Review of Economic Studies (2020) **0**, 1–38 doi:10.1093/restud/rdz066

Advance access publication 6 January 2020

Approximate Random Allocation Mechanisms

MOHAMMAD AKBARPOUR
Graduate School of Business, Stanford University

and

AFSHIN NIKZAD
Department of Economics, University of Southern California

First version received March 2017*; Editorial decision November* 2019*; Accepted December* 2019 (*Eds.*)

We generalize the scope of random allocation mechanisms, in which the mechanism first identifies a feasible "expected allocation" and then implements it by randomizing over nearby feasible integer allocations. The previous literature has shown that the cases in which this is possible are sharply limited. We show that if some of the feasibility constraints can be treated as goals rather than hard constraints, then, subject to weak conditions that we identify, any expected allocation that satisfies all the constraints and goals can be implemented by randomizing among nearby integer allocations that satisfy all the hard constraints exactly and the goals approximately. By defining *ex post* utilities as goals, we are able to improve the *ex post* properties of several classic assignment mechanisms, such as the random serial dictatorship. We use the same approach to prove the existence of ϵ-competitive equilibrium in large markets with indivisible items and feasibility constraints.

Key words: Market design, Matching, Random allocation, Intersecting constraints.

JEL Codes: C78, D47, D82.

1. INTRODUCTION

When cash transfers are limited and goods are indivisible, it can sometimes be impossible to allocate goods in an efficient and envy-free ("fair") way. This challenge is faced, for example, when assigning students to courses, resettling refugees, or setting a competitive sports schedule. Early economic studies of this problem by Hylland and Zeckhauser ("HZ") and Bogomolnaia and Moulin ("BM") assume that each agent must receive just a single good and show that it is then possible to allocate the probabilities of receiving each good in an efficient, envy-free manner (Hylland and Zeckhauser, 1979; Bogomolnaia and Moulin, 2001). Budish, Che, Kojima, and Milgrom ("BCKM") propose expanding this approach to a wider set of multi-item allocation problems in which the constraints may be more complex than merely a set of one-item-per-person constraints (Budish *et al.*, 2013). For example, in course allocation, a student may wish to have at least one class in science and one in humanities in a particular term. They apply the result from combinatorial optimization that for any expected allocation that satisfies all the constraints, if the

The editor in charge of this paper was Christian Hellwig.

constraints have a particular "bihierarchy" structure, then the expected allocation can always be achieved by randomizing among pure allocations in which each fractional expected allocation is rounded up or down to an adjacent integer and all the constraints are simultaneously satisfied (Edmonds, 2003). When the conditions are satisfied, BCKM show that this sometimes makes it possible to use mechanisms that select efficient, envy-free expected allocations and to implement those through randomization.

However, as BCKM also state, the bihierarchy condition can be a necessary condition, and so even their expansion of the previous works can rule out some potential applications. For instance, the condition is violated in school choice if a school with limited capacity has at least two of the walk-zone, gender, or racial diversity constraints.

The goal of this article is to generalize this approach to a much broader class of allocation problems by reconceptualizing the role of constraints. Our analysis shows that many more constraints can be managed if some of them are "soft," in the sense that they can bear small errors with relatively small costs. More precisely, we partition the full set of constraints into a set of *hard* constraints that must always be satisfied exactly, and a set of *soft* constraints that should be satisfied approximately. The main theorem of the article identifies a rich constraint structure that is approximately implementable, meaning that if an expected allocation satisfies all the constraints, then it can be implemented by randomizing among pure allocations that satisfy all the hard constraints and satisfy the soft constraints with only small errors.

The importance of this result arises from the way it can expand potential applications by breaking the theoretical barrier of implementing intersecting constraints, which we do by designing a general framework that models them as "goals." For example, in the school choice setting, the requirement that each student must be assigned to exactly one school is (in our conception) a hard constraint that must be satisfied, but the requirement that some fraction of students in a school live in the walk zone may be a soft constraint. Allowing this flexibility is particularly important when the constraints are inconsistent, and in other cases it provides greater scope for accommodating individual student preferences.

1.1. *Model and contributions*

In this article, we analyse a general model of matching with indivisible objects. Section 2 introduces the building blocks of our model. In Section 2.1, we propose a new notion of approximate implementation. A constraint is approximately satisfied if the probability of violating that constraint is exponentially decreasing in the size of the constraint.[1] We partition the set of constraints into a set of hard constraints that are inflexible and a set of soft constraints that are flexible, and we call it a *hard–soft partitioned* constraint set. We say that a hard–soft partitioned constraint set is *approximately implementable* if for any feasible fractional assignment that satisfies both hard and soft constraints, there exists a lottery over pure assignments such that the following three properties hold: (1) the expected value of the lottery is equal to the fractional assignment, (2) the outcome of the lottery satisfies hard constraints, and (3) the outcome of the lottery satisfies soft constraints approximately.[2] The question that we ask is: what kinds of hard–soft partitioned constraint structures are approximately implementable?

The main theoretical contribution of the article is stated in Theorem 1. The theorem identifies a rich structure for soft constraints under which the whole structure is approximately implementable,

1. For instance, if a school has a capacity for 1,000 students and half of them should come from the walk zone, then the size of this capacity constraint is 1,000 and the size of the walk-zone constraint is 500.

2. Quantitatively, a soft constraint of size μ is approximately satisfied if the probability of violating the constraint by more than $\epsilon\%$ is less than $e^{-\mu\epsilon^2/3}$ for an upper quota constraint and less than $e^{-\mu\epsilon^2/2}$ for a lower quota constraint.

given that the structure of hard constraints is the same maximal structure introduced in BCKM—the "bihierarchical" structure. The structure we identify allows any set of (possibly intersecting) soft constraints, as long as adding each individual soft constraint to the set of hard constraints preserves its bihierarchical structure. We complement this theorem by showing that our bounds on the approximation errors are tight.

We prove Theorem 1 by constructing a matching algorithm which approximately implements any feasible fractional assignment. The proof is sketched in Section 3.2. At the core of our proof is a matrix operation—*Operation* $\mathcal{X}$—that takes a fractional assignment as its input and (randomly) generates another assignment with fewer fractional elements as its output. By iterative applications of Operation $\mathcal{X}$, an integral assignment is generated.[3] The (random) assignment matrix satisfies the martingale property, *i.e.*, the expected value of the assignment matrix after the next iteration remains the same as its current value. We apply probabilistic concentration bounds to our randomized mechanism in order to prove that soft constraints are satisfied with small errors. It is worth mentioning that the previous literature on the implementation of fractional assignments relies on the Birkhoff–von Neumann theorem (Birkhoff, 1946; Von Neumann, 1953) (in HZ and BM) or its generalizations, such as (Edmonds, 2003) (*e.g.* the implementation method of BCKM is based on a theorem of Edmonds on deterministic rounding of mathematical programs). Our article, on the other hand, develops an implementation method by building on techniques from the randomized rounding literature (Ageev and Sviridenko, 2004; Gandhi *et al.*, 2006).

Our theoretical results reveal that there are trade-offs between the complexity of the set of constraints and the quality of the error bounds. On the one hand, in Section 3.4, we show that for sufficiently simple ("hierarchical") set of hard constraints, our mechanism implements *any* arbitrary set of soft constraints, with no compromise in the quality of the error bounds. On the other hand, we show that if one insists on the *bi*hierarchical structure of hard constraints, our mechanism implements any arbitrary set of soft constraints at the expense of weaker error bounds.

In light of the tightness result for the general environment considered in Theorem 1, one might ask: is it possible to prove stronger bounds by considering a simpler economic environment? Our second theorem considers a setting with agent *types*. We say two agents have the same type if the set of constraints imposed on them is the same. Theorem 2 shows that a modified version of our allocation mechanism guarantees that none of the soft constraints would be violated with more than an additive, deterministic error equal to the number of agent types.

We close the discussion of our bounds by stating a caveat: our theoretical bounds for the violation of soft constraints are weak for "small" constraints. For instance, consider a school with capacity for 250 students. Theorem 1 guarantees that the probability of admitting more than 275 students (a 10% violation) is less than 0.43, which is hardly a guarantee. That said, one should note that these are worst-case bounds proved for *all* possible problem instances. In Section 3.6, we investigate the empirical performance of our mechanism for a typical school choice environment by running simulations in a setting similar to the NYC high schools. For the same constraint with size 250, simulations show that the empirical probability of admitting more than 275 students is less than 0.064. The bounds improve with the size of the constraint. For instance, for a school with capacity 500, the theoretical and empirical bounds for a 10% violation reduce to 0.19 and 0.024, respectively.

In Section 4, we discuss the applications of our framework in implementing intersecting constraints in the school choice setting. In particular, we introduce a new method to accommodate walk-zone priorities in the school choice. Many school choice systems handle walk-zone constraints by requiring schools to dedicate a specific fraction of their seats to students within

3. It is worth mentioning that the matching algorithm stops in *polynomial* time, which is an important requirement for practical matching algorithms in relatively large markets.

their "walk zone." This means that the lottery has a "discontinuity" issue, since it treats two students who are a few blocks away, but on the two sides of a walk-zone border, very differently. Our framework, however, allows for a design that treats students in a "continuous" manner with respect to their *distances* from the schools.

The rest of the article explores the theoretical applications of our framework. In Section 5, we address the issue that even if a constructed fractional assignment is fair, there could be very large discrepancies in *realized* utilities, as discussed in Kojima (2009). We show that when a fractional assignment is implemented via Theorem 1, an agent's *ex post* utility is approximately equal to her *ex ante* utility. We then provide two examples of how our utility guarantees can be applied to two classic allocation mechanisms: the random serial dictatorship (RSD) mechanism and the pseudo-market mechanism. We improve these mechanisms by incorporating intersecting soft constraints, as well as providing approximate guarantees for the agents' *ex post* utilities in settings with such constraints.

In our next application in Section 6, we prove the existence of ϵ-competitive equilibrium (ϵ-CE)[4] in a market with indivisible objects and distributional constraints. In our environment, each agent is allowed to impose some (possibly intersecting) constraints on her allocation. This can be applied to, for instance, an online advertisement setting where multiple advertisers are buying impressions, who prefer to diversify the set of their audience. Moreover, the methods we develop to prove the existence can also find an ϵ-CE with high probability for arbitrary small ϵ, provided that it has access to a solver that finds δ-CE in markets with divisible items, for sufficiently small δ.

1.2. *Related work*

Randomization is commonplace in everyday life and has been studied in various settings such as school choice, course allocation, and house allocation (Abdulkadiroğlu and Sönmez, 1998; Abdulkadiroğlu *et al.*, 2005; Budish, 2011; Pathak and Sethuraman, 2011). Perhaps, the most practically popular random mechanism is to draw a fair random ordering of agents and then let the agents select their most favourite object (among those remaining) according to the realized random ordering without violating the constraints. This mechanism, which is known as RSD is a desirable mechanism, as it is strategy-proof and *ex post* Pareto efficient (Abdulkadiroğlu and Sönmez, 1998; Chen and Sonmez, 2002). Nevertheless, RSD is *ex ante* inefficient, *ex post* (highly) unfair, and cannot handle lower quotas (Bogomolnaia and Moulin, 2001; Hatfield, 2009; Kojima, 2009). Several papers compare PS and RSD and analyse their connections in large markets (Manea, 2009; Kojima and Manea, 2010; Che and Kojima, 2010; Liu and Pycia, 2016).

The idea to construct a fractional assignment and then implement it by a lottery over pure assignments was first introduced in Hylland and Zeckhauser (1979) for cardinal utilities. Bogomolnaia and Moulin (2001) construct a mechanism, the *Probabilistic Serial Mechanism* (PS), for ordinal utilities based on the same technique. Both papers model one-to-one matching markets with no other constraints. Hashimoto (2018) shows that an infinite-market mechanism can be asymptotically approximated by a finite-market mechanism that keeps feasibility, *ex post* individual rationality, and *ex post* incentive compatibility. He uses the generalized random priority mechanism as the approximating mechanism, and applies his method to approximate an extension of the pseudo-market mechanism of HZ where there is a continuum of agents with multi-unit demands. There, the primary focus is on feasibility with no intersecting constraints (no additional

4. An ϵ-equilibrium in an indivisible objects setting is a vector of prices and a partition of objects in which all agents' utilities are at least $(1-\epsilon)$ of their utilities in the competitive equilibrium if objects were divisible, no agent's budget constraint is violated, and the market clears.

seats to students) and strategy-proofness (*ex post* incentive compatibility). The approximated competitive equilibrium from equal incomes (CEEI) mechanism is exactly feasible and exactly strategy-proof, and efficiency and envy-freeness are achieved only in approximate senses. Budish *et al.* (2013) build on those two papers by considering a richer constraint structure.[5] Our article generalizes this literature by designing a randomized mechanism which can accommodate a much richer class of constraints.

The literature takes different approaches for accommodating constraints in assignment problems. There is work that treats constraints as hard or flexible. They consider constraints such as distributional constraints or constraints such as stability and strategy-proofness. We review some of this work below.

Fragiadakis and Troyan (2017) consider hard distributional constraints in stable assignment problems. They introduce a mechanism that exploits the submitted preferences and, in the case of finding a solution, respects all distributional constraints. Nguyen and Vohra (2019) consider the problem of finding stable matchings in the presence of proportionality constraints and design an algorithm which finds stable matchings while treating the proportionality constraints as "soft" constraints. Kamada and Kojima (2015, 2019) observe that under distributional constraints existing matching mechanisms typically suffer from inefficiency and instability. In the former work, they propose a mechanism that performs better in terms of efficiency, stability, and incentives, while respecting the distributional constraints. In the latter work, they relax stability and focus on feasible, individually rational, and fair assignments. They characterize the class of constraints on individual schools under which a student-optimal fair matching exists.

The approximate satisfaction of constraints has been studied in a few recent papers. Budish (2011) studies the problem of combinatorial assignment by introducing a notion of approximate CEEI, which treats course capacities as flexible constraints. A "soft bound" approach is introduced in Ehlers *et al.* (2014), where the authors introduce a deferred acceptance algorithm with soft bounds in which they adjust group-specific lower and upper bounds to achieve a fair and non-wasteful mechanism. Nguyen *et al.* (2016) and Nguyen and Vohra (2018) respectively study one- and two-sided matching markets with complementarities. They accommodate complementarities in the agents' preferences in exchange for bounded violations of the capacity constraints. In a work subsequent to ours, Ashlagi *et al.* (2019) consider RSD under distributional constraints. They adopt our model with agent types (Section 3.5) and design a variation of RSD with dynamic menus which finds an assignment that approximately satisfies the distributional constraints. Recently, Che *et al.* (2019) have shown that accommodating complementarities is possible when finding stable assignments in many-to-one large matching markets, given that the firms' choice functions satisfies mild continuity and convexity assumptions.

Notions such as stability, incentive compatibility, or efficiency can also be seen as constraints to be satisfied by an assignment mechanism. Che and Tercieux (2019) observe that when agents' preferences are correlated over objects, standard mechanisms such as deferred acceptance and top trading cycles are either inefficient or unstable, even asymptotically. Then, they propose a new variant of deferred acceptance that is asymptotically efficient, asymptotically stable, and asymptotically incentive compatible. In a related work, Liu and Pycia (2016) focus on ordinal mechanisms in which no small group of agents can substantially change the allocations of

5. In a recent work, Pycia and Ünver study a more general structure (the *Totally Unimodular* or TU structure) and show that they can accommodate constraints such as strategy-proofness and envy-freeness as linear constraints as long as they fit into the TU structure (Pycia and Ünver, 2015). Our approach is conceptually different from theirs since we consider flexible constraints (*i.e.* goals) which may not fit into the TU structure. Kesten *et al.* (2017) also work with fractional assignments and improves RSD.

others, and show that all asymptotically efficient, symmetric, and asymptotically strategy-proof mechanisms lead to the same allocations in large markets.

There are some key points that separate our article from these works. First, we propose a framework which can handle "intersecting" constraints. For instance, in the school choice setting, we can accommodate racial, gender, and walk-zone priority constraints simultaneously. Second, we provide a rich language for the market maker to declare a partitioned constraint set, which contains both flexible and inflexible constraints. Third, our mechanism runs in polynomial time, whereas the approach introduced in Budish (2011), as discussed in Budish *et al.* (2016) and Rubinstein (2014), is computationally hard.[6]

Compared to BCKM, who build their implementation method based on a theorem of Edmonds on deterministic rounding of mathematical programs, we build our implementation method based on randomized rounding. Various rounding techniques have been developed in the computer science literature; Ageev and Sviridenko (2004), Gandhi *et al.* (2006), and Chekuri *et al.* (2010) are among the closest to our work. Ageev and Sviridenko (2004) introduce a deterministic rounding method, called *pipage rounding*, and Gandhi *et al.* (2006) and Chekuri *et al.* (2010) design rounding methods following the same idea, although in a randomized fashion and for different applications. We remark that none of these methods could be used directly to handle our application, *i.e.,* a bihierarchical constraint structure with upper and lower quotas. We design our implementation method by extending the approach of Gandhi *et al.* (2006) to bihierarchical structures. The techniques in Gandhi *et al.* (2006)—though they inspired our design—are specifically designed for the job scheduling problem. As a result, their randomized algorithms accommodate neither non-local soft constraints, nor (bi)hierarchical hard constraints.

