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

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

Abstract. The Birkhoff-von Neumann Theorem shows that any bistochastic matrix can be written as a convex combination of permutation matrices. In particular, in a setting where n objects must be assigned to n agents, one object per agent, any random assignment matrix can be resolved into a deterministic assignment in accordance with the specified probability matrix. We generalize the theorem to accommodate constraints encountered in many real-life market design problems. Specifically, the theorem can be extended to any environment in which the set of constraints can be partitioned into two hierarchies. Further, we show that this bihierarchy structure constitutes a maximal domain for the theorem, and we provide a constructive algorithm for implementing a random assignment under bihierarchical constraints. We provide several applications, 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 beyond 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 Auction.

1. Introduction

A prevalent allocation problem facing an organization is to assign agents to indivisible objects. Examples range from the allocation of slots in public schools, to the assignment of tasks within an organization, to the allocation of course seats or dormitory rooms at universities, to the allocation of organs amongst patients needing transplants. In many

---

Date: August 4, 2009.

Budish: NEED UPDATING. Che: Department of Economics, Columbia University, 420 West 118th Street, 1016 IAB New York, NY 10027, 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 Mihai Manea, Hervé Moulin and Al Roth for helpful comments. We especially thank Jay Sethuraman who pointed out close connections between our analysis and integer programming literature.

such problems, monetary transfers are impractical or undesirable or outright illegal, so they are not used. Introducing randomness in the assignment then becomes crucial for ensuring fairness, for assignment of potentially heterogeneous objects can be very unfair, no matter how it is done. Randomization over different assignments can help to "even things out."

This paper answers a fundamental question in such market design problems: *what random assignments can be implemented (and how)*.[1] To motivate the conceptual difficulty, consider a random assignment in which three agents $\{1, 2, 3\}$ are assigned to three objects $\{a, b, c\}$ according to the following matrix:

$$P = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \end{pmatrix}$$

where entry P_{ia} represents the probability that agent i is assigned to object a. If we resolve each agent's lottery independently, we might allocate some object to two agents while some other object goes unallocated, violating feasibility. To be feasible, whenever agent 1 receives object a, she cannot have anything else, so b should be assigned to agent 2, which in turn implies agent 3 must receive c. Likewise, whenever agent 1 does not receive object a, she must receive b, and a and c must go to agents 3 and 2, respectively. Consequently, random assignment is resolved by choosing the assignments $(1 \to a, 2 \to b, 3 \to c)$ and $(1 \to b, 2 \to c, 3 \to a)$ with equal probabilities.

As the above example illustrates, implementation of a given random assignment requires resolving uncertainty in a way that judiciously correlates assignments across agents. The precise method for correlating one agent's allocation to another's is not always obvious from the random assignment itself, and whether a method can be found to implement *every* arbitrary random assignment is unclear. The celebrated Birkhoff-von Neumann theorem (Birkhoff (1946); von Neumann (1953)) provides an answer for this simple setting. It shows that any bistochastic matrix[2] can be expressed as a convex combination of permutation matrices, i.e., bistochastic matrices each containing only zeros or ones as entries. This ensures that any random assignment can be implemented in the single-unit assignment setting, in which the number of agents equals the number of objects, and agents have unit demand. The theorem has proved useful in two important market design

[1]Through, we use the term "implementation" for the particular purpose described here; in particular, the term will not used to mean "satisfying incentive compatibility," as is often used in the literature.

[2]A square matrix is bistochastic if (i) all entries are between 0 and 1 inclusive, and (ii) all rows and columns sum to one.

algorithms, namely the pseudo-market mechanism by Hylland and Zeckhauser (1979) and more recently the probabilistic serial mechanism by Bogomolnaia and Moulin (2001).

Many real world assignment problems, however, include features and constraints that are not accommodated by the Birkhoff-von Neumann theorem (indeed, despite their theoretical appeal, we are not aware of either the pseudo-market mechanism or the probabilistic serial mechanism ever being used in practice). Two useful generalizations are straightforward to handle. First, in many allocation problems it is acceptable for some agents or objects to remain unassigned; for instance, in the (public) school choice problem, some students may exit the system, opting for a private school. Second, some agents might consume multiple units; for instance, in the course-allocation problem, students seek a schedule consisting of multiple courses. The first generalization can be accomodated easily by introducing a "null" object, which represents the option of not receiving any proper object. The second is only slightly more complicated.

Other constraints are more challenging to deal with. One difficult problem occurs when different groups of agents must be treated differently in the assignment. For instance, schools may wish to seek balance between the genders, or may require that the number of students of a certain racial, ethnic, geographic or income group may not exceed and/or fall short of some target quota. Another difficult problem occurs when the supply of objects is not exogenously fixed but rather produced endogenously according to some technological constraint. This feature arises when a public school authority wishes to install multiple school programs in one building and the relative sizes of these programs can be changed, but the total number of students in these programs is constrained by the building size. In multi-unit assignment problems such as course allocation, agents might have constraints on the kinds of objects they are able to consume. For instance, a student might have a preference to take no more than a certain number of courses in a particular subject matter, or be required to take at least a certain number.

These are just a few well-known examples of constraints that are not readily accommodated by the Birkhoff-von Neumann Theorem. There are many other similar examples. It is thus useful to extend the theorem beyond the original one-to-one setting.

We generalize the Birkhoff-von Neumann theorem in two directions. First, we allow for the number of agents to differ from the number of objects, and for the supply of objects and agents' demands for them to take any integer amount, positive or negative (a negative integer is interpreted as supply by that agent of the object in question). Accordingly, the notion of a random assignment is generalized to allow for any real (positive or negative) values for the entries of the associated matrix.

