization: Efficient Auction with Liquidity-Constrained Buyers," *European Economic Review*, 44, 667–681.

Maskin, E., and J. Riley (1984): "Optimal Auctions with Risk Averse Buyers," *Econometrica*, 52, 1473–1518.

Matthews, S. A. (1984): "On the Implementability of Reduced Form Auctions," *Econometrica*, 52, 1519–1522.

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

Milgrom, P. R. (2008): "Assignment Messages and Exchanges," forthcoming, *American Economic Journal: Microeconomics*.

Moore, J. (1984): "Global Incentive Constraints in Auction Design," *Econometrica*, 52, 1523–1535.

Myerson, R. (1981): "Optimal Auction Design," *Mathematics of Operations Research*, 6, 58–73.

Nemhauser, G., and M. Trick (1998): "Scheduling a Major College Basketball Conference," *Operations Research*, 46, 1–8.

Pai, M., and R. Vohra (2009): "Optimal Auctions with Financially Constrained Bidders," Northwestern University, unpublished mimeo.

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

Pratt, J. W. (2007): "Fair (and Not So Fair) Division," *Journal of Risk and Uncertainty*, 35, 203–236.

Pratt, J. W., and R. J. Zeckhauser (1990): "The fair and efficient division of the Winsor family silver," *Management Science*, 36(11), 1293–1301.

Roth, A. E., T. Sonmez, and U. Ünver (2005): "Pairwise Kidney Exchange," *Journal of Economic Theory*, 125, 151–188.

Roth, A. E., and M. A. O. Sotomayor (1990): *Two-sided matching: a study in game-theoretic modeling and analysis*. Econometric Society monographs, Cambridge.

Shapley, L., and M. Shubik (1972): "The Assignment Game I: The Core," *International Journal of Game Theory*, 1, 111–130.

Shmoys, D. B., and E. Tardos (1993): "An Approximation Algorithm for the Generalized Assignment Problem," *Mathematical Programming*, 62, 461–474.

Sönmez, T., and U. Ünver (2008): "Course Bidding at Business Schools," forthcoming, *International Economic Review*.

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

Watkins, W., and R. Merris (1974): "Convex Sets of Doubly Stochastic Matrices," *Journal of Combinatorial Theory, Series A*, 16, 129–130.

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