Other rounding methods have been used in the literature for (approximately) implementing fractional allocations. Nguyen *et al.* (2016) and Nguyen and Vohra (2018) model matching markets with complementarities. They use *iterative rounding* (Lau *et al.*, 2011) to design implementation schemes specific to their problem structure. The goal there is to handle complementarities (in a setup with only capacity constraints), while our article is concerned with implementing generalized constraint structures, and not with complementarities.[7]

The problem of reduced-form implementation in the auction literature is also related to our work (Matthews, 1984; Border, 1991; Che *et al.*, 2013). In this problem, an *interim* allocation, which describes the marginal probabilities of each bidder obtaining the good as a function of his type, is constructed. Then, as we do in our problem, they ask which interim allocations can be implemented by a lottery over feasible pure allocations. The approximate satisfaction of constraints, however, is not studied in that literature.

2. SETUP

Consider an environment in which a finite set of *objects* O has to be allocated to a finite set of N. We denote the set of agent–object pairs, $N \times O$, by E, where each $(n,o) \in E$ is an *edge*.[8] Sometimes we use "e" to denote edges. A *pure assignment* is defined by a non-negative matrix $X = [X_{no}]$, where each $X_{no} \in \{0,1\}$ denotes the amount of object o which is assigned to agent n for all $(n,o) \in E$. We require the matrix to be binary valued to capture the indivisibility of the objects.

6. Bronfman (2018) also use an approximation approach and propose a polynomial time algorithm to solve the problem of couples in the Israeli Medical Match problem.

7. The specific structure of Nguyen *et al.* (2016) allows them to provide small additive bounds on the violation of capacity constraints by using techniques different than ours. We also show that under certain structures, our technique can provide small additive error bounds (Section 3.5).

8. We use the term *edge* since, in the graph-theoretical representations of the problem, the share of an object assigned to an agent is shown by a weighted edge in the graph.

A *block* $B \subseteq E$ is a subset of edges. A *constraint* S is a triple $(B, \underline{q}_B, \bar{q}_B)$, which is a block B associated with a vector of integer *quotas* $(\underline{q}_B, \bar{q}_B)$ as the lower and upper quotas on B. A *structure* is a subset $\mathcal{E} \subseteq 2^E$; *i.e.*, a collection of blocks. A *constraint set* is a set of constraints. Let $\mathbf{q} = [(\underline{q}_B, \bar{q}_B)_{B \in \mathcal{E}}]$.

We say that X is *feasible with respect to* $(\mathcal{E}, \mathbf{q})$ (or simply, with respect to $\mathcal{E}$ when $\mathbf{q}$ is clearly known from the context) if

$$\underline{q}_B \leq \sum_{e \in B} X_e \leq \bar{q}_B \qquad \forall B \in \mathcal{E}. \tag{2.1}$$

We call a block $B \in \mathcal{E}$ *agent n's capacity block* when $B = \{X_{nj} | j \in O\}$. Similarly, we call a block $B \in \mathcal{E}$ an *object m's capacity block* when $B = \{X_{im} | i \in N\}$. We sometimes refer to the capacity blocks of agents and objects as *row blocks* and *column blocks*, respectively. A *capacity constraint* is a constraint $(B, \underline{q}_B, \bar{q}_B)$, where $B \in \mathcal{E}$ is a capacity block. We sometimes refer to the capacity constraints of agents and objects as *row constraints* and *column constraints*, respectively.

A *fractional assignment* is defined by a matrix $x = [x_{no}]$, where each $x_{no} \in [0, 1]$ is the quantity of object o assigned to agent n. To distinguish between pure and fractional assignments, we usually use X to denote a pure assignment and x for a fractional assignment. We sometimes use the term *expected assignment* to address fractional assignments. For any (pure or fractional) assignment x, we use x_n to denote the vector $(x_{n1}, \ldots, x_{n|O|}) \in \mathbb{R}^{|O|}$, *i.e.*, x_n denotes the allocation of agent n.

Given a structure $\mathcal{E}$ and associated integer quotas $\mathbf{q}$, a fractional assignment matrix x is *implementable under quotas* $\mathbf{q}$ if there exist positive numbers $\lambda_1, \ldots, \lambda_K$, which sum up to one, and pure assignments $X_1, \ldots, X_K$, which are feasible under $\mathbf{q}$, such that

$$x = \sum_{i=1}^{K} \lambda_i X_i.$$

We also say that a structure $\mathcal{E}$ is *universally implementable* if, for any quotas $\mathbf{q} = (\underline{q}_B, \bar{q}_B)_{B \in \mathcal{E}}$, every fractional assignment matrix satisfying $\mathbf{q}$ is implementable under $\mathbf{q}$.

The existing theoretical result on the implementability of a structure, which is discussed in BCKM's paper (Budish *et al.*, 2013), builds on a classic combinatorial optimization result (Edmonds, 2003). It shows that the *bihierarchy* is a sufficient condition for the universal implementability of a structure. More precisely, a structure $\mathcal{H}$ is a *hierarchy* if for every pair of blocks B and B' in $\mathcal{H}$, we have that $B' \subset B$ or $B \subset B'$ or $B \cap B' = \emptyset$. A simple hierarchy is depicted in Figure 1. A structure $\mathcal{H}$ is a *bihierarchy* if there exists two hierarchies $\mathcal{H}_1$ and $\mathcal{H}_2$ such that $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$. The following theorem identifies a sufficient and almost necessary condition under which a structure is universally implementable.

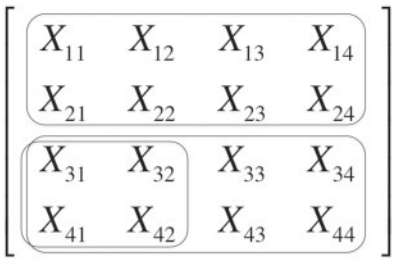

FIGURE 1
A hierarchy.

Theorem 0 (BCKM) *If a structure $\mathcal{H}$ is a bihierarchy, then it is universally implementable. In addition, if $\mathcal{H}$ contains all agents and objects capacity blocks, then it is universally implementable if and only if it is a bihierarchy.*

2.1. *Approximate implementation*

In many assignment problems, the involved constraints are intersecting and the bihierarchy assumption fails. The following example clarifies the bihierarchy limitations in the school choice setting.

Example 1 *In the Boston School Program (as of January 2016), 50% of each school's seats were set aside for walk-zone priority students. Consider a school which also has a group-specific quota on low socioeconomic status (SES) students. Together with the requirement that each student should be assigned to one school, these blocks do not form a bihierarchy.*

In this article, we show that by treating some constraints as *goals*, rather than inflexible constraints, we can accommodate many more constraints. More precisely, we ask the market maker to partition the full set of constraints into a set of hard constraints that must be satisfied exactly and a set of soft constraints that must be satisfied approximately. Accordingly, the constraint structure will be partitioned into two sets: a set of *hard* blocks, $\mathcal{H}$, which are blocks of inflexible constraints, and a set of *soft* blocks, $\mathcal{S}$, which are blocks of flexible constraint. We refer to $\mathcal{E}=\mathcal{H}\cup\mathcal{S}$ as a *hard–soft partitioned structure*, or simply a partitioned structure.

Another way in which our framework generalizes BCKM is that in our model elements of soft constraints can have arbitrary weights; that is, for a soft block B', we say X is feasible with respect to B' if

$$\underline{q}_{B'} \leq \sum_{e\in B'} w_e X_e \leq \bar{q}_{B'},$$

where w_e can take any arbitrary value in $[0,1]$, and $\underline{q}_{B'}$ and $\bar{q}_{B'}$ can be any non-negative real number. The weights associated with an edge need not be equal for all blocks. Recall that, similar to BCKM, for a hard block B, we require $w_e=1$ for all $e\in B$ and restrict $\underline{q}_B$ and $\bar{q}_B$ to be integers. This generalization of weights expands the scope of practical applications of the model, as discussed in Section 4.

Our goal in this article is to identify structural conditions imposed on $\mathcal{H}$ and $\mathcal{S}$ under which $\mathcal{E}=\mathcal{H}\cup\mathcal{S}$ is "approximately implementable." In the following, we rigorously define the notion of approximate implementation.

Definition 1 *Given a hard–soft partitioned structure $\mathcal{E}=\mathcal{H}\cup\mathcal{S}$, we say $\mathcal{E}$ is* Approximately Implementable *if for any vector of quotas $\mathbf{q}$ and any expected assignment x which is feasible with respect to $(\mathcal{E},\mathbf{q})$, there exists a lottery (probability distribution) over pure assignments $X_1,\ldots,X_K$ such that, if we denote the outcome of the lottery by the random variable X, the following properties hold:*

P1. **Assignment preservation:** $\mathbb{E}[X]=x$.
P2. **Exact satisfaction of hard constraints:** *All constraints in $\mathcal{H}$ are satisfied.*

P3. **Approximate satisfaction of soft constraints:** *For any soft block $B \in \mathcal{S}$, any set of weights $\{w_e : e \in B, w_e \in [0,1]\}$ with $\sum_{e \in B} w_e x_e = \mu$, and for any $\epsilon > 0$, we have*

$$\Pr(\mathsf{dev}^+ \geq \epsilon\mu) \leq e^{-\mu\frac{\epsilon^2}{3}} \tag{2.2}$$

$$\Pr(\mathsf{dev}^- \geq \epsilon\mu) \leq e^{-\mu\frac{\epsilon^2}{2}} \tag{2.3}$$

where dev^+ *and* dev^- *are defined as follows:*

$$\mathsf{dev}^+ = \max\left(0, \sum_{e \in B} w_e X_e - \mu\right),$$

$$\mathsf{dev}^- = \max\left(0, \mu - \sum_{e \in B} w_e X_e\right).$$

Property 1 simply states that there exists a lottery which implements x. Property 2 states that hard constraints are satisfied with no error. Property 3 defines our notion of approximation in a fashion similar to Chernoff concentration bounds.[9] By this property, the probability of violating a soft constraint by a factor greater than ϵ decays exponentially with the right-hand side (or the left-hand side) of the constraint. Property 3 also guarantees that the probability of violating soft constraints by a multiplicative factor ϵ exponentially decays with ϵ. For example, in a school with $2,000$ seats, the probability of admitting more than $2,100$ students is bounded above by 0.19, while the probability of admitting more than $2,200$ students is no more than 0.0012.

The probabilistic bounds of Definition 1 might not seem practical for small markets. We address this concern in Section 3.6. One may also wonder why implementation in our setting is a non-trivial problem, and why simple implementation approaches fail. We discuss this issue in Supplementary Appendix E.

3. THE MAIN THEOREMS

Given a partitioned structure $\mathcal{E} = \mathcal{H} \cup \mathcal{S}$, we first identify structures for $\mathcal{H}$ and $\mathcal{S}$ under which $\mathcal{E}$ is approximately implementable in the sense of Definition 1. We then state a generalized version of our main theorem and show that given a *bi*hierarchy of hard constraints, *any* soft constraint can be approximately satisfied, but with a weaker notion of approximate satisfaction. Finally, in Section 3.5, under more specific constraint structures, we provide more powerful bounds that are additive.

First of all, note that Theorem 0 shows that even if $\mathcal{S} = \emptyset$ (*i.e.* there are no soft constraints), in order for $\mathcal{E}$ to be universally implementable, bihierarchy is a sufficient and almost necessary condition. We use the term "almost" because, while bihierarchy is not a necessary condition for universal implementability in general, it is necessary in the presence of all agents' and objects' capacity blocks, as noted by Budish *et al.* (2013).[10] We maintain this maximal structure and let hard blocks form a bihierarchy; *i.e.*, we assume $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$, where $\mathcal{H}_1$ and $\mathcal{H}_2$ are two hierarchies. Then, given a bihierarchical hard structure, we aim to identify a structural condition, if any, for soft blocks $\mathcal{S}$ under which $\mathcal{E} = \mathcal{H} \cup \mathcal{S}$ is approximately implementable. It is worth pointing out that when $\mathcal{H}$ is a bihierarchy, a fully general set of soft constraints is not approximately implementable (as shown in Appendix B.4).

9. Chernoff bounds are explained in Supplementary Appendix D.

10. While there are more general sufficient conditions for universal implementability (*e.g. Total Unimodularity* of the coefficient matrix of the linear constraints (Schrijver and Cook, 1997)), these conditions are more abstract and convey little intuition about the structural properties of the constraints.

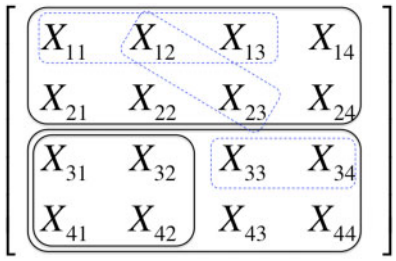

FIGURE 2
The solid blocks form a hierarchy $\mathcal{H}_1$. The dashed blocks are in the deepest level of $\mathcal{H}_1$. A block that, for example, contains X_{32} and X_{33} is not in the deepest level of $\mathcal{H}_1$.

3.1. *The structure of soft blocks*

Now we will show that if $\mathcal{H}$ forms a bihierarchy, there exists a rich structure for the soft blocks $\mathcal{S}$ under which $\mathcal{E}=\mathcal{H}\cup\mathcal{S}$ is approximately implementable. To do so, we need to define one new concept. For a block $B\in\mathcal{S}$, we say that B is in the *deepest level of* $\mathcal{H}_1$ if for any block $C\in\mathcal{H}_1$, either $B\subseteq C$ or $B\cap C=\emptyset$. (See Figure 2 for an illustration.) We also say that $B\in\mathcal{S}$ is in the *deepest level of a bihierarchy* $\mathcal{H}=\mathcal{H}_1\cup\mathcal{H}_2$ if it is in the deepest level of either of $\mathcal{H}_1$ or $\mathcal{H}_2$.

Theorem *1* (The main theorem) *Let $\mathcal{E}=\mathcal{H}\cup\mathcal{S}$ be a hard–soft partitioned structure such that $\mathcal{H}$ is a bihierarchy and any block in $\mathcal{S}$ is in the deepest level of $\mathcal{H}$. Then, $\mathcal{E}$ is approximately implementable.*

3.2. *Proof overview for Theorem 1*

We present an overview of the proof here. The full proof is in Appendix A. The proof is constructive. We propose a randomized mechanism that, given a partitioned structure satisfying the properties described in Theorem 1, approximately implements a given feasible fractional assignment. To do so, let us define a constraint to be *tight* if it is binding, and to be *floating* otherwise. This definition also applies to the implicit constraints $0\leq x_e\leq 1$ for all $e\in E$. We say an edge e is a *floating edge* if $0<x_e<1$. A block associated with a tight constraint is a *tight block*.

The core of our randomized mechanism is a probabilistic operation that we design, called *Operation* $\mathcal{X}$. We iteratively apply Operation $\mathcal{X}$ to the initial fractional assignment until a pure assignment is generated. At each iteration t, the fractional assignment x_t is converted to x_{t+1} in a way such that the following properties are satisfied:

1. The number of floating constraints decreases,
2. $\mathbb{E}(x_{t+1}|x_t)=x_t$, and
3. x_{t+1} is feasible with respect to $\mathcal{H}$.

The first property guarantees that after a finite (and small) number of iterations,[11] the obtained assignment is pure. The second property ensures that the resulting pure assignment is equal to the original fractional assignment *in expectation*. The third property guarantees that all hard constraints are satisfied throughout the whole process of the mechanism. We will also be able to

11. Our randomized mechanism stops after at most $|\mathcal{H}|+|E|$ iterations.

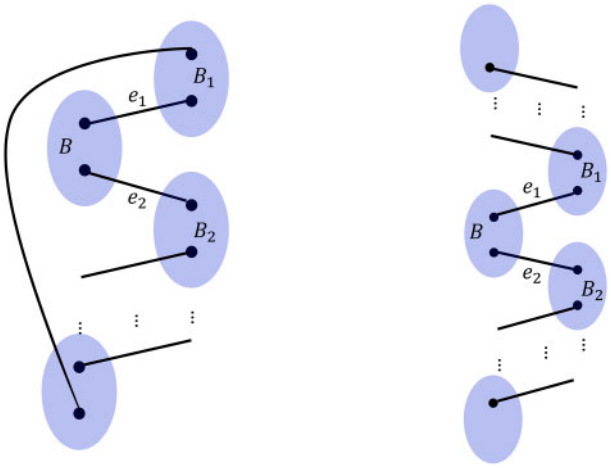

FIGURE 3
Illustration of a floating cycle on the left and a floating path on the right.

show that the soft constraints are approximately satisfied at the end of the iterative process. This will follow from the properties of Operation $\mathcal{X}$.

Operation $\mathcal{X}$ has two steps: first it finds a subset of edges with a special structure, a *floating path* or a *floating cycle*. In the second step, the floating path or cycle is changed in a way that the assignment "gets closer" to a pure assignment.