The second direction is in terms of the kinds of constraints we accommodate. In principle, it is possible to define constraints on any subset of entries in the matrix. The Birkhoff-von Neumann theorem allows for just two kinds of constraints: constraints on whole rows of the matrix (the number of objects an agent consumes), and constraints on whole columns of the matrix (the number of times an object is allocated). We allow for constraints to be placed on arbitrary subsets of the matrix, and we allow for both floor and ceiling constraints, not necessarily equal to each other as in the Birkhoff-von Neumann theorem. What we require is that the set of constraint sets can be partitioned into two hierarchies; that is, for any pair of sets in the same hierarchy, either they are disjoint or one set is a subset of the other. Under this **bihierarchy** structure, we show that any generalized random assignment matrix can be expressed as a convex combination of integer-valued matrices satisfying all the constraints on the sets in the two hierarchies. This means that as long as the subsets of entries that are subject to quota constraints form a bihierarchical structure and the quota ceilings/floors are integer-valued, any random assignment satisfying these constraints can be implemented by conducting a lottery over non-random assignments each of which satisfies all the constraints. Moreover, we provide a constructive algorithm for implementing random assignments under bihierarchical constraints.

The above extension is the most general statement to our knowledge and accommodates all kinds of real-world constraints we discussed above. Furthermore, we show that ours is the most general statement possible subject to some technical conditions. That is, if the desired constraints do not form a bihierarchy, then there exists a random assignment that satisfies all the constraints but cannot be implemented by a lottery over outcomes each of which satisfies the constraints.

We believe that our generalization of the Birkhoff-von Neummann theorem will find a wide variety of applications in mechanism design (and that the "necessity" result will facilitate the understanding of what kinds of random mechanisms are *not* possible). In this paper we discuss two distinct ways to use the bihierarchical structure to yield new design possibilities.

The first kind of application involves extending some known mechanisms that require the Birkhoff-von Neumann theorem for implementation to accommodate real-life constraints mentioned above. To motivate, consider assignment problems such as school choice, or housing or office allocation, where one agent is assigned to one object. This problem is commonly handled by a procedure known as the random priority mechanism

(RP). A priority mechanism is a deterministic mechanism in which agents choose their favorite remaining objects one at a time according to an exogenously specified priority order. In the random priority mechanism, agents are randomly ordered with equal probability and then the priority mechanism is conducted according to the realized order. Because the randomization rule is specified by the mechanism itself, RP can be conducted without the Birkhoff-von Neumann theorem and can be easily made to accommodate such features as group-specific quotas and/or flexible capacity constraints. (For example, suppose an agent reaches her turn to choose; if the group quota of her favorite remaining object has been already filled, then the object can be made simply unavailable to her.)

One problem with RP, though, is that the assignment it produces is inefficient ex ante in the sense that all agents can be made better off by another assignment that increases their chances of obtaining more preferred objects (the nature of inefficiency will be explained more formally later.) An influential recent paper by Bogomolnaia and Moulin (2001) proposes a mechanism called probabilistic serial (PS), which eliminates the form of inefficiencies under RP.[3] Despite this advantage, we are not aware of any incidences where PS has ever been used in practice. PS directly produces a random assignment in response to agents' reports of their preferences, so it requires the Birkhoff-von Neumann theorem for implementation. One practical issue with PS is whether, given its reliance on Birkhoff-von Neumann theorem, the mechanism can be adapted to accommodate the kinds of real-life constraints mentioned above. We suggest one possible adaptation of PS that handles general bihierarchical constraints and can be implemented by our generalization of Birkhoff-von Neumann theorem.

In our second application, we show how our method can be used to implement *any* random assignment in a way that guarantees that agents' utilities are always "close" to the (expected) utility associated with the random assignment. This result is of course vacuous in the context of single-unit assignment. But it can be quite powerful in the context of multi-unit resource allocation problems such as course allocation, task assignment, and the fair division of estates. One attractive method for solving such problems is to treat objects as "divisible" and solve for an optimal fractional (random) assignment. The difficulty is that there are many ways to resolve a given random assignment, some of which could entail an outcome quite different from the original random assignment. For

[3] RP is strategyproof in any market, whereas PS is strategyproof only in an ordinal sense; namely, no agent under that mechanism can do strictly better by lying in the sense of first-order stochastic dominance (Bogomolnaia and Moulin 2001). In large finite markets PS becomes strategyproof (Kojima and Manea 2008).

instance, suppose in the context of course allocation that under an attractive divisible-goods allocation procedure some student receives a fractional assignment in which he has a one-half probability of obtaining each of twenty courses, ten of which are "good" and ten of which are "bad." The only constraint is that he receive ten courses overall. One way to resolve this lottery would be to give the agent a one-half chance of obtaining all ten good courses and a one-half chance of obtaining all ten bad courses. This resolution exposes the agent to substantial risk. We propose a method to avoid such resolution of uncertainty. The idea is to "impose" artificial new constraints in a way that bounds the extent to which each agent's utility can vary over different resolutions of the (artificially constrained) random assignment. We add different constraints for different agents in a way that depends on their preferences, and use the generalized Birkhoff-von Neumann theorem to ensure that the resulting artificially constrained random assignment can be implemented.

Our framework lends itself to extension to the two-sided matching setting in which both sides of the market are agents. This can be done by interpreting the object side as another set of agents. The utility guarantee can then be readily adapted to a two-sided matching problem: starting with any random matching, we can find an exact matching that always gives the agents on both sides realized utilities that are similar to those they expect from the random matching. This method can be used to design a fair schedule of inter-league matchups in sports scheduling or a fair speed-dating mechanism.

Related Literature. The paper extends the Birkhoff-von Neumann Theorem due to Birkhoff (1946) and von Neumann (1953). Extensions of this theorem have been obtained by Watkins and Merris (1974), Lewandowski, Liu, and Liu (1986) and de Werra (1984), among others. All these results are special cases of this paper. (***to be expanded***)

In a precursor to our work, Milgrom (2008) extends the Shapley-Shubik model of matching by introducing a particular bihierarchy of constraints to describe substitution possibilities among various goods. He shows that a competitive equilibrium in integer assignments exists, that the bihierarchy constraints describe goods that are substitutes (no complementarities), and that certain constraints must form a hierarchy for certain results, including the substitutes property, to be satisfied generally. (*** give more discussion? ***)

*** write about the literatures on application (PS, course allocation, etc.)***

2. Setup

An **environment** is a tuple $\mathcal{E} = \langle N, O, \mathcal{H}, q \rangle$ where N and O are sets of objects and agents where $|O|, |N| \geq 2$, $\mathcal{H} \subset 2^{N \times O}$ is a set of subsets from $N \times O$ that includes all singletons, and $q = (\underline{q}_S, \overline{q}_S)_{S \in \mathcal{H}}$ is the set of quota constraints for 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 $S \in \mathcal{H}$. Our convention to include all singleton sets in $\mathcal{H}$ follows from the natural real-life restriction that individual assignments are discrete. This convention simplifies our analysis but plays no important role. We denote by $\mathcal{P}_{\mathcal{E}}$ the set of random assignments that are feasible in $\mathcal{E}$.