First step If there is a floating edge that is not part of any tight block, then choose that edge as a floating path and start the second step. Otherwise, there must exist a tight block that contains at least one floating edge. Consider the smallest possible such block, namely B. Without loss of generality, suppose that $B \in \mathcal{H}_1$. Since B is tight and the quotas are integers, then there must exist at least 2 floating edges inside B, namely e_1, e_2.

If neither e_1 nor e_2 are part of a tight block in $\mathcal{H}_2$, then choose e_1, e_2 as a floating path and start the next step. If both e_1 and e_2 are part of some tight block in $\mathcal{H}_2$, let B_i be the smallest possible tight block in $\mathcal{H}_2$ that contains e_i, for $i = 1, 2$. Since quotas are integers, and since B_i is tight, then it must contain another floating element, namely e_i'. The idea is to continue this search from both directions until we return to one of the blocks that we have previously visited, which gives us a *floating cycle* (Figure 3), or until we find floating edges on both sides of the search that are part of no tight constraint, which gives us a *floating path* (Figure 3). In the remaining case where exactly one of e_1, e_2 is part of a tight block in $\mathcal{H}_2$, we can find a floating cycle or path through a similar procedure.

Second step Once we identify a floating cycle or a floating path of a fractional assignment x, Operation $\mathcal{X}$ stochastically changes the assignment x to a new assignment x', in the way we define next. If neither a floating cycle nor a floating path exists, then the assignment must be pure. (See Lemma A.4 in the Appendix.)

To define Operation $\mathcal{X}$, we need some new notations. Suppose that we are given a fractional assignment x. For any block B and any $\epsilon > 0$, let $x \uparrow_\epsilon B$ denote a new (fractional) assignment in which x_e is changed to $x_e + \epsilon$ for all $e \in B$, and remains unchanged otherwise. Similarly, let $x \downarrow_\epsilon B$ denote the fractional assignment in which x_e is changed to $x_e - \epsilon$ if $e \in B$, and remains unchanged otherwise. Therefore, $(x \uparrow_\epsilon B) \downarrow_\epsilon B'$ denotes the fractional assignment in which the value of any

edge $e \in B - B'$ is $x_e + \epsilon$, the value of any edge $e \in B' - B$ is $x_e - \epsilon$, and the value of any other edge is the same as its value in x.

We now define Operation $\mathcal{X}$ for a given floating cycle. The definition for a floating path is similar. Let $F = \langle e_1, \ldots, e_l \rangle$ be a floating cycle in x. We first partition F into two subsets:

$$F_o = \{e_i : i \text{ is odd}\},$$
$$F_e = \{e_i : i \text{ is even}\}.$$

Given an assignment x, a floating cycle F, and two non-negative reals ϵ and ϵ' (which we will describe how to set), the output of Operation $\mathcal{X}$ is an assignment $x' \in \mathbb{R}^{N \times O}$, where:

$$x' = \begin{cases} (x \uparrow_\epsilon F_o) \downarrow_\epsilon F_e & \text{with probability } \frac{\epsilon'}{\epsilon + \epsilon'} \\ (x \downarrow_{\epsilon'} F_o) \uparrow_{\epsilon'} F_e, & \text{with probability } \frac{\epsilon}{\epsilon + \epsilon'} \end{cases}$$

Here, ϵ and ϵ' are chosen to be the largest possible numbers such that both of the assignments $(x \uparrow_\epsilon F_o) \downarrow_\epsilon F_e$ and $(x \downarrow_{\epsilon'} F_o) \uparrow_{\epsilon'} F_e$ remain feasible with respect to all hard constraints. This finishes the definition of Operation $\mathcal{X}$.

Operation $\mathcal{X}$ satisfies properties (1) and (3) by construction—it reduces the number of floating constraints and it never violates any hard constraint. In addition, it satisfies the martingale property (*i.e.* property (2)). This holds because for any edge $x_{(i,j)}$ that changes in one iteration of Operation $\mathcal{X}$, one of the following can happen:

1. If $(i,j) \in F_o$, then Operation $\mathcal{X}$ increases $x_{(i,j)}$ by ϵ with probability $\frac{\epsilon'}{\epsilon + \epsilon'}$ and decreases it by ϵ' with probability $\frac{\epsilon}{\epsilon + \epsilon'}$. In this case, the expected amount by which $x_{(i,j)}$ changes is equal to $\epsilon \cdot \frac{\epsilon'}{\epsilon + \epsilon'} - \epsilon' \cdot \frac{\epsilon}{\epsilon + \epsilon'} = 0$.
2. If $(i,j) \in F_e$, then Operation $\mathcal{X}$ decreases $x_{(i,j)}$ by ϵ with probability $\frac{\epsilon'}{\epsilon + \epsilon'}$, and increases it by ϵ' with probability $\frac{\epsilon}{\epsilon + \epsilon'}$. In this case, the expected amount by which $x_{(i,j)}$ changes is equal to $-\epsilon \cdot \frac{\epsilon'}{\epsilon + \epsilon'} + \epsilon' \cdot \frac{\epsilon}{\epsilon + \epsilon'} = 0$.

Therefore, $\mathbb{E}(x_{t+1} | x_t) = x_t$. Hence, by the end of the iterative process, the expected value of the final pure allocation is equal to x.

The most challenging step is to prove that at the end of the process, the soft constraints that are in the deepest level of $\mathcal{H}$ are approximately satisfied. We only discuss the intuition for this step here. Operation $\mathcal{X}$ is designed in such a way that it never increases (or decreases) two or more elements of a soft block at the same iteration. Consequently, elements of each soft block become *negatively correlated*. The negative correlation property then allows us to employ probabilistic concentration bounds (Chernoff bounds, as explained in Supplementary Appendix D) to prove that the soft constraints are approximately satisfied.

Remarkably, Operation $\mathcal{X}$ never exploits the structure of the soft blocks and only takes as input the structure of the hard blocks, $\mathcal{H}$. The property that it never increases (or decreases) two or more elements of a soft block in one iteration holds *regardless of the structure of the soft blocks.* Consequently, the main theorem holds even if the set of soft constraints includes *all* constraints that are in the deepest level of the bihierarchy.

3.3. *Tightness*

As we just discussed, Operation $\mathcal{X}$ exploits the negative correlation property of the elements of a soft block. We derive our results by applying Chernoff concentration bounds for independent

random variables, which are also applicable for negatively correlated variables. One may ask: is it possible to exploit the negative correlation property and improve the error bounds of Theorem 1 for approximate satisfaction of the soft constraints? Next, we show that those bounds are *tight*, up to multiplicative constants in the exponents.

Proposition 1 *Consider a lottery that, given any hard–soft partitioned constraint structure, guarantees to satisfy the hard constraints and gives the following guarantees for the satisfaction of soft constraints: there exists a constant* $\overline{\epsilon} \in (0,1)$ *such that for any* $\epsilon \in (0,\overline{\epsilon})$*, and for any soft constraint defined on a block* S *with* $\sum_{e\in S} x_e = \mu$*, the lottery guarantees that*

$$\Pr\left[\sum_{e\in S} X_e \leq \mu(1-\epsilon)\right] \leq f(\mu,\epsilon),$$

$$\Pr\left[\sum_{e\in S} X_e \geq \mu(1+\epsilon)\right] \leq f(\mu,\epsilon).$$

Then, there exists a constant $c>0$ *such that, for any* $\epsilon \in (0,\overline{\epsilon})$*,* $\lim_{\mu\to\infty} \frac{e^{-\frac{\epsilon^2\mu}{c}}}{f(\mu,\epsilon)} = 0$.

Proposition 1 shows that there exists a constant $c>0$ such that any lottery that satisfies the hard constraints can approximately satisfy soft constraints (in the sense of Definition 1) with a probabilistic guarantee no better than $e^{-\frac{\epsilon^2\mu}{c}}$. We prove this result in Appendix B.1. The proof works by constructing a sequence of instances (indexed by the number of agents) such that no lottery can perform better than the exponential bounds provided by the proposition in that sequence. While the proof reveals that any constant $c \leq 2/3$ suffices for the result to hold, it does not optimize to attain the largest possible such c.

3.4. *The trade-off between hard and soft structures*

Theorem 1 requires the soft blocks to be in the deepest level of the hard structure. Under what conditions it would be possible to implement an *arbitrarily complex* set of soft constraints? We will show that this would be possible if either the structure of hard blocks is "sufficiently simple," or with weaker probabilistic bounds. These results expose a trade-off between the power of the probabilistic bounds that we provide and the complexity of the structure of soft constraints with respect to the structure of hard constraints.

3.4.1. Arbitrary soft structure with simpler hard structure. First and foremost, it follows from Theorem 1 that if the hard structure is a single hierarchy, then the soft constraint structure can be arbitrarily complex, without any loss in the power of the bounds. This is formalized in the following proposition.

Proposition 2 *Let* $\mathcal{E} = \mathcal{H} \cup \mathcal{S}$ *be a hard–soft partitioned constraint set, where* $\mathcal{H}$ *is a single hierarchy; i.e.,* $\mathcal{H}_1 = \emptyset$ *or* $\mathcal{H}_2 = \emptyset$*. Then, for all* $\mathcal{S} \subseteq 2^E$*,* $\mathcal{E}$ *is approximately implementable.*

We discuss the applications of this result in Section 4.

3.4.2. Arbitrary soft structure with weaker approximation bounds. It follows from Theorem 1 that if the hard structure has its maximal form (*i.e.* bihierarchy), our implementation

mechanism can still approximately satisfy any soft constraint, but with weaker approximation guarantees. To formalize this idea, we need a new definition. We say that a block $B \in \mathcal{S}$ is in *depth k of hierarchy* $\mathcal{H}_1$ if B can be partitioned into k subsets $B_1, B_2, \ldots, B_k$ such that all are in the deepest level of $\mathcal{H}_1$ and, moreover, no partitioning of B into $k-1$ subsets satisfies this property. We also say that $B \in \mathcal{S}$ is in *depth k of bihierarchy* $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$ if it is in depth k of either of $\mathcal{H}_1$ or $\mathcal{H}_2$.

Proposition 3 *Let $\mathcal{E} = \mathcal{H} \cup \mathcal{S}$ be a hard–soft partitioned structure such that $\mathcal{H}$ is a bihierarchy. Then, $\mathcal{E}$ is approximately implementable in the sense of Definition 1, with one difference: for any soft block $B \in \mathcal{S}$ that is in the depth k of $\mathcal{H}$, equations (2.2) and (2.3) will change to:*

$$\Pr(\mathsf{dev}^+ \geq \epsilon\mu) \leq k \cdot e^{-\mu \frac{\epsilon^2}{3k}} \tag{3.4}$$

$$\Pr(\mathsf{dev}^- \geq \epsilon\mu) \leq k \cdot e^{-\mu \frac{\epsilon^2}{2k}}. \tag{3.5}$$

Note that when $k = 1$, the above bounds coincide with the bounds of Theorem 1. Therefore, this proposition generalizes Theorem 1. We prove this result in Appendix B.2. The essential component of the proof is applying a union bound on (2.2) and (2.3).

Thus, implementing an arbitrary soft constraint structure is feasible with a compromise over either the generality of the hard structure, or the strength of the probabilistic bounds.

3.5. *Additive bounds when agents have types*

In this section, we show that it is possible to design Operation $\mathcal{X}$-based lotteries with additive error guarantees when there is only a small number of *types* of agents in the economic environment. For the sake of exposition, we use school choice as our motivating example.

Let N and O represent the set of students and schools, respectively. There is a partitioned structure $\mathcal{E} = \mathcal{H} \cup \mathcal{S}$, where $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$ is a bihierarchy. Suppose that the hierarchy $\mathcal{H}_1$ is the set of all row blocks and that the hierarchy $\mathcal{H}_2$ contains the set of all column blocks (note that we allow $\mathcal{H}_2$ to contain other blocks as well). The row blocks ensure that every student will be assigned to a school, and the column blocks ensure that the schools' capacity constraints will be satisfied. Let $\mathcal{S}$ be the set of blocks that are in the deepest level of $\mathcal{H}_2$. Throughout this section, we assume that the variables in soft constraints have coefficients that are either 0 or 1 (similar to hard constraints).

We say a student $n \in N$ *participates* in a constraint if there exists some object $o \in O$ such that the coefficient of x_{no} is positive in that constraint. We say two students have the same *type* if whenever one of them participates in a constraint in $\mathcal{H}_2 \cup \mathcal{S}$, the other one also does. We denote the set of all types by $\mathcal{T}$.

For example, consider a school choice problem where each school has a hard capacity constraint, as well as a soft constraint on the number of students from low socioeconomic status. In this case, $|\mathcal{T}| = 2$: the two types correspond to the students with low socioeconomic status and the rest of the students. Our main result in this section states that any feasible fractional assignment is approximately implementable with additive error at most $|\mathcal{T}|$. That is, with probability one, soft constraints will not be violated by more than $|\mathcal{T}|$.

To state the main theorem of this section, we first modify the definition of approximation implementation to the case of deterministic additive bounds.

Definition 2 *We say that a partitioned structure $\mathcal{E} = \mathcal{H} \cup \mathcal{S}$ is approximately implementable with additive error* k*, if all conditions of Definition 1 are satisfied with the difference that for Property 3*

(the approximate satisfaction of soft constraints), the new requirement is that for any soft block $B \in \mathcal{S}$ *with* $\sum_{e \in B} x_e = \mu$, *we have* $|\sum_{e \in B} X_e - \mu| \leq k$.

Theorem 2 *When there are T student types, any feasible fractional assignment x is approximately implementable with additive error T.*

The proof is in Appendix B.3. Applying the implementation method of Theorem 1 directly does not provide the deterministic bounds of Theorem 2. We use a different method: we first expand the set of hard constraints by adding constraints that bound the number of students assigned from each type to each school from above and below, and then use Operation $\mathcal{X}$ for implementation.

This theorem shows that in designing random allocation mechanisms for specific economic applications, one may be able to provide stronger and even deterministic bounds. In real-world settings, are the soft constraints closer to the ones in Theorem 1 or Theorem 2? The answer depends on the specific application in hand. Let us elaborate with an example.

Consider a school choice problem with two different walk zones, where each school has minimum quotas for both low SES students and students with disabilities. Hence, $|\mathcal{T}| = 8$. This is likely an acceptable error bound in a school choice setting. However, if the number of walk zones goes up to 10, we would have $|\mathcal{T}| = 40$, which may or may not be an acceptable error bound. The number of types can grow even further in some other applications. For instance, in Section 4.1 we discuss a new method, recently adopted by Boston Public Schools, in which the walk zone of a school is a certain "radius" around its location. So, the number of walk zones would be as large as the number of schools, which makes the error bounds of Theorem 2 undesirable in this market. The error bounds of Theorem 1, on the other hand, are agnostic to the number of student types and do not depreciate when more student types are added by introducing additional soft constraints. The computational experiments in Supplementary Appendix A.5 demonstrate this for the *empirical* error bounds; that is, they do not depreciate when more student types are added.

From a practitioner's perspective, the choice between the methods provided by Theorem 1 and Theorem 2 depends on the level of complexity of the soft constraints and the level of tolerance for violating them. While Theorem 1 offers probabilistic guarantees for possibly complex structures, Theorem 2 provides deterministic bounds which are appealing for simpler soft structures where the number of types is small. We discuss the applications of these theorems in Section 4.

3.6. *Computational experiments on probabilistic bounds*

To assess the performance of our probabilistic guarantees in potential applications, we provide computational experiments in school choice settings with several constraints such as diversity and walk-zone constraints. We discuss the results in detail in Supplementary Appendix A. In sum, the experiments show that our bounds perform (much) better than the theoretical worst-case bounds of Theorem 1. In the most basic example, for a goal to admit 250 students from a specific walk zone, our *theoretical* bounds guarantee that the probability of a 10% violation is no more than $e^{-250 \times 0.1^2/3} \simeq 0.434$. Nevertheless, simulations show that the *empirical* probability of a 10% error is less than 0.064. When the number of students goes up to 500, meanwhile, the theoretical and empirical violation probabilities change to 0.188 and 0.024, respectively.

We extend our experiments in several ways by (1) using NYC public high school data (Nycdoe, 2019) for the number of schools and their capacity constraints, (2) including walk-zone and several (intersecting) diversity constraints in the assignment problem, and (3) including correlation in students' preferences. In our experiments with the NYC high school data, for instance, in at most 2% of the schools there is a 10% or higher violation of the capacity constraint, and in at most

6% of the schools there is a 10% or higher violation of the walk-zone constraint.[12] The fraction goes up to 6% because walk-zone constraints have a smaller right-hand side, which is half of the right-hand side of the capacity constraint. That said, even for this smaller right-hand side, violations become rare with slightly larger error tolerance; *e.g.*, the probability of violation by more than 15% is around 0.01. To compare this with typical violations that may happen in real world, we note that, for instance in our data from Manhattan, nearly 17% of high schools have violations of more than 10% in their capacity constraints.

Why do our probabilistic bounds perform better empirically? The proof of the main theorem provides some intuition. We first prove that the random variables in each soft block are *negatively correlated*. Then, since negative correlation is stronger than *independence*, we apply standard concentration bounds for independent random variables to prove our bounds. Therefore, we expect our algorithm to perform better in practice due to negative correlation. In Appendix B.5, we show why negative correlation can lead to improved bounds using an example.

4. INTERSECTING CONSTRAINTS IN PRACTICE

Intersecting constraints arise in a variety of settings. We consider two real-world settings that admit intersecting constraints: school choice (discussed here) and refugee resettlement (discussed in Supplementary Appendix B). We show how our framework can incorporate soft constraints in these settings.

Consider a school choice setting, where a set $N=\{1,\cdots,n\}$ of students are to be assigned to a set $O=\{1,\cdots,k\}$ of schools. Several types of constraints arise in this market. A few examples are capacity constraints of schools, reserved capacities for students in walk zones, affirmative action policies,[13] and grade-based quotas.[14] The bihierarchy assumption often fails in this setting since such constraints typically intersect. (See Example 1.) However, several of these constraints can be considered as flexible constraints.[15]

We model the school choice problem in our setting as follows. Let $\mathcal{H}$ be a single hierarchy which includes the student-side inflexible constraints. Each student should be assigned to exactly one school. Hence, one can define $\mathcal{H}$ to be the set of all student-side capacity blocks, where $\underline{q}_B=\bar{q}_B=1$ for all $B\in\mathcal{H}$. Suppose all other constraints are soft. Then, by Proposition 2, any general set of constraints can be approximately satisfied.

We can go further by considering a setting where $\mathcal{H}$ also includes school-side capacity blocks; that is, school capacities cannot be violated. In this case, structures with arbitrary soft blocks are not approximately implementable, but it follows from Theorem 1 that a reasonably general

12. These are substantially lower than what Theorem 1 guarantees. The median NYC school has capacity above 500. As we discussed before, the bound proved in Theorem 1 for a 10% violation of a constraint with size 500 is 0.19. For walk-zone constraints, since the size is lower, the theoretical bounds are larger.

13. Affirmative action is defined as "positive steps taken to increase the representation of women and minorities in areas of employment, education, and culture from which they have been historically excluded" Stanford Encyclopedia of Philosophy (2013). One goal of such policies is to increase diversity and to balance out the social effects that weaken specific groups. Another argument in favour of affirmative action policies is that they increase structural integration, which "serves the ideal of equal opportunity" (Jacobs, 2004). Affirmative action policies are usually implemented as minimum quotas on students within a minority group. See Abdulkadiroğlu and Sönmez (2003), Hafalir *et al.* (2013), and Kominers and Sönmez (2016) for theoretical analysis of affirmative action policies.

14. Schools may have grade-based diversity policies. For instance, New York City's Educational Option program has quotas based on test scores; see Abdulkadiroğlu *et al.* (2005).

15. In fact, New York City public school system data show that the capacity constraints of schools are on average violated by around 20%. We discuss these data in Supplementary Appendix A.

structure is implementable. In particular, we say that a block is *local* if it involves one student with possibly multiple schools or one school with possibly multiple students, but not multiple schools and multiple students at the same time. In other words, a block is local if it includes a subset of the elements of a single column or a single row.[16]

Proposition 4 *Let $\mathcal{E}=\mathcal{H}\cup\mathcal{S}$ be a structure such that $\mathcal{H}$ is the set of all agents' and objects' capacity blocks and $\mathcal{S}$ only contains local blocks. Then $\mathcal{E}$ is approximately implementable.*

Last but not least, the soft constraints in school choice are sometimes such that Theorem 2 can provide reasonable additive bounds for practical applications. For instance, in Boston Public Schools (BPS) program and until a few years ago, for students in kindergarten through Grade 8, two main considerations are walk zones (East, West, and North zones, as reported in Abdulkadiroğlu *et al.* (2005)) and the SES status ("free lunch" and "paid lunch" students). This forms six different types of students. Therefore, Theorem 2 guarantees that all constraints can be satisfied by an additive error of at most 6.

4.1. *An alternative approach to walk-zone priorities*

We can employ our framework to develop an alternative approach for handling walk-zone priorities in school choice. A common way to implement walk-zone priorities is to partition the city into artificial zones and impose quotas on students living in the same zone as the school. By construction, this method treats students who live just inside and outside of a zone's border very differently. Recently, some public school systems have adopted a new method, in which the walk zone of a student is a certain "radius" around where the student lives. For instance, BPS recently revised its assignment policy; in particular, it now states:

> *BPS will offer a customized list of school choices for every family based on their home address. It includes every school within a one-mile radius of their home...*[17]

Even this method has some discontinuous behaviour: effectively, it draws a one-mile radius circle around each *school*, and considers the students inside that circle as the walk-zone students of that school. Thus, this method is essentially same as the traditional walk-zone method, with the difference that each school has its own walk zone. Again, two students who live just inside and outside of a school's zone are treated differently.

Building on our framework, we propose a new method to handle walk-zone priorities. Let d_{sc} be the "priority function" of assigning a student s to a school c. The walk-zone constraint can be stated as $\sum_{s\in N} d_{sc}x_{sc}\geq q_c$, where q_c can be used to adjust the significance of walk-zone priority. In the standard walk-zone priority formulation, $d_{sc}=1$ if s and c are in the same walk zone, and $d_{sc}=0$ otherwise. However, we can define d_{sc} to be, *e.g.*, $1/z_{sc}$, where z_{sc} is the *distance* of student s from school c (or the commute time). Our setting allows for any arbitrary priority function. This way of accommodating walk-zone priorities can ensure that there is no "discontinuity" on the borders of different zones.

16. This model of "local" structures, which is a special case of our model, has been studied in Khuller *et al.* (2006) as well.

17. https://www.bostonpublicschools.org/assignment, accessed 10/07/2018.

5. APPLICATION I: UTILITY GUARANTEES

We now turn into the question of *ex post* properties of our implementation mechanism. As discussed before, a key motivation for randomization is to restore fairness. Nevertheless, even if the constructed fractional assignment is fair, there could be very large discrepancies in *realized* utilities, as discussed in Kojima (2009). The following example clarifies this point.

Example 2 *Suppose there are two agents and we wish to allocate* $2k$ *objects between them. Each agent is supposed to receive* k *objects. Both agents receive a utility* v_i *from object* i*, where* $v_1 > v_2 > \cdots > v_{2k}$*, and their utilities are additive. In a fair fractional allocation, each agent receives half of each object. We can implement this allocation in two different ways:* (1) *randomly choose one agent and let that agent choose her favourite* k *objects, or* (2) *choose* k *objects randomly, assign them to agent* 1*, and assign the remaining objects to agent* 2*. It is clear that the second way is more fair ex post since in the first way one agent always receives all of the most popular objects.*

Here, our goal is to show that when a fractional assignment x is implemented via Theorem 1, an agent's *ex post* utility is approximately equal to her *ex ante* utility, in a sense to be formalized soon. In the case of Example 2, Operation $\mathcal{X}$ produces an (*ex post*) allocation closer to the second implementation method. To show how our "utility guarantee" can be applied to different settings, we provide examples from two classic allocation mechanisms: the random serial dictatorship (RSD) mechanism and the pseudo-market mechanism. Our results extend these methods by handling intersecting constraints and, in addition, by providing approximate guarantees for the agents' *ex post* utilities in settings with such constraints.

5.1. *Setup*

We introduce some notation before presenting our utility guarantees.

Definition 3 *For two non-negative random variables* x, y*, we write* $x \lesssim y$ *if there exists a constant* $\mu > 0$ *such that for any* $\epsilon > 0$*,*

$$\Pr\Big(x \geq \mu(1+\epsilon)\Big) \leq e^{-\mu\epsilon^2/3},$$
$$\Pr\Big(y \leq \mu(1-\epsilon)\Big) \leq e^{-\mu\epsilon^2/2}.$$

We also say that x *is approximately upper bounded by* y *or, equivalently,* y *is approximately lower bounded by* x *when* $x \lesssim y$ *holds. When* $x = y$ *and* $\mathbb{E}[x] = \mu$*, then if the above inequalities hold, we say that* x *is approximately equal to* μ*, and denote it by* $x \approx \mu$*.*

For example, if a random variable x is approximately lower bounded by a constant μ, then the probability of x being less than $\mu(1-\epsilon)$ decreases exponentially in μ, for any $\epsilon > 0$. Notably, the two probabilistic bounds in the above definitions are essentially the same bounds as (2.2) and (2.3). These are the typical multiplicative forms of Chernoff concentration bounds.

Consider agents with von Neumann–Morgenstern utility functions that are additive across objects. That is, the utility of an agent i from any (fractional or pure) allocation x is defined by $u_i(x) = \sum_{k=1}^{|O|} x_{ik} u_{ik}$. Without loss of generality, it is supposed that $u_{ik} \in [0,1]$ for all i, k. Consider a hard–soft partitioned structure $\mathcal{E} = \mathcal{H} \cup \mathcal{S}$, with the restriction that all of the row blocks are in the deepest level of $\mathcal{H}$. We do not impose any restrictions on columns.

The following result shows that when the assignment x is implemented using Operation $\mathcal{X}$, the *ex post* utility of an agent i is approximately equal to her *ex ante* utility, $u_i(x)$.

Proposition 5 (Utility guarantee) *Let x be a feasible fractional assignment with respect to $\mathcal{E}$ and let the assignment X be the outcome of the mechanism that implements x via Theorem 1 (i.e. by the iterative application of Operation $\mathcal{X}$). Then, $u_i(X) \approx u_i(x)$.*

The restriction that all of the row blocks are in the deepest level of $\mathcal{H}$ is required since an agent's utility is a function of all of the elements of the row corresponding to her. As Theorem 1 requires the soft blocks to be in the deepest level of $\mathcal{H}$, the row blocks corresponding to the auxiliary constraints should be in the deepest level of $\mathcal{H}$ as well. It is also possible to provide utility guarantees without assuming that all of the row blocks are in the deepest level of $\mathcal{H}$; however, the guarantees will be weaker, similar to those of Proposition 3.

We remark that similar approximate guarantees can be provided for the *ex post* social welfare. Formally, define the *social welfare* and the *average welfare* under assignment x respectively by $W(x) = \sum_{i=1}^{|N|} u_i(x)$ and $\overline{W}(x) = \frac{W(x)}{|N|}$. A straightforward application of Proposition 3 then implies that

$$\Pr\Big(W(X) \leq (1-\epsilon)W(x)\Big) \leq |N| \cdot e^{-\overline{W}(X)\epsilon^2/2}.$$

Similar to our utility bounds, the above bound for social welfare is interesting when agents' utilities are relatively large, which is the case when several objects (in expectation) are allocated to agents. In a school choice setting where students have unit demand, for instance, these bounds cannot guarantee fairness. More generally, since each student is assigned to a single school, it is typically impossible to guarantee *ex post* fairness—after all, some student has to go to a less popular school. However, even in the school choice setting, our bounds provide *ex post* guarantees for schools' utilities, since a large number of students are being assigned to each school.

We emphasize that Budish (2011) and BCKM also provide *ex post* guarantees, but their guarantees have different mathematical and economic interpretations. In particular, Budish (2011) focuses on finding approximate competitive equilibrium from equal incomes. He defines a "maximin share" in the following way: an agent is allowed to divide objects into N bundles, and then receive the bundle with minimum utility. He then proves that in his mechanism, each agent's utility is at least equal to his maximin share, approximately. BCKM, who focus on implementing arbitrary fractional assignments, can provide utility bounds that guarantee the *ex post* utility of an agent is different from its *ex ante* utility by at most the utility difference between the most valuable and the least valuable objects, and this guarantee is deterministic.

We provide our utility bounds for a generalized constraint structure which allows for intersecting soft constraints. The generality of this structure makes the results in Budish (2011) and BCKM inapplicable and, thus, we provide bounds by exploiting the negative correlation property of Operation $\mathcal{X}$. We now provide two examples of classic assignment mechanisms in which our implementation method based on Operation $\mathcal{X}$, and thus our utility guarantees, may be applied.

5.2. *Example 1: approximate random serial dictatorship*

The contribution of this section is modifying RSD in a multi-unit demand setting with intersecting constraints to guarantee its *ex post* "approximate fairness."

The RSD mechanism is one of the most popular mechanisms for the allocation of indivisible objects. In a simple single-unit demand setting, the RSD mechanism first draws an ordering of agents uniformly at random and then lets the agents select their favourite object (among the

remaining objects) one by one according to the realized random ordering. In a multi-unit demand setting RSD is defined similarly, except that each agent can select her favourite bundle of objects at her turn.

The RSD mechanism is strategy-proof, *ex post* Pareto efficient,[18] and *ex ante* fair[19] (Abdulkadiroğlu and Sönmez, 1998; Chen and Sonmez, 2002). On the downside, it can be *ex ante* inefficient, *ex post* unfair, and it cannot accomodate lower quotas (Bogomolnaia and Moulin, 2001; Hatfield, 2009; Kojima, 2009). While Che and Kojima (2010) show that under some conditions the *ex ante* inefficiency vanishes in large markets, the *ex post* unfairness (as illustrated in Example 2) remains a concern. We address this concern by employing the utility bound developed in Proposition 5.

We adopt the same model as in Section 5.1. Recall that there we considered a hard–soft partitioned structure, $\mathcal{E}=\mathcal{H}\cup\mathcal{S}$. Here, we suppose that all of the lower quotas are set to zero. When this condition holds, RSD extends to our setting in a natural way in the following way: The mechanism orders agents randomly. Then, one by one in that order, each agent is allowed to choose any subset of the objects that does not cause a violation of any of the (upper quota) constraints in $\mathcal{E}$. We denote the resulting pure assignment by $X_\pi^{\mathcal{E}}$, where π denotes the ordering of agents that is chosen by the mechanism.

We are now ready to introduce the new mechanism, the *Approximate Random Serial Dictatorship* (ARSD) mechanism, and prove that this mechanism preserves the *ex ante* fairness properties of RSD while being *ex post* approximately fair.

The idea is simple: RSD induces an *ex ante* assignment. This assignment can be constructed as follows. Let Π denote the set of all orderings over agents. Define $x_{rsd}=\frac{1}{|\Pi|}\sum_{\pi\in\Pi}X_\pi^{\mathcal{E}}$. Then, implement x_{rsd} via Theorem 1, so that the constraints in $\mathcal{H}$ are satisfied and the constraints in $\mathcal{S}$ are approximately satisfied. We now summarize these steps.

The approximate random serial dictatorship mechanism (ARSD)

1. Agents report their cardinal values for objects (*i.e.* each agent a reports $u_{a1},\ldots,u_{a|O|}$).
2. The mechanism computes x_{rsd}.
3. The mechanism implements x_{rsd} via Theorem 1.

Note that the RSD and ARSD mechanisms implement the *exact* same *ex ante* assignment, but their *ex post* properties are different. It is well-known that this *ex ante* assignment satisfies desirable fairness properties (Abdulkadiroğlu and Sönmez, 1998).[20] We call x_{rsd} the *ex ante ARSD assignment*, and the outcome of the ARSD mechanism the *ex post ARSD assignment*. Finally, we say that a mechanism is *strategy-proof* (or dominant-strategy-incentive-compatible) if it is a weakly dominant strategy for every player to report her (private) cardinal values for objects truthfully to the mechanism.

Proposition 6 *The ARSD mechanism is strategy-proof. Moreover, the utility of any agent in the ex post ARSD assignment is approximately equal to her utility in the ex ante ARSD assignment.*

18. Recall that a pure assignment of objects to agents is said to be *ex post* Pareto efficient if there exists no other pure assignment in which any agent is weakly better off and at least one agent is strictly better.

19. RSD is *ex ante* fair, *e.g.*, in the sense that it respects *equal treatment of equals*. An allocation mechanism is said to respect equal treatment of equals if agents with the same utilities over bundles of objects have the same allocations. RSD satisfies the "equal treatment of equals" and the "SD envy-freeness" criteria.

20. For example, it respects equal treatment of equals, in the sense that the allocations of agents who are identical up to relabelling are the same.

Some intuition for this result comes from the way Operation $\mathcal{X}$ randomly allocates the objects. The negative correlation property of Operation $\mathcal{X}$ guarantees that when an agent receives a popular object, she is (weakly) less likely to receive yet another popular one. Our method alleviates the discussed *ex post* unfairness of RSD in settings where each agent receives a large number of objects. For settings where agents receive a small number of objects, our bounds are not practically relevant. In that case, however, the implementation strategy of Theorem 2 can lead to better bounds.

5.3. *Example 2: the approximate pseudo-market mechanism*

Hylland and Zeckhauser (1979) propose a remarkable design for assigning n objects to n agents in an *ex ante* efficient way. They allocate all agents with an equal amount of an artificial currency, ask them to report their von Neumann–Morgenstern preferences, and then solve for the CEEI of this "pseudo-market." The resulting fractional assignment is *ex ante* efficient and envy-free by the properties of the competitive equilibrium allocation. BCKM generalized that framework to a multi-unit demand setting, where objects may have capacity constraints. We propose a generalization of HZ and BCKM's mechanisms. Our contribution is to allow for a rich family of soft constraints, including intersecting constraints. In addition to that, the outcome of our mechanism is approximately *ex post envy-free*, a property that can be guaranteed only *ex ante* in HZ and BCKM.