Definition 1. The constraint structure $\mathcal{H}$ is **Birkhoff-von Neumann decomposable (BvN decomposable)** if, for each $(\underline{q}_S, \overline{q}_S)_{S \in \mathcal{H}}$ and P with $\underline{q}_S \leq P_S \leq \overline{q}_S$ for all $S \in \mathcal{H}$, there exist $\lambda^1, ..., \lambda^K$ and $P^1, ..., P^K$ such that

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

If $\mathcal{H}$ is BvN 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 constraints in $\mathcal{H}$ can be implemented as a lottery over deterministic outcomes that respects constraints in $\mathcal{H}$. Equivalently, $\mathcal{H}$ is BvN decomposable if and only if for every random assignment P there exist $\lambda^1, \lambda^2, \ldots, \lambda^K$ and $P^1, ..., P^K$ such that

(1) $P = \sum_{k=1}^{K} \lambda^k P^k$,
(2) $\lambda^k > 0, k = 1, ..., K$, and $\sum_{k=1}^{K} \lambda^k = 1$,

(3) $P^k_S \in \{\lfloor P_S \rfloor, \lceil P_S \rceil\}$ for all $k \in \{1, \dots, K\}$ and $S \in \mathcal{H}$.[4]

This definition requires that each assignment in decomposition round the random assignment up or down to the nearest integer, with respect to each $S \in \mathcal{H}$. This definition coincides with the original definition since the nearest integer to any feasible random assignment must satisfy any (integer-valued) quotas and since the singleton sets are all included in $\mathcal{H}$ by assumption.

3. The Rounding Theorem

A collection $\mathcal{H}$ of constraint sets is a **hierarchy** if $S \subset S'$ or $S' \subset S$ or $S \cap S' = \emptyset$ for every $S, S' \in \mathcal{H}$. 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. (Partition 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 latter two in any arbitrary fashion.)

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

The proof of Theorem 1 is in the appendix. We also provide in the appendix a constructive algorithm for implementing the theorem. The Theorem 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.

Below is a particularly simple environment, which we call the Birkhoff-von Neumann environment. The environment is defined by $\mathcal{E}^{BvN} \equiv \langle N, O, \mathcal{H}^{BvN}, q \rangle$ where

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

[4] 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.

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 $\mathcal{H}_N^{BvN}$ and $\mathcal{H}_O^{BvN}$ where

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

and clearly $\mathcal{H}_N^{BvN}$ and $\mathcal{H}_O^{BvN}$ 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 Theorem 1.

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

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

3.1.1. *Flexible Capacity, Group-specific Quotas.* Consider situations where objects can be produced or procured in a flexible manner subject to certain constraints. In the context of school choice, the school authority may wish to run several education programs within one building; in such a case, relative sizes of different programs may be adjustable as long as the total number of attending students is within capacity of the building. Similarly, within each education program, different disciplines and activities may be provided flexibly subject to resource constraints in the program. Such a situation can be represented by a hierarchy $\mathcal{H}_O$ containing sets of the form $S = N \times O'$ with $|O'| \geq 2$. The ceiling $\overline{q}_S$ then describes the total capacity that can be allocated within O'. Note that the hierarchical structure allows for the flexible production to be nested; e.g., a subset of programs may be chosen and, within each chosen program, a subset of subprograms may be chosen, and so on.

As another case in which $\mathcal{H}_O$ plays a role, consider a situation in which the mechanism designer needs to treat different groups of agents differently. For example, affirmative action policies may impose quotas specific to applicants of certain racial or economic profiles.

Such a practice is called "controlled choice" and is used in many school districts in the United States.[5] Some other forms of constraints are mathematically similar. For example, a subset of schools in New York City (the so-called Educational Option programs) require balanced distributions of test score: Namely, 16 percent of the seats should be allocated to students who were rated top performers in a standardized English Language Arts exam, 68 percent to middle performers, and 16 percent to lower performers (Abdulkadiroğlu, Pathak, and Roth 2005).[6] 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 to 20 percent,[7] and a number of school choice programs in Japan have similar quotas based on residential areas as well. Such constraints 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.

Moreover, clearly it is possible to express both of the above constraints simultaneously in $\mathcal{H}_O$ that forms a hierarchy.

3.1.2. *Course Allocation.* Consider the course allocation problem described as follows. Within a school, seats in courses need to be allocated to students. Each course has its own quota. Each student can enroll in more than one course but cannot receive more than one seat in the same course for an obvious reason. Moreover, she does not take more than a certain number from a subset of courses, either because the system prohibits it or because she does not want to. For example, a student might be prohibited 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.

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

[6]An exact implementation can be slightly different from the quotas analyzed by most of the literature cited above. See Kojima (2008a) for further discussion on this point.

[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).

Such restrictions can be modeled using a bihierarchy such that $\mathcal{H}_N \supset \mathcal{H}_N^{BvN}$. By setting $\overline{q}_{\{(i,a)\}} = 1$ and $\overline{q}_{\{i\}\times O} > 1$ for each $i \in N$ and $a \in O$, we can assure that student i can enroll up to $\overline{q}_{\{i\}\times O} > 1$ courses while she will receive 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).[8]

Note that the hierarchy $\mathcal{H}_O$ can be an arbitrary hierarchy without violating a bihierarchical structure of $\mathcal{H} = \mathcal{H}_N \cup \mathcal{H}_O$. A consequence of this observation is that flexible production and group-specific quota as described in Section 3.1.1 can be incorporated into the course allocation problem while retaining the bihierarchical structure of the constraint sets.

3.1.3. *Interleague Play.* Some professional sports, most notably 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.[9] 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 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 sets form a bihierarchy.

3.2. **Necessity of a bihierarchical structure.** What is the role of bihierarchy in making decomposition possible? We provide some answers to this question in this section.

[8]While very flexible, there are some limitations to the kinds of constraints that can be accommodated without violating bihierarchy. The course-allocation procedure proposed in Budish (2008) accommodates arbitrary constraints. See Section 5 for further discussion.

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

First of all, the following example offers a non-bihierarchical constraint structure that is not BvN decomposable.

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

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

Clearly, $\mathcal{H}$ is not a bihierarchy. Suppose each set in $\mathcal{H}$ has 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 constraint 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 $(S_1,\ldots,S_l) \in \mathcal{H}^l$ is an **odd cycle** if $S_i \neq S_j$ for all $i \neq j$, l is odd, and there exists a sequence $(x_1,\ldots,x_l) \in (N \times O)^l$ such that for each $i = 1,...,l$, $x_i \in S_i \cap S_{i+1}$ 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 shows the following observation (a formal proof is in the Appendix).

Lemma 1. *If there exists an odd cycle formed by elements of $\mathcal{H}$, then $\mathcal{H}$ is not BvN decomposable.*

This lemma helps to understand one role the bihierarchy structure plays: *It rules out odd cycles.* To see this, suppose $\mathcal{H}$ contains an odd cycle, $\{S_1,...,S_l\}$. Suppose they were partitioned into two hierarchies, $\mathcal{H}_N$ and $\mathcal{H}_O$, and 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, producing a contradiction.

Is bihierarchy necessary for BvN decomposition? The next example shows that this is not the case.

Example 2. Consider an environment with 2 goods and 2 agents as before, but let

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

with each quota being one. The following random assignment

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

can be decomposed by a convex combination of deterministic assignments as

$$P = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

Note that the current constraint sets do not allow for an odd cycle although it is not a bihierarchy.

Notice, however, Example 2 is somewhat non-standard in that some row and column constraints are not present. One could easily see that BvN decomposability would fail if all row and column constraints are added to the constraint sets of Example 2 as the new collection of constraint sets has an odd cycle. This observation turns out to be true more generally. We show that *bihierarchy is in fact necessary for BvN 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 BvN decomposable.*

Recall that the condition $\mathcal{H}^{BvN} \subset \mathcal{H}$ is natural in any bilateral matching setting and is imposed in all applications in this paper. The formal proof is in the Appendix. The basic strategy of the proof is to show that there exists an odd cycle whenever $\mathcal{H}$ is not a bihierarchy while containing $\mathcal{H}^{BvN}$.

Remark 1. In light of Examples 1 and 2, one might wonder whether an absence of odd cycles is sufficient for BvN 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.

4. Application: Single-unit assignment

Consider a problem of assigning indivisible objects to agents who can consume at most one object each. University housing allocation, public housing allocation, office assignment, and student placement in public schools are real-life examples.

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. 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 assignments is

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

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 any 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.

As discussed in Section 3.1.1, there are additional constraints of interest assumed away by Bogomolnaia and Moulin (2001). First, the objects may be produced or procured endogenously based on preferences of the agents. In the context of school choice, for instance, different types of objects may represent multiple school programs in the same building, where sizes of programs can be changed as long as the sum of students in these programs is constrained by the building size. There may also exist quotas specific to certain ethnic or economic groups, as in controlled choice programs of many school districts in the United States. As explained in Section 3.1.1, the former class of constraints can be represented by sets of the form $S = N \times O'$ with $|O'| \geq 2$ while the latter by $N' \times \{a\}$ for $a \in O$ and $N' \subsetneq N$. In either case the ceiling constraint $\overline{q}_S$ specifies the quota associated with S, while we set $\underline{q}_S = 0$ for all S.

Based on Theorem 1, the probabilistic serial mechanism can be generalized using the following eating algorithm. *As in $\mathcal{H}^{BvN}$, 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 sets form a bihierarchy, Theorem 1 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 real-life market design environments than has been possible.[10] An important remaining question is whether our adaptation of PS will continue to have the desirable properties in 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 from the current focus, so we postpone them to our companion paper (Budish, Che, Kojima, and Milgrom 2009).

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 heirloom 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 draft. 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 2008). Designing good course allocation mechanisms remains a challenging market design problem.[11]

One possible market design idea is, like the PS mechanism, to find a desirable random assignment directly. Such an approach is taken by Hylland and Zeckhauser (1979) in 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

[10]For instance, the upcoming Korean school choice program involves quotas at each school for applicants from outside districts, but the schools have no priorities. Hence the problem becomes essentially random assignment with group specific quotas. Currently, the algorithm appears to be that of the so-called Boston mechanism (see Abdulkadiroğlu and Sönmez (2003) for description). One can apply the PS instead, for instance.

[11]Axiomatic analysis on this problem has also obtained negative conclusions. Papai (2001) shows that sequential dictatorships (considered unrealistic for many applications) are the only deterministic mechanisms that are nonbossy, strategy-proof, and Pareto optimal. Ehlers and Klaus (2003), Hatfield (2008), and Kojima (2008b) provide similarly pessimistic results.

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 satisfies attractive fairness properties, such as envy freeness.) 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 e.g. Pratt (2007)). As with Hylland and Zeckhauser's mechanism, the resulting assignment promises to have desirable ex-ante efficiency and fairness properties. This simple multiunit extension of the Hylland-Zeckhauser model conforms to a bihierarchy, so our Theorem 1 can be used to implement the random assignments. Obviously, we can accommodate additional constraints within the bihierarchy framework.

In the multi-unit assignment setting, Theorem 1 has another important application. In multi-unit assignment, a given random assignment can be implemented in many different ways, some much less fair than others. To motivate, suppose that two agents are to divide n objects (where n is even) and ordinal preferences of agents are common. A fair random assignment would be for each agent to receive half of each object. One way to implement this random assignment is to give any $n/2$ randomly chosen objects to one agent and give the remaining $n/2$ to the other. This method could entail a highly unfair outcome ex post if n is large, however, in which one agent gets the $n/2$ best objects and the other gets the $n/2$ worst ones.