We adopt the basic setup defined in Section 5.1. Recall that there we considered a hard–soft partitioned structure, $\mathcal{E}=\mathcal{H}\cup\mathcal{S}$. Here, we assume that $\mathcal{H}=\mathcal{H}_1\cup\mathcal{H}_2$, where $\mathcal{H}_1$ is the set of all row blocks and $\mathcal{H}_2$ is the set of all column blocks, respectively.[21] In addition, we allow the set of soft constraints to contain any sub-row, *i.e.*, any block in the deepest level of $\mathcal{H}_1$. All of the lower quotas are set to 0. The structure of $\mathcal{E}$ ensures the existence of a feasible fractional solution, *i.e.,* a CEEI if objects were divisible.

We define a few notions before presenting the mechanism. A vector $x_i=(x_{i1},\ldots,x_{i|O|})$ is a *feasible bundle for agent i* if x_i satisfies all (hard and soft) row and sub-row constraints from $\mathcal{E}$ in which agent i participates. Let $\mathcal{F}_i$ be the set of all feasible bundles for agent i. Given a vector of prices for objects, $p=[p_k]_{k\in O}$, we say x_i is a *budget feasible bundle for agent i with respect to p* if $\sum_{k\in O} p_k x_{ik}\leq B$. Let $\mathcal{B}_i(p)$ be the set of all budget feasible bundles for agent i with respect to p. Finally, denote the capacity of an object k by q_k. Recall that $u_i(x_i)$ denotes the utility of an agent i from a feasible bundle x_i.

The approximate pseudo-market mechanism

1. Agents report their cardinal object values (*i.e.* each agent a reports $u_{a1},\ldots,u_{a|O|}$).
2. Assign to each agent an artificial budget B. Compute a vector of nonnegative prices $p=[p_k]_{k\in O}$ and a fractional assignment $x=[x_i]_{i\in N}$ such that:
   (a) $x_i=\text{argmax}_{x\in\mathcal{F}_i\cap\mathcal{B}_i(p)}\{u_i(x)\}$, for all $i\in N$,
   (b) $\sum_{i\in N} x_{ik}\leq q_k$, for all $k\in O$, and $\sum_{i\in N} x_{ik}<q_k$ only if $p_k=0$.
3. Implement x via Theorem 1.

21. We can relax the structure of the hard constraints by allowing $\mathcal{H}_1$ to be a hierarchy that contains additional sub-row constraints, in exchange for weaker guarantees for the agents' *ex post* utilities (in the sense of Proposition 3).

In Step 21, we construct the fractional allocation by solving for the competitive equilibrium of the market, giving all agents an artificial budget of B. The existence of the price vector and the fractional assignment of Step 21 follows directly from Theorem 6 of BCKM. We call this assignment the *ex ante assignment*. In Step 21, the mechanism generates the *ex post assignment* by implementing the *ex ante* assignment.

Since each agent is solving an individual utility maximization problem (stated in 2-a), the assignment x is envy-free. Recall that an assignment x is *envy-free* if $u_i(x_j) \leq u_i(x_i)$ for all $i, j \in N$. We now show that the implementation step (Step 21) maintains some of the nice features of the *ex ante* assignment, including envy-freeness, approximately. We say that a random assignment X is *approximately envy-free* if $u_i(X_j) \lesssim u_i(X_i)$ for all $i, j \in N$.

Proposition 7 *The assignment generated by the approximate pseudo-market mechanism is approximately envy-free. Furthermore, the utility of each agent in the assignment is approximately lower bounded by her utility in the ex ante assignment. The ex post assignment is equal to the ex ante assignment in expectation, satisfies the hard constraints, and approximately satisfies the soft constraints.*

Finally, we remark that the structure of hard constraints in the above proposition can be relaxed. In particular, we can allow the hierarchy $\mathcal{H}_1$ to contain additional sub-row constraints, in exchange for weaker guarantees for the agents' *ex post* utilities (in the sense of Proposition 3). The proof remains the same, *mutatis mutandis*.

6. APPLICATION II: COMPETITIVE EQUILIBRIUM

In this section, we apply our implementation method to prove the existence of an ϵ-competitive equilibrium (ϵ-CE) in large markets in allocation problems with indivisible objects, where agents have additive utilities and possibly intersecting constraints. It is known that the standard existence results of competitive equilibrium (CE) fail in settings with indivisibilities (Henry, 1970). Following this result, a body of literature studies conditions under which the existence of CE in the presence of indivisibilities is guaranteed.

Dierker (1971) shows that an equilibrium exists, provided that the number of agents is large relative to the number of commodities, or if agents are insensitive to "small" price changes, and therefore may slightly violate their budget constraint. Broome (1972) shows that if at least one commodity is divisible, then there exists an "approximate" equilibrium, where the approximation is in two dimensions: The allocation is only approximately feasible, and agents are only nearly optimizing. Mas-Colell (1977) establishes the existence of competitive equilibrium when there exist at least one divisible commodity and a continuum of agents. Budish (2011) shows that when (1) the capacity constraints are relaxed and (2) agents are provided slightly different budgets at random, an approximate competitive equilibrium exists in a combinatorial economy with indivisible objects. The error rate in satisfying the capacity constraints grows with the total number of commodities and the maximum number of commodities that each agent is interested in. More recently, Babaioff *et al.* (2017) study a model close to ours but without distributional constraints. They consider an environment with two agents with equal budgets and show that competitive equilibrium exists when vanishingly small perturbations are added to the budgets.

Relative to the previous literature, this section has one conceptual and one technical contribution. On the conceptual side, we prove the existence of ϵ-CE in an environment where *each agent imposes a set of (possibly intersecting) constraints as part of her preferences*. Unlike the mentioned prior work, our specification does not impose a limit on the total number of commodities, or on the number of commodities that an agent is interested in. The specification

of hard constraints by the agents is of practical interest in settings such as online advertisement, where advertisers are typically allowed to target specific groups of users; for instance, an advertiser can specify, in part, that "I want at most 40,000 ads to be shown to users who live in Northern California, with at most 15,000 of them to those living outside of the Bay Area." An application of our result is using competitive equilibrium as a solution concept for pricing online ad impressions, which has recently been considered by Facebook (Hou *et al.*, 2016). Our solution readily extends to the case where the agents can also specify soft constraints, where the probabilistic bounds of Theorem 1 and Proposition 3 would be applicable.

On the technical aspect, the proof employs the implementation of fractional assignments via Operation $\mathcal{X}$, and then applies the "probabilistic method," as described in Alon and Spencer (2004), to establish the existence of ϵ-CE. We remark that the utility guarantees of Section 5 alone do not suffice to establish the existence, since the analysis here should also accommodate (hard) budget constraints. Nevertheless, we can use the probabilistic guarantees of Proposition 3 for soft constraints to accommodate the (hard) budget constraints.

For our first theorem, we will suppose that the set of constraints imposed by each agent is a hierarchy, and prove the existence of an ϵ-CE when the market is sufficiently large. We will dismiss the hierarchy assumption in our second theorem in exchange for a slightly stronger large market assumption. We present the theorems after defining the economy formally.

Consider an economy with a set of agents and a set of objects, respectively denoted by N, O. Any agent $a \in N$ is endowed with an initial budget of $w_a \in \mathbb{R}^+$. Objects are in unit supply.[22] Each agent imposes a set of hard constraints, $\mathcal{H}_a$, on the assignment. We suppose that all of the constraints in $\mathcal{H}_a$ involve no other agent than a (*i.e.* the constraints in $\mathcal{H}_a$ are local)[23] and that all of the corresponding lower quotas in $\mathcal{H}_a$ are equal to 0.[24]

A subset of objects $S \subseteq O$ is *feasible* with respect to $\mathcal{H}_a$ if all of the constraints in $\mathcal{H}_a$ are satisfied when the set of objects assigned to agent a is equal to S. Agents have *additive utilities* across feasible subsets of objects: there exist values $(u_{ao})_{o \in O}$ such that an agent a's utility from owning a subset of objects X_a which is feasible with respect to $\mathcal{H}_a$ is $\sum_{o \in X_a} u_{ao}$.[25] Without loss of generality, it is supposed that $u_{ao} \in [0,1]$ for all a, o. The *utility function of agent a* is a function $u_a : 2^O \to \mathbb{R}_+$ such that, for any $S \subseteq O$, $u_a(S)$ denotes the maximum utility that agent a can attain from owning a subset of S which is feasible with respect to $\mathcal{H}_a$.

For any $S \subseteq O$, we use $\mathbb{1}_S$ to denote the binary vector $(y_1, \ldots, y_{|O|})$, where $y_o = 1$ if $o \in S$ and $y_o = 0$ otherwise. Define the *set of feasible bundles* for agent a by

$$F_a = \left\{ \mathbb{1}_S : S \subseteq O, S \text{ is feasible with respect to } \mathcal{H}_a \right\}.$$

For a price vector $\mathbf{p} = (p_1, \ldots, p_{|O|})$, the budget set of an agent a is defined by

$$B_a(\mathbf{p}) = \left\{ \mathbb{1}_S : S \subseteq O, \sum_{o \in S} p_o \leq w_a \right\}.$$

The indirect utility function of agent a is defined by

$$v_a(\mathbf{p}) = \max_{y \in F_a \cap B_a(\mathbf{p})} \left\{ u_a(y) \right\}.$$

22. This can easily be extended to a multi-unit supply by considering each "copy" of an object as an object.
23. The notion of local constraint was defined in Section 4.
24. This is the setting for the pseudo-market mechanism of BCKM.
25. Recall that for any (pure or fractional) assignment x, we use x_i to denote the vector $(x_{i1}, \ldots, x_{i|O|}) \in \mathbb{R}^{|O|}$, *i.e.*, x_i denotes the allocation of agent i.

Definition 4 *For a price vector* $\mathbf{p}$ *and a pure assignment* X *of objects to agents,* $(\mathbf{p}, X)$ *is called an* ϵ*-Competitive Equilibrium (*ϵ*-CE) if:*

1. *For any object* o *we have* $\sum_{a\in N} X_{ao} \leq 1$*, with* $\sum_{a\in N} X_{ao} < 1$ *only if* $p_o = 0$.
2. *For all* $a \in N$, $X_a \in F_a \cap B_a(\mathbf{p})$.
3. *For all* $a \in N$, $u_a(X_a) \geq v_a(\mathbf{p}) \cdot (1-\epsilon)$.

In our first theorem, we suppose that $\mathcal{H}_a$ is a hierarchy for all agents $a \in N$, and show that for any arbitrary small $\epsilon > 0$, an ϵ-CE always exists when the market is *sufficiently large*, as defined below. We remark that this does not hold when $\epsilon = 0$: then, a CE does not always exist in sufficiently large markets, even when $\mathcal{H}_a = \varnothing$ for all $a \in N$, as shown in Supplementary Appendix C.1. Later, in our second theorem, we will dismiss the hierarchy assumption in exchange for a slightly stronger large market assumption.

Definition 5 (The large market assumption) *Consider a sequence of markets,* $\mathcal{M}_1, \ldots, \mathcal{M}_q, \ldots$*, where the set of agents, their budgets, and the number of the hard constraints imposed by each agent remain the same in all of the markets in the sequence.*[26] *Let* O_q *denote the set of objects,* $u_a^q : 2^{O_q} \to \mathbb{R}_+$ *denote the utility function of agent* a*, and* $\mathcal{H}_a^q$ *denote the set of hard constraints imposed by agent* a *in the market* $\mathcal{M}_q$*. We are in the* large market regime *if, as* $q \to \infty$*, we have* $u_a^q(O_q) \to \infty$ *for all agents* $a \in N$.

Proposition 8 *Suppose that* $\mathcal{H}_a$ *is a hierarchy for all agents* $a \in N$*. Then, for any fixed* $\epsilon > 0$*, there exists* q_0 *such that for all* $q > q_0$*, there exists an* ϵ*-CE in the market* $\mathcal{M}_q$.

Next, we dismiss the assumption of Proposition 8 that agents can impose only hierarchical constraints on the assignment. This generalization comes in exchange for a slightly stronger large market assumption which assumes that the right-hand sides of the agents' constraints grow with the market size.

Definition 6 (The large market assumption for intersecting constraints) *Under the* strengthened large market assumption *all of the assumptions of Definition 5 hold. In addition, the right-hand sides of all of the constraints imposed by agents approach infinity with* q.

Proposition 9 *Suppose that the strengthened large market assumption holds. Then, for any fixed* $\epsilon > 0$*, there exists* q_0 *such that for all* $q > q_0$*, there exists an* ϵ*-CE in the market* $\mathcal{M}_q$.

The proofs of Proposition 8 and Proposition 9 are technically involved and deferred to Online Appendix C.2.

Our large market assumptions in Definition 5 and Definition 6 require the number of hard constraints to be fixed as the market size grows. The practical plausibility of this assumption is necessarily context dependent. For instance, in an online advertisement setting, constraints are typically imposed on specific categories of agents (*e.g.* "male, under 40 years old"). For relatively large markets, the number of agents in any category is substantially more than the number of such categories.

26. We can allow these parameters to grow, but at a sufficiently slow rate.

7. CONCLUSION

We study the mechanism design problem of allocating indivisible objects to agents in a setting where cash transfers are precluded and the final allocation needs to satisfy some constraints. One efficient and *ex ante* fair solution to this problem is the "expected assignment" method, in which the mechanism first finds a feasible fractional assignment, and then implements that fractional assignment by running a lottery over feasible pure assignment. The previous literature have characterized a maximal "constraint structure" that can be accommodated into the expected assignment method. Such a structure rules out many real-world applications. We show that by reconceptualizing the role of constraints and treating some of them as *goals* rather than hard constraints, one can accommodate many more constraints.

The key theorem of the article identifies a rich constraint structure that is approximately implementable, meaning that any expected assignment that satisfies both hard constraints and soft constraints (*i.e.* goals) can be implemented by a lottery over pure assignments in a way such that hard constraints can be exactly satisfied and goals can be satisfied with only small errors.

Our framework allows designs that preserve some of the *ex ante* properties of the expected assignment in the *ex post* assignment. For instance, an envy-free or efficient expected assignment remains approximately envy-free and efficient *ex post*. We then apply this idea to modify the random serial dictatorship mechanism and the pseudo-market mechanism by expanding the structure of the constraints that they can accommodate. We also employ our framework to prove the existence of ϵ-equilibrium in an economy with indivisible objects, where agents can impose intersecting constraints as part of their preferences.

We are hopeful that the proposed framework for partitioning constraints into hard and soft, and the randomized mechanism we developed will pave the way for designing improved allocation mechanisms in practice.

Acknowledgments. Several conversations with Eric Budish, Paul Milgrom, Roger Myerson, and Alvin Roth have been essential. We are grateful to Ben Brooks, Gabriel Carroll, Yeon-Koo Che, Kareem Elnahal, Alex Frankel, Emir Kamenica, Fuhito Kojima, Matthew Jackson, Micheal Ostrovsky, Bobby Pakzad-Hurson, Bob Lucas, Ilya Segal, Bob Wilson, Alex Wolitzky, as well as several anonymous referees for their great suggestions.

Supplementary Data

Supplementary data are available at *Review of Economic Studies* online.

APPENDICES

A. PROOF OF THEOREM 1

In this section, we present the complete proof of Theorem 1. As discussed in the proof overview of the theorem, the proof is constructive. We will propose an implementation mechanism (or, equivalently, a lottery) that approximately implements a partitioned structure that satisfies the properties described in Theorem 1.

To describe the main idea of our mechanism, we need to introduce the notion of *tight* and *floating* constraints: a constraint is tight if it is binding. This notion is precisely defined in the following definition. First, for any block B, let $x(B)=\sum_{e\in B}x_e$.

Definition. *A constraint $S=(B,\underline{q}_B,\bar{q}_B)$ is* tight *if, either $x(B)=\underline{q}_B$ or $x(B)=\bar{q}_B$; otherwise, S is* floating. *Similarly, we say that a block B is tight when the constraint corresponding to it is tight.*

Note that this definition naturally applies to the (implicit) constraints that for all $e\in E$, we must have that $0\leq x_e\leq 1$.

In the core of our randomized mechanism is a stochastic operation that we call *Operation* $\mathcal{X}$. We iteratively apply Operation $\mathcal{X}$ to the initial fractional assignment. In each iteration t, the fractional assignment x_t is converted to x_{t+1} in a way such that: (1) the number of floating constraints decreases, (2) $\mathbb{E}(x_{t+1}|x_t)=x_t$, and (3) x_{t+1} is feasible with respect to

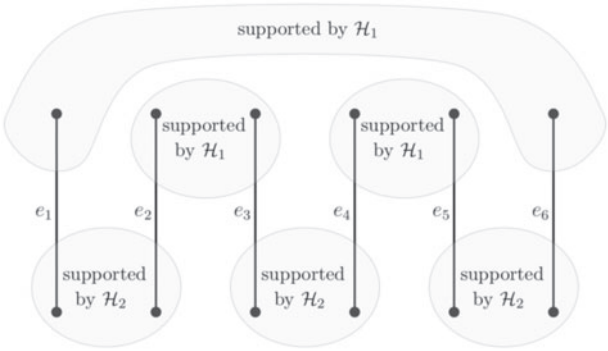

FIGURE A1
A floating cycle of length 6.