Based on Theorem 1, we provide a method to avoid such outcomes. To state the claim, 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 $\underline{q}_{\{i\}\times O} = \overline{q}_{\{i\}\times O}$ for each $i \in N$.

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 Theorem 1, and also:*

(1)
$$\sum_a P'_{ia} v_{ia} - \sum_a P''_{ia} v_{ia} \in [-\Delta_i, \Delta_i],$$

(2)
$$\sum_a P'_{ia} v_{ia} \in \left[\sum_a P_{ia} v_{ia} - \Delta_i, \sum_a P_{ia} v_{ia} + \Delta_i \right],$$

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

A proof sketch can be given based on Theorem 1. The idea is to supplement the actual constraints of the problem with a set of "artificial" hierarchical constraints, in a manner that limits variation in utilities. Specifically, if there are n objects, for each agent we create n additional constraints, one each for his j most preferred objects for $j = 1 \dots n$. The resulting collection of the constraint sets is still a bihierarchy after this addition, so Theorem 1 guarantees that the random assignment can be implemented with all of the constraints satisfied. Satisfying the new constraints on all "upper contour" sets (i.e., set of j most preferred objects for each j) means that in each realized (deterministic) assignment, each agent receives her j most preferred objects, for each j, with approximately the same probability as the original random assignment. Our method thus ensures that each realized assignment inherits roughly the same fairness property as the original random assignment; more precisely, the approximation error is at most the utility difference between those of the most valuable and the least valuable objects. The current method is most useful when each agent obtains a large number of objects, since in such a case the valuation of one object will often be relatively small compared to the utility of the bundle of goods as a whole.

Remark 2. Theorem 3 can be applied to any given random assignment regardless of the mechanism to decide the random assignment. Thus the method can be used in conjunction with, for instance, a multi-unit demand generalization of the probabilistic serial mechanism, although the mechanism has more serious incentive issues in the multi-unit demand setting (Kojima 2008b).

5.1. **Further Applications of Theorem 3.** Theorem 3 turns out to be useful in other applications. This section describes them in some detail.

5.1.1. *The Maximin Approach to Fair Division.* Consider the following problem. The social planner has a number of indivisible objects O to be allocated to agents N. Utility of agents is additive in objects up to a fixed quota (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 social planner wishes 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:

$$\text{maximize } \omega \text{ subject to} \tag{3}$$

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

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 (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 (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 (3).

Corollary 2. *There exists a solution of the problem (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 (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 we acknowledge that Corollary 2 is only a mild extension, we emphasize the methodological innovation. Corollary 2 is shown by Theorem 3, which in turn is a direct consequence of Theorem 1. Our contribution here is that apparently dissimilar results such as utility guarantee and fair division mechanisms can be derived from one fundamental result.

5.1.2. *Scheduling Jobs on Parallel Machines: Minimize Makespan Problem.* Our approach can 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 job is 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 machines 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 (a fractional solution is often easy to find because linear programming techniques are applicable). 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* $c = (c_{ia})_{(i,a)\in N\times O} \in \mathbb{R}_+^{|N|\times|O|}$, $d = (d_a)_{a\in O} \in \mathbb{R}_+^{|O|}$ *and* $t \in \mathbb{R}_+$ *be given. If there is a feasible solution* P *to the (in)equalities,*

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

$$\sum_{a\in J_i(t)} P_{ia}c_{ia} \le d_i, \qquad \text{for } a \in O,$$

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

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

$$\begin{aligned}
(4) \qquad & \sum_{i \in M_a(t)} P'_{ia} = 1, && \textit{for } a \in O, \\
& \sum_{a \in J_i(t)} P'_{ia} c_{ia} \leq d_i + t, && \textit{for } a \in O, \\
& P'_{ia} \in \{0, 1\}, && \textit{for } a \in J_i(t), i \in N.
\end{aligned}$$

Proof. By Theorem 3, there exists P' that is integer-valued and satisfies (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, some agents in N can be matched with more than one agent in O each and vice versa.

Theorem 4. *Consider a problem as in Theorem 1 where the collection of constraint sets 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 a 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 all of the conditions of Theorem 1, 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 feasible match, that is, a deterministic assignment matrix, in which differences in strength of opponents is bounded by one game with the strongest opponent and one with the weakest opponent in the other league, no matter how many games are scheduled for each team.

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. We do not claim that our method can be directly used to such complicated situations. Rather, our point here is to suggest a possibility that our analysis may be a useful first step to solve some problems that have not been considered to be related to questions such as school choice or fair allocation. See Nemhauser and Trick (1998) for further discussion of sport scheduling.

7. Generalization

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

Let X be a finite set and $\mathcal{H}$ be a collection of subsets of X. We call 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 BvN decomposable if, for each $(\underline{q}_S, \overline{q}_S)_{S \in \mathcal{H}}$ and P with $\underline{q}_S \leq P_S \leq \overline{q}_S$ for all $S \in \mathcal{H}$, there exist $\lambda^1, ..., \lambda^K$ and $P^1, ..., P^K$ such that

(1) $P = \sum_{k=1}^K \lambda^k P^k$,
(2) $\lambda^k > 0, k = 1, ..., K$, and $\sum_{k=1}^K \lambda^k = 1$,
(3) P_x^k is an integer for each x,
(4) $\underline{q}_S \leq P_S^k \leq \overline{q}_S$ for each $k = 1, ..., K$ 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 BvN decomposable if either it forms a bihiearchy or its dual forms a bihierarchy.*

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 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.

8. Conclusion

We generalize the Birkhoff-von Neumann theorem so that the implementation of lotteries is possible whenever the set of constraints can be partitioned into two hierarchies. Thus, given any random assignment satisfying constraints in two hierarchies, the assignment can be realized by using a lottery over outcomes each of which satisfies all the constraints. Moreover, we provide a maximal domain result, that indicates that the bihierarchical structure is necessary (subject to a technical condition) to guarantee that a random assignment can always be implemented by lotteries over feasible outcomes. We presented several applications, including (i) random assignment mechanisms (especially the probabilistic serial mechanism), (ii) utility guarantee for problems with multi-unit demand, (iii) fair division, (iv) the minimize-makespan problem, (v) two-sided matching, and (vi) optimal assignment.

As the basic result is applicable to a wide range of situations as exemplified in this paper, we envision that the result will prove useful in other economic and non-economic applications. Finding more applications is an interesting topic left for the future.

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 $+1$.

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

We can easily use one of the conditions to say that if the incidence matrix of a hypergraph is totally unimodular, then the hypergraph is BvN decomposable.

Lemma 3. *(Ghouila-Houri) 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 $E_i^1, E_i^2, \ldots, E_i^{k_i}$ defined recursively as follows: Set $E_i^0 \equiv \emptyset$ and, for each $j = 1, ...,$ 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 $j = 1, ..., k_i$. 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 $j = 1, ..., l$ for some $l \leq k_1$. These sets alternate in colors in $j = 1, 2..,$ 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 $j = 1, ..., m$ for some $m \leq k_2$. These sets alternate in colors in $j = 1, 2..,$ 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 $E \cap \mathcal{H}_1$ containing x is less by one than or equal to the number of red sets in $E \cap \mathcal{H}_1$ 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. By Lemma 2, for any random assignment P, the vertices of the set

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

are integer valued, so the hypergraph $\mathcal{X}$ is BvN 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 hyperedges 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, $\mathcal{X}$ is BvN decomposable. □

A.1. **Computable Algorithm for Bihierarchy.** For the case of bihierarchy, we provide the following algorithm to find a decomposition. Observe that the algorithm gives a constructive proof of the Theorem for the bihierarchy case. We say a set $S \subset X$ is **integral** [resp. **nonintegral**] (under P) if $P_S \in \mathbb{Z}$. In case there is no confusion, we suppress the qualifier inside the parenthesis.

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 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 $i = 1, 2$.

Our algorithm consists of two parts: the Fission algorithm and the Decomposition algorithm.

□ **Fission Algorithm**

1. Within-Hierarchy Unnesting Phase

(1) Let $\mathcal{C}^0_i$, $i = 1, 2$, be the collection of all integral sets of $\mathcal{H}_i$ under P.

(2) **Step** $t = 1, \ldots$**:** Find sets S, S' in $\mathcal{C}^{t-1}_i$ such that $S \subsetneq S'$.

(a) If no such sets exist, *move to* the **Dividing Phase**.

(b) If such sets exist, find a pair of sets $S \subsetneq S'$ such that S' is not a subset of any other set in $\mathcal{C}_i^{t-1}$ and S' is the only proper superset of S' in $\mathcal{C}_i^{t-1}$ (it is easy to see that such sets exist). Then *remove* S' *from* $\mathcal{C}_i^{t-1}$ *and replace it by* $S' \setminus S$*, and call the resulting collection* $\mathcal{C}_i^t$*, and iterate to Step* $t+1$.

(3) The unnesting phase stops in finite interations for each $\mathcal{H}_i$, $i = 1, 2$. *Call the resulting collections* $\mathcal{D}_1^0$ *and* $\mathcal{D}_2^0$*, respectively, and move to* **Dividing Phase**.

2. Dividing Phase

(1) **Step** $t = 1, \ldots$ Find $S \in \mathcal{D}_1^{t-1}$ and $S' \in \mathcal{D}_2^{t-1}$ such that $S \setminus S'$, $S' \setminus S$ and $S \cap S'$ are all nonempty and integral.

(a) If no such sets exist, *stop and move to the Cross-Hierarchy Unnesting Phase.*

(b) If such sets exist, then *remove them from* $\mathcal{D}_1^{t-1} \cup \mathcal{D}_2^{t-1}$ *and add* $S \setminus S'$ *and* $S \cap S'$ *to* $\mathcal{D}_1^{t-1}$*, and add* $S' \setminus S$ *to* $\mathcal{D}_2^{t-1}$*, and call them* $\mathcal{D}_1^t$ *and* $\mathcal{D}_2^t$*, respectively, and move to Step* $t+1$.

(2) This process ends in finite steps. Call the resulting collections, $\mathcal{G}_1^0$ and $\mathcal{G}_2^0$, respectively, and move to the Cross-Hierarchy Unnesting Phase.

3. Cross-Hierarchy Unnesting Phase

(1) **Step** $t = 1, \ldots$ Find $S \in \mathcal{G}_i^{t-1}$ and $S' \in \mathcal{G}_j^{t-1}$, $i = 1, 2, j = 3-i$, such that $S \subset S'$.

(a) If no such sets exist, *stop and move to the Decomposition Algorithm.*

(b) If such sets exist, then remove S from $\mathcal{G}_i^{t-1}$ and S' from $\mathcal{G}_j^{t-1}$, and add S and $S' \setminus S$ to $\mathcal{G}_j^{t-1}$, and call the resulting collections $\mathcal{G}_i^t$ and $\mathcal{G}_j^t$, respectively, and iterate to Step $t+1$.

(2) This process ends in finite steps. Call the resulting collections, $\overline{\mathcal{H}}_1$ and $\overline{\mathcal{H}}_2$, respectively, and move to the Decomposition Algorithm.

We make several observations on $\overline{\mathcal{H}}_1$ and $\overline{\mathcal{H}}_2$:

□ **Observations**

(1) The sets in $\overline{\mathcal{H}}_1 \cup \overline{\mathcal{H}}_2$ are integral under P. Conversely, if all sets in $\overline{\mathcal{H}}_1 \cup \overline{\mathcal{H}}_2$ are integral under any $P' \in \mathcal{F}$, then any set in $\mathcal{H}$ that is integral under P is also integral under P'. This latter observation follows from the fact that, by the Fission Algorithm, any set in $\mathcal{H}$ that is integral under P can be expressed as a union of disjoint sets in $\overline{\mathcal{H}}_1 \cup \overline{\mathcal{H}}_2$.

(2) The sets within each $\overline{\mathcal{H}}_i$, $i = 1, 2$, are disjoint: Any nesting within each hierarchy is eliminated by the end of the within-unnesting phases, and the dividing phase and cross-unnesting phases do not create any new nesting within each hierarchy.