$\mathcal{H}$. The first property guarantees that after a finite (and small) number of iterations,[27] the obtained assignment is pure. The second property makes sure that the resulting pure assignment is equal to the original fractional assignment *in expectation*. The third property guarantees that all hard constraints are satisfied throughout the whole process of the mechanism. As the last step, we need to show that by iteratively applying of Operation $\mathcal{X}$, soft constraints are approximately satisfied. This is a more technical property of Operation $\mathcal{X}$, which we discuss in Appendix A.4. Roughly speaking, we design Operation $\mathcal{X}$ in such a way that it never increases (or decreases) two (or more) elements of a soft constraint in the same iteration. Consequently, elements of each soft block become "negatively correlated." We then can employ Chernoff concentration bounds to prove that soft constraints are approximately satisfied.

In the rest of this section, we design Operation $\mathcal{X}$ and prove that it possesses the above-mentioned properties.

A.1. *Definitions*

In this section, we introduce the required notions for defining Operation $\mathcal{X}$. Given a feasible fractional assignment x, we define the following notions:

1. For any two links e, e', a block B is *separating* e, e' if B contains exactly one of them.
2. A block is *tight* if $\sum_{e \in B} x_e$ is equal to either the upper or the lower quota of the constraint corresponding to that block.
3. Given a hierarchy $\mathcal{H}$, a (hard) block $B \in \mathcal{H}$ is *supporting* a pair of links (e, e') if it is the smallest block among the blocks in $\mathcal{H}$ that contain both e, e', and moreover, no tight block in $\mathcal{H}$ separates e, e'.
4. We say that a hierarchy $\mathcal{H}$ is supporting the pair (e, e') if there exists a block in $\mathcal{H}$ that supports (e, e'). In particular, if the subset $\{e, e'\}$ is in the deepest level of $\mathcal{H}$, then (e, e') is supported by $\mathcal{H}$.
5. A *floating cycle* is a sequence $e_1, \ldots, e_l$ of distinct edges such that:

27. Our randomized mechanism stops after at most $|\mathcal{H}| + |E|$ iterations.

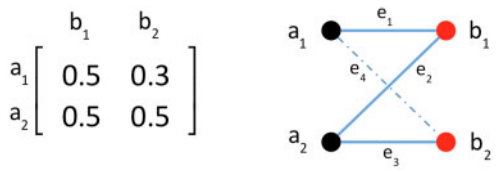

FIGURE A2
Example of a floating path: suppose that in the above fractional assignment $\mathcal{H}_1$ is the set of row blocks and $\mathcal{H}_2$ is the set of column blocks. Also, suppose the lower quotas and upper quotas are set to 0 and 1, respectively. Then, e_1, e_2, e_3 is a (minimal) floating path. However, e_1, e_4, e_3 is *not* a floating path.

- x_{e_i} is non-integral for all integers i,
- (e_i, e_{i+1}) is supported by $\mathcal{H}_1$ for even integers i,
- (e_i, e_{i+1}) is supported by $\mathcal{H}_2$ for odd integers i,

where the *length* of the cycle, l, is an even number and $i+1=1$ for $i=l$. Figure A1 represents a floating cycle of length 6. A floating cycle is said to be *minimal* if it does not contain a smaller floating cycle as a subset. We often drop the minimal phrase and whenever we say a *floating cycle*, we refer to a minimal floating cycle, unless otherwise specified.

Next, we define the notion of *floating paths*; loosely speaking, their structure is very similar to floating cycles, except in their endpoints. Floating paths start from a hierarchy and end in the same hierarchy if their length is even, otherwise, they end in the other hierarchy.

6. A *floating path* is a sequence $e_1, e_1, \ldots, e_l$ of distinct edges such that:

- x_{e_i} is non-integral for all integers i.
- There exists $a \in \{1,2\}$ such that if we define $\overline{a} = \{1,2\} \setminus \{a\}$, then:
  - (e_i, e_{i+1}) is supported by $\mathcal{H}_a$ for even integers $i < l$.
  - (e_i, e_{i+1}) is supported by $\mathcal{H}_{\overline{a}}$ for odd integers $i < l$.
- No tight block in $\mathcal{H}_a$ contains e_1, and no tight block in $\mathcal{H}_b$ contains e_l where $b = a$ if l is even and $b = \overline{a}$ if l is odd.

Figure A2 contains a visual example of a floating path. A floating path is said to be *minimal* if it does not contain a smaller floating path as a subset. Whenever we say a *floating path*, we refer to a minimal floating path, unless otherwise specified.

Finally, we introduce the following crucial concept.

Definition. *Assume we are given a fractional assignment x. For any block B and any $\epsilon > 0$, let $x \uparrow_\epsilon B$ denote a new (fractional) assignment in which the element of the matrix corresponding to edge e is increased by ϵ if $e \in B$ (i.e. it changes to $x_e + \epsilon$), and it remains unchanged otherwise. Similarly, let $x \downarrow_\epsilon B$ denote the fractional assignment in which the element of the matrix corresponding to edge e is decreased by ϵ if $e \in B$ (i.e. it changes to $x_e - \epsilon$), and it remains unchanged otherwise.*

Example A.1 *$(x \uparrow_\epsilon B) \downarrow_\epsilon B'$ denotes the fractional assignment in which the value of any edge $e \in B - B'$ becomes $x_e + \epsilon$, the value of any edge $e \in B' - B$ becomes $x_e - \epsilon$, and the value of the rest of the edges does not change.*

A.2. *Operation $\mathcal{X}$*

Operation $\mathcal{X}$ can be applied on a given floating cycle or a floating path of a fractional assignment x (if none of them exist, then the assignment must be pure by Lemma A.4). We first define this operation for a given floating cycle. Let $F = \langle e_1, \ldots, e_l \rangle$ be a floating cycle in x. Define

$$F_o = \{e_i : i \text{ is odd}\},$$

$$F_e = \{e_i : i \text{ is even}\}.$$

We call the pair (F_o, F_e) the *odd-even decomposition* of F. Given two non-negative reals ϵ, ϵ' (which we describe how to set soon), Operation $\mathcal{X}$ generates an assignment $x' \in \mathbb{R}^{N \times O}$ in one of the following ways:

- $x' = (x \uparrow_\epsilon F_o) \downarrow_\epsilon F_e$ with probability $\frac{\epsilon'}{\epsilon + \epsilon'}$
- $x' = (x \downarrow_{\epsilon'} F_o) \uparrow_{\epsilon'} F_e$ with probability $\frac{\epsilon}{\epsilon + \epsilon'}$.

Both ϵ and ϵ' are chosen to be the largest possible numbers such that both of the assignments $(x \uparrow_\epsilon F_o) \downarrow_\epsilon F_e$ and $(x \downarrow_{\epsilon'} F_o) \uparrow_{\epsilon'} F_e$ remain feasible, in the sense that they satisfy all hard constraints.

The definition of Operation $\mathcal{X}$ on a floating path is the same as its definition on a floating cycle. To summarize, we give a formal definition of Operation $\mathcal{X}$ below.

Definition A.1 *Consider a fractional assignment x and a floating path or a floating cycle, namely F, given as the inputs to Operation $\mathcal{X}$. Then Operation $\mathcal{X}$ generates a new assignment x', where $x' = (x \uparrow_\epsilon F_o) \downarrow_\epsilon F_e$ with probability $\frac{\epsilon'}{\epsilon + \epsilon'}$ and*

$x' = (x\downarrow_{\epsilon'} F_o)\uparrow_{\epsilon'} F_e$ with probability $\frac{\epsilon}{\epsilon+\epsilon'}$, where ϵ, ϵ' are positive numbers chosen to be the largest possible numbers such that both $(x\uparrow_{\epsilon} F_o)\downarrow_{\epsilon} F_e$ and $(x\downarrow_{\epsilon'} F_o)\uparrow_{\epsilon'} F_e$ are feasible assignments.

We also denote x' (which is a random variable) by $x \updownarrow F$.

A.3. *The implementation mechanism*

Our implementation mechanism which is based on Operation $\mathcal{X}$ is formally defined below.

The implementation mechanism based on Operation $\mathcal{X}$:.

1. A fractional assignment x is reported to the mechanism.
2. Set i to 1 and let $x_i = x$.
3. Repeat the following as long as x_i contains a floating cycle or a floating path:
   (a) If x_i contains a floating cycle, let F be an arbitrary floating cycle, otherwise, let F be an arbitrary floating path.
   (b) Define x_{i+1} to be $x_i \updownarrow F$.
   (c) Increase i by one.
4. Report x_i as the outcome of the mechanism.

In the rest of this section, we show that the above mechanism approximately implements x in the sense of Definition 1.

The first step of the proof is verifying that if the assignment has no floating cycles or paths, then it is necessarily pure. We prove this claim in Claim 2. The next step of the proof is to show that Operation $\mathcal{X}$ is well-defined in the sense that both ϵ, ϵ' cannot be zero at the same time. We will state and prove this fact in Lemma A.4. Next, we prove the following three important properties of Operation $\mathcal{X}$:

1. The outcome of Operation $\mathcal{X}$ satisfies the hard constraints.
2. Operatoin $\mathcal{X}$ satisfies the martingale property, *i.e.*
$$\mathbb{E}\left[x \updownarrow F \,\middle|\, x\right] = x$$
3. The outcome of Operation $\mathcal{X}$ has more tight constraints (compared to x).

These properties are proved separately in three Lemmas below.

Lemma A.1 *The outcome of Operation $\mathcal{X}$ satisfies the hard constraints.*

Proof. By definition, Operation $\mathcal{X}$ chooses ϵ, ϵ' such that both of its two possible outcomes are feasible with respect to $\mathcal{H}$. ■

Lemma A.2 *Operation $\mathcal{X}$ satisfies the martingale property, i.e.*
$$\mathbb{E}\left[x \updownarrow F \,\middle|\, x\right] = x.$$

Proof. We prove the lemma by verifying that this property holds for any entry (i,j) of the assignment matrix, *i.e.,* if $(x \updownarrow F)_{(i,j)}$ denotes the (i,j)th element of $x \updownarrow F$, then we have
$$\mathbb{E}\left[(x \updownarrow F)_{(i,j)} \,\middle|\, x\right] = x_{(i,j)}.$$
In simple words, we prove that operation $\mathcal{X}$ does not change the value of entry (i,j) of the assignment matrix in expectation.

Observe that by the definition of Operation $\mathcal{X}$
$$\mathbb{E}\left[x \updownarrow F \,\middle|\, x\right] = \frac{\epsilon'}{\epsilon+\epsilon'} \cdot ((x\uparrow_{\epsilon} F_o)\downarrow_{\epsilon} F_e) + \frac{\epsilon}{\epsilon+\epsilon'} \cdot ((x\downarrow_{\epsilon'} F_o)\uparrow_{\epsilon'} F_e).$$
The claim is trivial if $(i,j) \notin F$. So, assume $(i,j) \in F$. Then, we either have $(i,j) \in F_o$ or $(i,j) \in F_e$:

1. If $(i,j) \in F_o$, then Operation $\mathcal{X}$ increases $x_{(i,j)}$ by ϵ with probability $\frac{\epsilon'}{\epsilon+\epsilon'}$ and decreases it by ϵ' with probability $\frac{\epsilon}{\epsilon+\epsilon'}$. In this case, the expected amount by which $x_{(i,j)}$ changes is equal to $\epsilon \cdot \frac{\epsilon'}{\epsilon+\epsilon'} - \epsilon' \cdot \frac{\epsilon}{\epsilon+\epsilon'} = 0$.
2. If $(i,j) \in F_e$, then Operation $\mathcal{X}$ decreases $x_{(i,j)}$ by ϵ with probability $\frac{\epsilon'}{\epsilon+\epsilon'}$, and increases it by ϵ' with probability $\frac{\epsilon}{\epsilon+\epsilon'}$. In this case, the expected amount by which $x_{(i,j)}$ changes is equal to $-\epsilon \cdot \frac{\epsilon'}{\epsilon+\epsilon'} + \epsilon' \cdot \frac{\epsilon}{\epsilon+\epsilon'} = 0$.

This proves the lemma. ■

Lemma A.3 *The outcome of operation $\mathcal{X}$ has more tight constraints (compared to x).*

Proof. Suppose F is a floating cycle in x. The proof for the path case is almost identical. We show that $x \updownarrow F$ has more tight constraints than x. To do so, we first show that a tight constraint remains tight after Operations $\mathcal{X}$. Second, we show that at least one of the floating constraints in x becomes tight in $x \updownarrow F$.

To prove the first step, we show that for any tight constraint S, its corresponding block, B, contains an equal number of elements (edges) from the sets F_o and F_e. This fact is formally proved below.

Claim 1 *Suppose we are given a floating cycle F in the fractional assignment x, and let (F_o, F_e) be the odd-even decomposition of F. Then, any tight block (in x) contains an equal number of elements from F_o and F_e.*

Proof. Let $S=(B, \underline{q}_B, \bar{q}_B)$ be a tight constraint and w.l.o.g. assume $B \in \mathcal{H}_1$. Then, it must be that for any element $e_i \in B \cap F_e$, the element that comes right after e_i in F, *i.e.,* e_{i+1}, belongs to B. This holds because by the definition of floating cycles, (e_i, e_{i+1}) is supported by $\mathcal{H}_1$, which means no tight block in $\mathcal{H}_1$ separates e_i, e_{i+1}. Consequently, both e_i and e_{i+1} belong to B, or else B itself would separate e_i, e_{i+1}.

Therefore, for any element $e_i \in B \cap F_e$, there exists a distinct element $e_{i+1} \in B \cap F_o$ which corresponds to e_i. Similarly, any element in $B \cap F_o$ corresponds to a distinct element in $B \cap F_e$. This proves the claim. ■

Now recall that whenever Operation $\mathcal{X}$ increases (decreases) the elements in F_o, it decreases (increases) the elements in F_e. This fact and Claim 1 together imply that $x(B) = (x \updownarrow F)(B)$ (regardless of the choice of ϵ, ϵ'). This ensures that any tight constraint remains tight after operation $\mathcal{X}$.

We now prove the second step, which is to show that at least one of the floating constraints in x becomes tight in $x \updownarrow F$. Observe that any floating constraint $S=(B, \underline{q}_B, \bar{q}_B)$ provides a positive *slack* for setting the values of ϵ, ϵ'. In simple words, since S is a floating constraint, we have that $\underline{q}_B < x(B) < \bar{q}_B$. By this fact, we can compute the positive upper bounds that S imposes on ϵ, ϵ'. Finally, taking the minimum of these upper bounds (over all floating constraints S) determines the values for ϵ, ϵ'. We formalize this argument below. Let

$$\bar{s} = \bar{q}_B - x(B),$$

$$\underline{s} = x(B) - \underline{q}_B,$$

$$k = |F_o \cup B| - |F_e \cup B|.$$

Then, in order to guarantee that $x \updownarrow F$ satisfies constraint S, the following inequalities (that can be translated into upper bounds) are imposed on ϵ, ϵ' by Operation $\mathcal{X}$:

$$\begin{cases} \epsilon \cdot k \le \bar{s} & \text{if } k \ge 0 \\ \epsilon \cdot |k| \le \underline{s} & \text{if } k < 0 \end{cases} \tag{A.1}$$

$$\begin{cases} \epsilon' \cdot k \le \underline{s} & \text{if } k \ge 0 \\ \epsilon' \cdot |k| \le \bar{s} & \text{if } k < 0 \end{cases}. \tag{A.2}$$

Now, let $u(S), u'(S)$ respectively denote the (positive) upper bounds imposed by Inequalities (A.1),(A.2) on ϵ, ϵ'. By definition of ϵ, ϵ', we have that $\epsilon = \min_S u(S)$ and $\epsilon' = \min_S u'(S)$ where the minimum is over all the floating constraints S. This argument implies that:

Claim 2 *Operation $\mathcal{X}$ chooses ϵ, ϵ' such that $\epsilon, \epsilon' > 0$.*

Proof. It is enough to show that $u(S), u'(S) > 0$ for all S. This is implied by noting that, given a floating constraint S, we have $\bar{s}, \underline{s} > 0$. ■

The above argument also implies the existence of a floating constraint S_1 for which one of the corresponding inequalities in (A.1) is tight. Similarly, there exists a floating constraint S_2 for which one of the corresponding inequalities in (A.2) is tight. These two facts imply that after operation $\mathcal{X}$, either S_1 or S_2 becomes a tight constraint.