(3) There are no two sets $S, S' \in \overline{\mathcal{H}}_1 \cup \overline{\mathcal{H}}_2$ such that $S \subset S'$: This follows from Observation (2) and the cross-unnesting phase.

(4) Each element of X could be in at most one set of each $\overline{\mathcal{H}}_i$: This follows from Observation (2).

(5) If there are $S, S' \in \overline{\mathcal{H}}_1 \cup \overline{\mathcal{H}}_2$ such that $S \setminus S'$, $S' \setminus S$ and $S \cap S'$ are all nonempty, then all these sets must be non-integral: This follows from the original bihierarchy structure and the dividing phase.

Let $\mathcal{N} = \{x \in X | \nexists S \in \overline{\mathcal{H}}_1 \cup \overline{\mathcal{H}}_2 \text{ s.t. } x \in S\}$ be the elements of X not in any integral sets in $\mathcal{C}$.

□ **Decomposition Algorithm**

(1) **Step 0:**
   (a) If $\mathcal{N}$ is non-empty, then *circle any non-integral element of* $\mathcal{N}$ *and proceed to* **Termination - Dead End.**
   (b) If there does not exist any non-integral element in $\mathcal{N}$, then *move to* **Step 1.**

(2) **Step 1:**
   (a) If no set in $\overline{\mathcal{H}}_1 \cup \overline{\mathcal{H}}_2$ contains any non-integral element, then *stop the Decomposition Algorithm.* [Every element in X is then integral.]
   (b) If there exists a non-integral element x_1 in some $S_1 \in \overline{\mathcal{H}}_1 \cup \overline{\mathcal{H}}_2$, then *circle it and proceed to* **Step 2.**

(3) **Step** $t = 2, \ldots$**:**
   (a) In case $x_{t-1} \in S_{t-1}$ is circled, where $S_{t-1} \in \overline{\mathcal{H}}_i$, $i = 1, 2$:
      (i) Find a non-integral element in $S_{t-1} \setminus S_{t-2}$ different from x_{t-1}. (Let $S_0 \equiv \emptyset$.) Such an element exists since S_{t-1} is integral by Observation (1), and since, by Observation (5), $S_{t-1} \setminus S_{t-2}$ is non-integral whenever $S_{t-1} \cap S_{t-2}$ is non-empty. If there is any such element that is an element of S_τ for some $\tau \in \{1, \ldots, t-2\}$, then *erase the markings (e.g., circle and square) of all x_j, for $j = 1, ..., \tau - 1$,* and *proceed to* **Termination - Cycle** (note that $S_\tau \in \overline{\mathcal{H}}_{3-i}$ by observation (4)). Otherwise, choose one element $x_t \neq x_{t-1}$, $x_t \in S_{t-1} \setminus S_{t-2}$ arbitrarily and square it.
      (ii) If no set other than S_{t-1} contains x_t, then *stop and proceed to* **Termination - Chain**.
      (iii) Suppose another set, S_t, contains x_t. Then, $S_t \in \overline{\mathcal{H}}_{3-i}$ since, by Observation (2), all sets in $\overline{\mathcal{H}}_i$ are disjoint. Also, by the construction in Step 3(a)i above, $S_t \neq S_\tau$ for any $\tau = \{1, \ldots, t-1\}$. *Proceed to* **Step** $t + 1$.
   (b) In case $x_{t-1} \in S_{t-1}$ is squared, where $S_{t-1} \in \overline{\mathcal{H}}_i$, $i = 1, 2$:

(i) Find a non-integral element in $S_{t-1} \setminus S_{t-2}$ different from x_{t-1}. (Let $S_0 \equiv \emptyset$.) Such an element exists since S_{t-1} is integral by Observation (1), and since, by Observation (5), $S_{t-1} \setminus S_{t-2}$ is non-integral whenever $S_{t-1} \cap S_{t-2}$ is non-empty. If there is any such element that is an element of S_τ for some $\tau \in \{1, \dots, t-2\}$, then *erase the markings (e.g., circle and square) of all x_j, for $j = 1, ..., \tau - 1$,* and *proceed to* **Termination - Cycle** (note that $S_\tau \in \overline{\mathcal{H}}_{3-i}$ by observation (4)). Otherwise, choose one element $x_t \neq x_{t-1}$, $x_t \in S_{t-1} \setminus S_{t-2}$ arbitrarily and circle it.

(ii) If no set other than S_{t-1} contains x_t, then *stop and proceed to* **Termination - Chain**.

(iii) Suppose another set, S_t, contains x_t. Then, $S_t \in \overline{\mathcal{H}}_{3-i}$ since, by Observation (2), all sets in $\overline{\mathcal{H}}_i$ are disjoint. Also, by the construction in Step 3(b)i above, $S_t \neq S_\tau$ for any $\tau = \{1, \dots, t-1\}$. *Proceed to* **Step** $t+1$.

Since X is finite, we eventually reach the Termination Step.

(4) **Termination - Dead End**

(a) Construct a random assignment P^1 which is the same as P, except at the circled element x. P^1_x is obtained by raising P_x as high as possible without "crossing" any constraint in $\mathcal{H}$. This amount α is positive.

(b) Construct a random assignment P^2 which is the same as P, except at the circled element x. P^2_x is obtained by reducing P_x as low as possible without "crossing" any constraint in $\mathcal{H}$. The amount of reduction β is positive.

(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.

(5) **Termination - Chain**

(a) Observe that the sets in $\overline{\mathcal{H}}_1 \cup \overline{\mathcal{H}}_2$ containing either circled or squared term does not form a cycle. By construction, each of these sets contains precisely one circled element and one squared element.

(b) Construct a random assignment P^1 in $\mathcal{F}$ which is the same as P, except at the circled elements, X_C, and at squared elements, X_S. For each $x \in X_C$, set $P^1_x = P_x + \alpha$, and for each $x \in X_S$, set $P^1_x = P_x - \alpha$, where $\alpha > 0$ is the largest number that still satisfies all constraints in $\mathcal{H}$. By construction, $P^1_S = P_S$ for all $S \in \overline{\mathcal{H}}_1 \cup \overline{\mathcal{H}}_2$. Observation (1) then ensures that $P^1_S = P_S$ for each integral set $S \in \mathcal{H}$.

(c) Construct a random assignment P^2 which is the same as P, except at the circled elements, X_C, and at squared elements, X_S. For each $x \in X_C$, set $P^2_x = P_x - \beta$, and for each $x \in X_S$, set $P^2_x = P_x + \beta$, where $\beta > 0$ is the largest number that still satisfies all constraints in $\mathcal{H}$. By construction, $P^2_S = P_S$ for all $S \in \overline{\mathcal{H}}_1 \cup \overline{\mathcal{H}}_2$. Observation (1) then ensures that $P^2_S = P_S$ for each integral set $S \in \mathcal{H}$.

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

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

(6) **Termination - Cycle**

(a) Observe that the sets $S_t, ..., S_{t+k}$ form a cycle (recall that $t = 1$, unless the construction involves the erasing of markings). Since these sets altenate between $\overline{\mathcal{H}}_i$ and $\overline{\mathcal{H}}_{3-i}$, the order of the cycle must be even. Further, by construction, each of these sets contains precisely one circled element and one squared element.

(b) Construct a random assignment P^1 which is the same as P, except at the circled elements, X_C, and at squared elements, X_S. For each $x \in X_C$, set $P^1_x = P_x + \alpha$, and for each $x \in X_S$, set $P^1_x = P_x - \alpha$, where $\alpha > 0$ is the largest number that still satisfies all constraints in $\mathcal{H}$. By construction, $P^1_S = P_S$ for all $S \in \overline{\mathcal{H}}_1 \cup \overline{\mathcal{H}}_2$. Observation (1) then ensures that $P^1_S = P_S$ for all integral set $S \in \mathcal{H}$.

(c) Construct a mapping P^2 which is the same as P, except at the circled elements, X_C, and at squared elements, X_S. For each $x \in X_C$, set $P^2_x = P_x - \beta$, and for each $x \in X_S$, set $P^2_x = P_x + \beta$, where $\beta > 0$ is the largest number that still satisfies all constraints in $\mathcal{H}$. By construction, $P^2_S = P_S$ for all $S \in \overline{\mathcal{H}}_1 \cup \overline{\mathcal{H}}_2$. Observation (1) then ensures that $P^2_S = P_S$ for all integral set $S \in \mathcal{H}$.

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

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

Appendix B. Proofs of Lemma 1 and Theorem 2

Proof of Lemma 1. Suppose for contradiction that $\mathcal{H}$ is BvN decomposable. Consider a random assignment P specified by

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

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 $i \in \{1, \ldots, k\}$. Since $\mathcal{H}$ is BvN decomposable, there exist $P^1, P^2, \ldots, P^K$ and $\lambda^1, \lambda^2, \ldots, \lambda^K$ such that

(1) $P = \sum\limits_{k=1}^{K} \lambda^k P^k$,
(2) $\lambda^k \in (0, 1]$ for all k and $\sum_{k=1}^{K} \lambda^k = 1$,
(3) $P^k_S \in \{\lfloor P_S \rfloor, \lceil P_S \rceil\}$ for all $k \in \{1, \ldots, K\}$ 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 BvN 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 BvN 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 BvN 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 BvN 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 C. Proof of Theorem 3

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

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

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

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

$$\sum_{(i,a) \in S} P'_{ia}, \sum_{(i,a) \in S} P''_{ia} \in \left\{ \left\lfloor \sum_{(i,a) \in S} P_{ia} \right\rfloor, \left\lceil \sum_{(i,a) \in S} P_{ia} \right\rceil \right\} \quad \text{for all } S \in \mathcal{H}', \tag{5}$$

for any integer-valued matrices P' and P'' that are part of the decomposition. In particular, property (5) 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 $l' = 1, ..., \bar{l}/2$.

  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 $l' = 1, ..., m-1$ 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

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

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

Appendix D. 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" $\emptyset$ such that $v_{i\emptyset} = 0$ for all i and define $P^*_{i\emptyset} = \bar{q}_{\{i\} \times O} - P^*_{\{i\} \times O}$ for all i. With these technical definitions Theorem 3 is applicable to $(P^*_{ia})_{(i,a) \in N \times (O \cup \{\emptyset\})}$, 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 (3), and

$$\sum_a P'_{ia} v_{ia} \geq \sum_a P^*_{ia} v_{ia} - \overline{v}_i, \tag{6}$$

for each i.[12] Since $\sum_a P^*_{ia} v_{ia} \geq \omega^*$ for each i by construction, inequality (6) implies that $\sum_a P'_{ia} v_{ia} \geq \omega^* - \overline{v}_i$, implying $\omega' \geq \omega^* - \max_{i \in N} \overline{v}_i$. Finally, $w^* \geq w^{**}$ since w^* is the optimal value of a less constrained problem than problem (3). Thus we have $\omega' \geq \omega^{**} - \max_{i \in N} \overline{v}_i$, completing the proof. □

References

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

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 (2003): "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.

Budish, E. (2008): "A New Solution to the Course-Allocation Problem: Approximate Competitive Equilibrium from Approximately Equal Incomes," Harvard University, Unpublished mimeo.

Budish, E., and E. Cantillon (2008): "Strategic Behavior in Multi-unit Assignment Problems: Theory and Evidence from Course Allocations," Unpublished mimeo.

Budish, E., Y.-K. Che, F. Kojima, and P. Milgrom (2009): "Implementing Random Assignments, Part II: ***," Unpublished mimeo.

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

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

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

Hylland, A., and R. Zeckhauser (1979): "The Efficient Allocation of Individuals to Positions," *Journal of Political Economy*, 87, 293–314.

Kojima, F. (2008a): "Games of school choice under the Boston mechanism with general priority structures," *Social Choice and Welfare*, 31, 357–365.

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

Kojima, F., and M. Manea (2008): "Incentives in the Probabilistic Serial Mechanism," mimeo.

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

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.

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

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

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.

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.