To summarize, we first showed that if a constraint is tight, then it remains tight after operation $\mathcal{X}$. Moreover, we showed that there always exists at least one floating constraint which becomes tight after operation $\mathcal{X}$. Therefore, the number of tight constraints decreases, which proves the lemma. ■

Next, we show that if a fractional assignment contains neither a floating cycle nor a floating path, then it must be a pure assignment. This guarantees that the assignment generated by our implementation mechanism is always pure.

Lemma A.4 *An assignment is pure if and only if it does not contain floating cycles and floating paths.*

Proof. One direction is trivial: if the assignment is pure then it has no floating cycles or floating paths. We prove the other direction by showing that any assignment x which is not pure contains a floating path or a floating cycle. Since x is not pure, it must contain a floating edge e, *i.e.,* an edge e with $0<x_e<1$. We say that a floating edge e is $\mathcal{H}_1$-*loose* ($\mathcal{H}_2$-*loose*) if no tight block in $\mathcal{H}_1$ ($\mathcal{H}_2$) contains e. We say that e is *loose* if it is either $\mathcal{H}_1$-loose or $\mathcal{H}_2$-loose.

We need another definition before presenting the proof. Suppose $S=(B,\underline{q}_B,\bar{q}_B)$ is a tight hard constraint and e is a floating edge in B. Since S is tight, and since the quotas $\underline{q}_B,\bar{q}_B$ are integral, then B must also contain another floating edge e'. We denote this edge by $p(e,B)$. If there is more than one such edge, then let $p(e,B)$ denote one of them arbitrarily.

The proof has two cases, either there is a floating edge which is loose, or there is no such edge.

Case 1: There exists a loose edge. As the first step of the proof, note that we are done if there exists a floating edge which is both $\mathcal{H}_1$-loose and $\mathcal{H}_2$-loose: the edge would form a floating path of length 1. So, w.l.o.g. suppose there is a floating edge e which is not $\mathcal{H}_2$-loose. In this case, we iteratively construct a floating path that starts from edge e, *i.e.*, a path $F=\langle e_1,\ldots,e_l\rangle$ such that $e_1=e$. At the end, our iterative construction will either find such a path, or we will find a floating cycle.

Since e_1 is not $\mathcal{H}_2$-loose, then there must be a minimal tight block $B^1\in\mathcal{H}_2$ that contains e_1. Since B^1 is tight, and since the quotas are integral, then B^1 must also contain another floating edge $p(e_1,B^1)$. We extend our (under construction) floating path by setting $e_2=p(e_1,B^1)$. Now, if e_2 is $\mathcal{H}_1$-loose, then $\langle e_1,e_2\rangle$ is a floating path and the proof is complete. So, suppose e_2 is not $\mathcal{H}_1$-loose. Consequently, there must be a minimal tight block $B^2\in\mathcal{H}_1$ that contains e_2. Similar to before, B^2 must contain another floating edge $p(e_2,B^2)$; we extend F by setting $e_3=p(e_2,B^2)$.

By repeating this argument, we can extend F iteratively until the new floating edge that is added to F, namely e_k, either (i) is loose, or (ii) is contained in one of the previous tight blocks $B^1,\ldots,B^{k-1}$. If case (i) happens, then F is a floating path and we are done. If case (ii) happens, then we have found a floating cycle: suppose $e_k\in B_j$ with $j<k$. Then, it is straight-forward to verify that $\langle e_{j+1},\ldots,e_k\rangle$ is a floating cycle.

Case 2: There is no loose edge. Similar to Case 1, we iteratively construct a floating cycle $F=\langle e_1,\ldots,e_l\rangle$. The cycle starts from a floating edge e; initially, we have $e_1=e$. Since e_1 is not loose, there must be minimal tight blocks $B^0\in\mathcal{H}_1$ and $B^1\in\mathcal{H}_2$ such that $e_1\in B^0$ and $e_1\in B^1$. Then, let $e_2=p(e_1,B^1)$. Similarly, since e_2 is not loose, there must be a tight block $B^2\in\mathcal{H}_1$ such that $e_2\in B^2$. Let $e_3=p(e_2,B^2)$. By applying this argument repeatedly, we can extend F until the new floating edge that is added to F, namely e_k, satisfies $e_k\in B_j$ for some j with $0\leq j<k$. Then, it is straight-forward to verify that $\langle e_{j+1},\ldots,e_k\rangle$ is a floating cycle. ■

A.4. *Approximate satisfaction of soft constraints*

Here, we prove that soft constraints are approximately satisfied in the sense of Definition 1. Loosely speaking, Operation $\mathcal{X}$ is designed in a way such that it never increases (or decreases) two (or more) elements of a soft constraint at the same iteration. Consequently, elements of each soft constraint become "negatively correlated." This allows us to employ Chernoff concentration bounds to prove that soft constraints are approximately satisfied.

We show the approximate satisfaction of soft constraints by proving two lemmas below. In the first lemma, we formally (define and) prove that elements of each soft constraint are "negatively correlated"; the proof uses a negative correlation proof technique from Khuller *et al.* (2006). Then, in the second lemma, we prove the approximate satisfaction of soft constraints by applying Chernoff concentration bounds. Before stating the lemmas, we recall the definition of negative correlation.

Definition A.2 *For an index set B, a set of binary random variables $\{X_e\}_{e\in B}$ are negatively correlated if for any subset $T\subseteq B$ we have*

$$\Pr\left[\prod_{e\in T}X_e=1\right]\leq\prod_{e\in T}\Pr[X_e=1], \tag{A.3}$$

$$\Pr\left[\prod_{e\in T}(1-X_e)=1\right]\leq\prod_{e\in T}\Pr[X_e=0]. \tag{A.4}$$

Lemma A.5 *Let $\{X_e\}_{e\in E}$ denote the set of random variables which represent the outcome of the implementation mechanism (i.e. the integral assignment); also, let B be a block corresponding to an arbitrary soft constraint. Then, the set of random variables $\{X_e\}_{e\in B}$ are negatively correlated.*

Proof. We need to show that (A.3) and (A.4) hold for any subset $T\subseteq B$. We fix an arbitrary subset T and prove (A.3) for it; the proof for (A.4) is identical and follows by replacing the role of zeros and ones. Since the random variables are binary, we can prove (A.3) by showing that

$$\mathbb{E}\left[\prod_{e\in T} X_e\right] \leq \prod_{e\in T} \mathbb{E}[X_e] = \prod_{e\in T} x_e. \tag{A.5}$$

To prove (A.5), we introduce a set of random variables $\{X_{e,i}\}$, where $X_{e,i}$ denotes the value of entry e of the matrix after the ith execution of operation $\mathcal{X}$. So we would have $X_{e,0}=x_e$ for all e. Inductively, we show that for all i:

$$\mathbb{E}\left[\prod_{e\in T} X_{e,i+1}\right] \leq \mathbb{E}\left[\prod_{e\in T} X_{e,i}\right]. \tag{A.6}$$

The lemma is proved if (A.6) holds: assuming that operation $\mathcal{X}$ is executed j times, using (A.6) we can write

$$\mathbb{E}\left[\prod_{e\in T} X_e\right] = \mathbb{E}\left[\prod_{e\in T} X_{e,j}\right] \leq \mathbb{E}\left[\prod_{e\in T} X_{e,0}\right] = \prod_{e\in T} x_e$$

which shows (A.5) holds and proves the lemma.

To prove (A.6), we can alternatively show that

$$\mathbb{E}\left[\prod_{e\in T} X_{e,i+1} \,\middle|\, \{X_{e,i}\}_{e\in T}\right] \leq \prod_{e\in T} X_{e,i}. \tag{A.7}$$

We consider three cases to prove (A.7): since B is in the deepest level of a hierarchy, then operation $\mathcal{X}$ changes either 0, 1, or 2 elements of T. We prove this fact in a separate claim below.

Claim 3 *Suppose T is a block in the deepest level of a hierarchy, then, Operation $\mathcal{X}$ changes either 0, 1, or 2 elements of T.*

Proof. W.L.O.G. assume that T is in the deepest level of $\mathcal{H}_1$. We prove a stronger claim. Let T' be the largest subset of links that contains T and is in the deepest level of $\mathcal{H}_1$. We prove that Operation $\mathcal{X}$ changes at most 2 elements of T'. To this end, let F be the floating cycle or path used in Operation $\mathcal{X}$. We need to show that F contains at most 2 elements of T'; this proves the claim.

For contradiction, suppose F contains at least 3 elements of T'. Let the elements of F be denoted by the sequence $e_1,\ldots,e_l$, and let e_i,e_j,e_k be the first three elements of T' which appear in F, where $i<j<k$.

First, note that by the definitions of floating cycle and floating path, we must have that $j=i+1$. We will prove that $\langle e_j,e_{j+1}\ldots,e_{k-1},e_k\rangle$ makes a floating cycle, which contradicts with the minimality of F (recall that by definition, operation $\mathcal{X}$ always chooses minimal floating paths and cycles). To this end, first note that (e_j,e_{j+1}) is supported by $\mathcal{H}_2$: this holds because $e_{j-1},e_j\in T'$, which means (e_{j-1},e_j) is supported by $\mathcal{H}_1$. Consequently, (e_j,e_{j+1}) must be supported by $\mathcal{H}_2$ since F is a floating path or cycle. Similarly, (e_{j+1},e_{j+2}) is supported by $\mathcal{H}_1$, (e_{j+2},e_{j+3}) is supported by $\mathcal{H}_2$, and so on and so forth. Finally, note that (e_k,e_j) is supported by $\mathcal{H}_1$, since $e_k,e_j\in T'$. This proves that $\langle e_j,e_{j+1}\ldots,e_{k-1},e_k\rangle$ is a floating cycle, which concludes the claim. ■

We continue the proof of lemma by considering each of the three cases separately. The proof is trivial if Operation $\mathcal{X}$ changes 0 elements of T: (A.7) holds with equality. So, it remains to consider the two other cases.

First, assume that Operation $\mathcal{X}$ changes exactly one element of T, namely $e'\in T$. Let $T'=T\backslash\{e'\}$. Then we have

$$\begin{aligned}&\mathbb{E}\left[\prod_{e\in T} X_{e,i+1} \,\middle|\, \{X_{e,i}\}_{e\in T}\right]\\ &= \frac{\epsilon'}{\epsilon+\epsilon'}\cdot(X_{e',i}+\epsilon)\cdot\prod_{e\in T'} X_{e,i} + \frac{\epsilon}{\epsilon+\epsilon'}\cdot(X_{e',i}-\epsilon')\cdot\prod_{e\in T'} X_{e,i} = \prod_{e\in T} X_{e,i}\end{aligned}$$

which proves (A.7) with equality in this case. It remains to prove (A.7) for the case when Operation $\mathcal{X}$ changes exactly 2 elements of T, namely $e',e''\in T$. Let $T''=T\setminus\{e',e''\}$. Then, w.l.o.g. we can write:

$$\begin{aligned}
&\mathbb{E}\left[\prod_{e\in T}X_{e,i+1}\,\middle|\,\{X_{e,i}\}_{e\in T}\right]\\
&=\frac{\epsilon'}{\epsilon+\epsilon'}\cdot(X_{e',i}+\epsilon)(X_{e'',i}-\epsilon)\cdot\prod_{e\in T''}X_{e,i}+\frac{\epsilon}{\epsilon+\epsilon'}\cdot(X_{e',i}-\epsilon')(X_{e'',i}+\epsilon')\cdot\prod_{e\in T''}X_{e,i}\\
&=\prod_{e\in T}X_{e,i}-\epsilon\epsilon'\cdot\prod_{e\in T''}X_{e,i}\\
&\leq\prod_{e\in T}X_{e,i}
\end{aligned}$$

which proves (A.7) in the third case as well. This finishes the proof of lemma. ■

Lemma A.6 *The randomized mechanism based on Operation $\mathcal{X}$ satisfies the soft constraints approximately in the sense of Definition 1.*

Proof. Based on Definition 1, we need to prove that for any soft constraint defined on a block B of the links with $\sum_{e\in B}w_ex_e=\mu$, and for any $\epsilon>0$, we have

$$\Pr\left(\sum_{e\in B}w_eX_e-\mu<-\epsilon\mu\right)\leq e^{-\mu\frac{\epsilon^2}{2}},$$

$$\Pr\left(\sum_{e\in B}w_eX_e-\mu>\epsilon\mu\right)\leq e^{-\mu\frac{\epsilon^2}{3}}.$$

These probabilistic bounds, as we mentioned before, are known as Chernoff concentration bounds (see Supplementary Appendix D for more details). These bounds hold on any set of binary random variables which are negatively correlated (Popovici, 2014). Lemma A.5 just says that the set of random variables $\{X_e\}_{e\in B}$ are negatively correlated, which means Chernoff concentration bounds hold for $\{X_e\}_{e\in B}$. ■

B. REMAINING PROOFS AND EXAMPLES FROM SECTION 3

B.1. *Tightness of the probabilistic bounds*

Proof. Proof of Proposition 1 Fix an interval $I=[a,b]$ such that $3\leq a\leq b-1$. For any $\mu\geq 1$ and any constant $\epsilon\in(0,1)$, we construct an infinite family of problem instances. For the rest of the proof, we fix μ,ϵ. The infinite family of instances, $\mathcal{F}$, is indexed by a variable n, which denotes the number of agents involved in each instance. For any integer $n\geq\mu^3$, $\mathcal{F}$ contains one instance.[28] This instance contains a set of n agents, $N=\{1,\ldots,n\}$, and one object. The capacity of the object will be larger than 1 and is determined shortly when we specify the set of hard constraints. The variables $x_1,\ldots,x_n$ denote the assignment of agent i to the object. Note that, by definition, $0\leq x_i\leq 1$ must hold for all i, in both pure and fractional assignments.

Choose $k\in I$ such that μk is an integer. Let $A=\mu k$. Consider the fractional assignment that assigns $1/k$ to all variables, *i.e.*, $x_i=1/k$ for all $i\in N$. Define the hard–soft partitioned constraint structure as

$$\mathcal{H}=\left\{\lfloor n/k\rfloor\leq\sum_{i\in N}x_i\leq\lceil n/k\rceil\right\},$$

$$\mathcal{S}=\left\{\sum_{i\in S}x_i\geq\mu:\forall S\subseteq N,|S|=A\right\}.$$

We denote this assignment by x.

28. The condition $n\geq\mu^3$ could be replaced with $n\geq f(\mu)$ for any function $f(\mu)$ that grows faster than μ^2.

Our goal is showing that any integer assignment that satisfies the hard constraints violates at least $|\mathcal{S}|\cdot e^{-\frac{\epsilon^2\mu}{d}}$ of the soft constraints, where $d>0$ is a constant independent of ϵ,μ, this would imply that $f(\mu,\epsilon)\geq e^{-\frac{\epsilon^2\mu}{d}}$. Hence, setting c to be any constant smaller than d would prove the proposition.

Let x^* denote the outcome of the lottery that implements x with respect to $\mathcal{E}=\mathcal{H}\cup\mathcal{S}$. We should have $x_i^*=1$ for at most $\lceil n/k\rceil$ different elements $i\in N$; let S^* denote the set of all such elements. For notational simplicity, from now on we suppress the ceiling notation and treat n/k as an integer. (This simplifies the algebraic expressions; the proof remains the same.)

A set $S\subseteq N$ with $|S|=A$ is *feasible* if $|S\cap S^*|>\mu(1-\epsilon)$ and *infeasible* otherwise. Observe that the infeasible sets correspond to the soft constraints that are not approximately satisfied. Next, we will provide a lower bound on the number of infeasible sets. More precisely, let p denote the ratio of the number of infeasible sets to $|\mathcal{S}|$. Observe that

$$p\geq\frac{\binom{n(1-1/k)}{A(1-\frac{1-\epsilon}{k})}\binom{n/k}{\frac{A}{k}\cdot(1-\epsilon)}}{\binom{n}{A}}. \tag{B.8}$$

To simplify the above bound, we use the following fact.

Fact *I Das (2016)* *When $s=o(\sqrt{t})$ and $s=\omega(1)$,*[29]

$$\binom{t}{s}=\frac{1}{\sqrt{2\pi s}}\left(\frac{te}{s}\right)^s(1+o(1)).$$

Applying this fact to the numerator and denominator of (B.8) implies:

$$\begin{aligned}
p&=\frac{\frac{\left(\frac{n(1-1/k)e}{A(1-\frac{1-\epsilon}{k})}\right)^{A(1-\frac{1-\epsilon}{k})}(1+o(1))}{\sqrt{2\pi A(1-\frac{1-\epsilon}{k})}}\cdot\frac{\left(\frac{ne/k}{A(1-\epsilon)/k}\right)^{A(1-\epsilon)/k}(1+o(1))}{\sqrt{2\pi A(1-\epsilon)/k}}}{\frac{(\frac{ne}{A})^A(1+o(1))}{\sqrt{2\pi A}}}\\
&\geq\left(\frac{1-1/k}{1-\frac{1-\epsilon}{k}}\right)^{A(1-\frac{1-\epsilon}{k})}\cdot\left(\frac{1}{1-\epsilon}\right)^{A(1-\epsilon)/k}\cdot\frac{1+o(1)}{\sqrt{2\pi A}(1+o(1))}\\
&=\left(\frac{\left(1-\frac{\epsilon}{k-1+\epsilon}\right)^{k-1+\epsilon}}{(1-\epsilon)^{1-\epsilon}}\right)^{A/k}\cdot\frac{1+o(1)}{\sqrt{2\pi A}(1+o(1))}\\
&\geq\left(\frac{\left(e^{-\frac{\epsilon}{k-1+\epsilon}-\left(\frac{\epsilon}{k-1+\epsilon}\right)^2}\right)^{k-1+\epsilon}}{e^{-\epsilon(1-\epsilon)}}\right)^{A/k}\cdot\frac{1+o(1)}{\sqrt{2\pi A}(1+o(1))} \qquad \text{(B.9)}\\
&=\left(\frac{e^{-\epsilon-\frac{\epsilon^2}{k-1+\epsilon}}}{e^{-\epsilon(1-\epsilon)}}\right)^{A/k}\cdot\frac{1+o(1)}{\sqrt{2\pi A}(1+o(1))}\\
&=e^{-\epsilon^2\left(1+\frac{1}{k-1+\epsilon}\right)A/k}\cdot\frac{1+o(1)}{\sqrt{2\pi A}(1+o(1)),} \qquad \text{(B.10)}
\end{aligned}$$

where (B.9) holds since $e^{-\delta-\delta^2}\leq 1-\delta\leq e^{-\delta}$ holds for all $\delta\in[0,1/2]$. Note that the lower order terms above, which are suppressed by the $o(1)$ notation, vanish as μ approaches infinity, for any fixed $\epsilon>0$.

The proof is complete by observing that the right-hand side of (B.10) is larger than $e^{-\frac{\epsilon^2A/k}{d}}$ for any positive $d\leq 2/3$ and sufficiently large A (*i.e.* sufficiently large μ, since $A=\mu k$). ■

29. We recall that for two functions $f,g:\mathbb{R}_+\to\mathbb{R}_+$, $f=\omega(g)$ denotes $\lim_{x\to\infty}f(x)/g(x)=\infty$. Also, we write $g=o(f)$ when $f=\omega(g)$.

B.2. *Probabilistic guarantees for general soft constraints*

Proof. Proof of Proposition 2 By assumption, at least one of the $\mathcal{H}_1$ or $\mathcal{H}_2$ is empty. Without loss of generality, suppose $\mathcal{H}_1=\emptyset$. We add a "dummy" constraint to $\mathcal{H}_1$, which contains all the elements, *i.e.*, the constraint $0\leq\sum_{e\in E}x_e<\infty$. Clearly, any soft constraint block is in the deepest level of $\mathcal{H}_1$. Hence, by Theorem 1, $\mathcal{E}$ is approximately implementable. ■

Proof. Proof of Proposition 3 For simplicity we only give the proof for upper deviation, *i.e.*, for the probabilistic bound (3.4). The proof for (3.5) is similar. Since B has depth k, it can be partitioned into k blocks $B_1,\ldots,B_k$ all of which are in the deepest level of $\mathcal{H}$. In order to provide a guarantee on the satisfaction of the soft constraint corresponding to B, we add k constraints, one for each of $B_1,\ldots,B_k$, to our soft constraint set. The (soft) constraint corresponding to block B_i, denoted by C_i, would be

$$\sum_{e\in B_i}X_e\leq\mu_i,$$

where $\mu_i=\sum_{e\in B_i}x_e$. Since C_i is in the deepest level of $\mathcal{H}$, the following guarantee would hold on X, the outcome of our mechanism: (by Theorem 1)

$$\Pr(\mathsf{dev}_i^+\geq\epsilon_i\mu_i)\leq e^{-\mu_i\frac{\epsilon_i^2}{3}},$$

where ϵ_i can be any positive number and

$$\mathsf{dev}_i^+=\max\big(0,\sum_{e\in B_i}X_e-\mu_i\big).$$

The key is to define ϵ_i's such that

$$e^{-\mu_i\frac{\epsilon_i^2}{3}}=e^{-\mu\frac{\epsilon^2}{3k}},\tag{B.11}$$

$$\sum_{i=1}^k\epsilon_i\mu_i\leq\epsilon\mu.\tag{B.12}$$

If these two properties hold, then a union bound on the constraints $C_1,\ldots,C_k$ would prove the claim: by (B.11), the probability that (at least) one of the constraints C_i is violated with (additive) error more than $\epsilon_i\mu_i$ is at most $ke^{-\mu\frac{\epsilon^2}{3k}}$. On the other hand, if all constraints C_i are satisfied with (additive) error not more than $\epsilon_i\mu_i$, then using (B.12) we get:

$$\mathsf{dev}^+\leq\sum_{i=1}^k\mathsf{dev}_i^+\leq\sum_{i=1}^k\epsilon_i\mu_i\leq\epsilon\mu.\tag{B.13}$$

This would prove the claim. So, to finish the proof, it only remains to define ϵ_i's such that (B.11) and (B.12) would hold. To this end, define $\alpha_i=k\mu_i/\mu$ and let $\epsilon_i=\epsilon/\sqrt{\alpha_i}$. It is straight-forward to verify that this definition satisfies (B.11). To see that (B.12) also holds, we rewrite its left-hand side as follows:

$$\sum_{i=1}^k\epsilon_i\cdot\mu_i=\sum_{i=1}^k\frac{\epsilon}{\sqrt{\alpha_i}}\cdot\frac{\alpha_i\mu}{k}=\frac{\epsilon\mu}{k}\cdot\sum_{i=1}^k\sqrt{\alpha_i}\leq\epsilon\mu,$$

where in the last inequality uses the fact that $\sum_{i=1}^k\alpha_i=k$, which implies $\sum_{i=1}^k\sqrt{\alpha_i}\leq k$. The above inequality shows that (B.12) holds; this completes the proof. ■

B.3. *Proof of Theorem 2*

Denote the set of all types by $\mathcal{T}=\{1,\ldots,T\}$, and let $N(t)$ denote the set of students of type $t\in T$. Suppose we are given a fractional assignment which is feasible with respect to the constraint set $(\mathcal{E},\mathbf{q})$. We will show how to approximately implement x with additive error k, in the sense of Definition 2. In other words, we will construct a lottery with a (random) outcome X such that X satisfies the conditions in Definition 2.

To design this lottery, we first need to define a new hard structure, namely $\mathcal{H}'$, as follows. For each type $t\in T$ and each school $c\in O$, $\mathcal{H}'$ contains a hard block $\{x_{(s,c)}:s\in N(t)\}$. For each block $B=\{x_{(s,c)}:s\in N(t)\}$ belonging to $\mathcal{H}'$, define its corresponding lower and upper quotas to be $\underline{q}_B=\lfloor\sum_{s\in N(t)}x_{(s,c)}\rfloor$ and $\bar{q}_B=\lceil\sum_{s\in N(t)}x_{(s,c)}\rceil$.

Since $\mathcal{H}'$ is a hierarchy, and since any block in $\mathcal{H}'$ is in the deepest level of $\mathcal{H}_2$, then $\mathcal{H}_2\cup\mathcal{H}'$ is a hierarchy as well. Therefore, $\mathcal{H}_1\cup(\mathcal{H}_2\cup\mathcal{H}')$ is a bihierarchy. Hence, we can use Theorem 1 to implement x using a lottery such that the outcome of the lottery satisfies all of the "old" hard constraints as well as all of the "new" ones, *i.e.*, all of the hard

constraints corresponding to $\mathcal{H}_1 \cup \mathcal{H}_2$, and all of the hard constraints corresponding to $\mathcal{H}'$, respectively. We let X to be the outcome of this lottery. Theorem 1 implies that $\mathbb{E}[X] = x$ must hold. In the rest of the proof, we will show that X satisfies any soft constraint in $\mathcal{S}$ with additive error at most k. This will complete the proof of the theorem. Consider a soft constraint in $\mathcal{S}$ corresponding to a block B. We write such a constraint as

$$\underline{q}_B \leq \sum_{t \in \mathcal{T}(B)} \sum_{s \in N(t)} X_{(s,c)} \leq \overline{q}_B, \tag{B.14}$$

where $c \in O$ is a school and

$$\mathcal{T}(B) = \{t : \text{ there exists } s \in N \text{ such that } (s,c) \in B \text{ and } t = T(s)\},$$

i.e., $\mathcal{T}(B)$ denotes the set of types which are "involved" in the block B. Observe that

$$\left| \sum_{t \in \mathcal{T}(B)} \sum_{s \in N(t)} x_{(s,c)} - \sum_{t \in \mathcal{T}(B)} \sum_{s \in N(t)} X_{(s,c)} \right|$$

$$\leq \sum_{t \in \mathcal{T}(B)} \left| \sum_{s \in N(t)} x_{(s,c)} - \sum_{s \in N(t)} X_{(s,c)} \right| \leq |\mathcal{T}(B)|,$$

where the last inequality follows from (B.14). The fact that $|\mathcal{T}(B)| < T$ concludes the proof. ■

B.4. *Impossibility result for fully general structures*

The following example shows that without any structure on soft constraints, guaranteeing small errors is impossible. Let $N = \{1, \ldots, n\}$ and $O = \{1, \ldots, n\}$. Consider the following constraints: agent i wants to have exactly one of the objects $i, i+1$ (where for notational simplicity we have assumed $i+1 = 1$ when $i = n$), and each object has capacity 1, *i.e.*, there is only one copy of each object. These constraint can be modelled by a set of hard bihierarchical constraints, $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$, as follows:

$$\mathcal{H}_1 = \{x_{(i,i)} + x_{(i,i+1)} \leq 1\}_{i=1,\ldots,n},$$

$$\mathcal{H}_2 = \{x_{(i,i)} + x_{(i-1,i)} \leq 1\}_{i=1,\ldots,n},$$

where again for notational simplicity we have assumed $i - 1 = n$ when $i = 1$. Also, we define the following soft constraint:

$$\lfloor n/2 \rfloor \leq \sum_{i=1}^{n} x_{(i,i)} \leq \lceil n/2 \rceil.$$

Observe that the fractional assignment defined by

$$\bar{x}_{(i,i)} = \bar{x}_{(i,i+1)} = \frac{1}{2}, \qquad \forall i = 1, \ldots, n$$

satisfies all of the hard and soft constraints. However, any lottery that implements $\bar{x}$ and satisfies the hard constraints must severely violate the soft constraint by an additive factor of at least $\lfloor n/2 \rfloor$, as we show next.

First, observe that there exists a unique convex combination of pure assignments which is equal to $\bar{x}$ and it is defined by $\bar{x} = 0.5\bar{y} + 0.5\bar{z}$ where y, z are defined as follows:

$$\bar{y}_{(i,i)} = 1, \bar{y}_{(i,i+1)} = 0, \qquad \forall i = 1, \ldots, n$$

$$\bar{z}_{(i,i)} = 0, \bar{z}_{(i,i+1)} = 1, \qquad \forall i = 1, \ldots, n.$$

So, the outcome of the unique lottery that implements $\bar{x}$ must be $\bar{y}$ with probability 0.5 and $\bar{z}$ otherwise. In both of these cases, the soft constraint gets violated (*ex post*) by an additive factor of at least $\lfloor n/2 \rfloor$.

B.5. *The effect of negative correlation on the error bounds*

In this section, we construct an example to illustrate why our implementation method can provide better probabilistic guarantees in the presence of negative correlation rather than independence.

Let $N = \{1, \ldots, 2n\}$ denote a set of agents and $O = \{c_1, c_2\}$ denote a set of objects. We call each agent a student and each object a school. Consider a fractional assignments x where $x_{io} = 1/2$ for all $i \in N$ and $o \in O$. We would like to implement this fractional assignment by designing a lottery over pure assignments. The only hard constraint that should hold *ex post* in the lottery outcome is that each student must be assigned to precisely one school.

Each school has one soft constraint that needs to hold *ex post*, defined as follows. Precisely n of the students are *blue*, and the rest are *red*. The soft constraint of each school is admitting at most $n/2$ blue students. Let the random variable B denote the total number of blue students admitted to school o_1. Therefore, the error in satisfying the soft constraint of school o_1 is $\max\{0, B-n\}$.

In what follows, we will compare three different methods for implementing x. By symmetry, we compare these methods with respect to the approximate satisfaction of the soft constraint of o_1. We will use $\text{Var}[B]$ as an intuitive notion to rank these methods: the larger the variance, the larger probabilities of violation will be. To see why, note that the random variable B is *approximately* a Normal random variable with mean n, for sufficiently large n.[30] (By definition, $\mathbb{E}[B]=n$ must hold in any implementation method.) Therefore, the smaller the variance, the smaller the errors in satisfying the soft constraint will be.

The first implementation method is based on the idea of *independent rounding* of random variables, and the second and the third ones are based on the idea of *dependent rounding*.

Implementation with independence. Assign each student i to a school that is chosen independently and uniformly at random. In this case, observe that $\text{Var}[B]=n/2$.

Implementation with positive correlation. Fiping a single coin: if heads is observed, then all blue students are assigned to school o_1, and otherwise, they are assigned to school o_2. In this case, $\text{Var}[B]=n^2$.

Implementation with negative correlation. Flip a coin for each pair of students $(2i-1, 2i)$, for $i \in \{1,\dots,n\}$: if heads is observed, student $2i-1$ is assigned to school o_1 and student $2i$ to school o_2, otherwise, student $2i-1$ is assigned to school o_2 and student $2i$ to school o_2. Under this implementation method $\text{Var}[B]=n/4+o(n)$. To see why, let X denote the number of pairs $(2i-1, 2i)$ such that precisely one of the students involved in the pair is blue. Observe that $\text{Var}[B|X]=X/4$. On the other hand, since the set of blue students is distributed uniformly at random, we have that $X \leq n/2+o(n)$, with probability at least $1-1/n^2$ for sufficiently large n. (This is implied by Chernoff bounds.) Therefore, $\text{Var}[B]=n/8+o(n)$

We remark that the implementation method that features negative correlation, in fact, coincides with the implementation method of Theorem 1, if the capacity constraints of the schools are defined as hard constraints. We see that, among all of the implementation methods above, the one with the negative correlation property leads to a smaller $\text{Var}[B]$, and therefore a better *ex post* guarantee for (approximately) satisfying the soft constraint.

The intuition is that, in the third implementation method, for some of the pairs $(2i-1, 2i)$, both of the involved students have the same colour. In such pairs, for any blue student assigned to a school, a blue student will be assigned to the other school. (Intuitively, this is the source of the negative correlation property.) If all of the pairs satisfy this property, then an equal number of blue students would be assigned to each school and the soft constraints will be (strictly) satisfied. Although this does not hold for all of the pairs, it does hold for a significantly large number of them (about half of them). This reduces the variance compared to the implementation method with independent random variables and leads to better probabilistic guarantees for satisfying the soft constraint.

REFERENCES

ABDULKADIROĞLU, A., PARAG, P. A., and ROTH, A. E. (2005), "The New York City High School Match", *American Economic Review*, **95**, 364–367.

ABDULKADIROĞLU, A., PATHAK, P. A., ROTH, A. E., *et al*. (2005), "The Boston Public School Match", *American Economic Review*, **95**, 368–371.

ABDULKADIROĞLU, A. and SÖNMEZ, T. (1998), "Random Serial Dictatorship and the Core from Random Endowments in House Allocation Problems", *Econometrica*, **66**, 689–701.

ABDULKADIROĞLU, A. and SÖNMEZ, T. (2003), "School Choice: A Mechanism Design Approach", *American Economic Review*, **93** 729–747.

AGEEV, A. A. and SVIRIDENKO, M. I. (2004), "Pipage Rounding: A New Method of Constructing Algorithms with Proven Performance Guarantee", *Journal of Combinatorial Optimization*, **8**, 307–328.

ALON, N. and SPENCER, J. H. (2004), *The Probabilistic Method* (John Wiley & Sons).

ASHLAGI, I., SABERI, A., and SHAMELI, A. (2019), "Assignment Mechanisms under Distributional Constraints", in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms* (SIAM) 229–240. https://dl.acm.org/doi/10.5555/3310435.3310450.

30. The Normal approximation turns out to be a sharp approximation for sum of a large number of Bernoulli random variables.

BABAIOFF, M., NISAN, N., and TALGAM-COHEN, I. (2017), "Competitive Equilibrium with Indivisible Goods and Generic Budgets." In Workshop on Matching under Preferences (MATCH-UP).
BIRKHOFF, G. (1946), "Three Observations on Linear Algebra", *Universidad Nacional de Tucumd.n, Revista, Serie A*, **5**, 147–151. (In Spanish).
BOGOMOLNAIA, A. and MOULIN, H. (2001), "A New Solution to the Random Assignment Problem", *Journal of Economic Theory*, **100**, 295–328.
BORDER, K. C. (1991), "Implementation of Reduced Form Auctions: A Geometric Approach", *Econometrica*, **59**, 1175–1187.
BRONFMAN, S., ALON, N., HASSIDIM, A., *et al.* (2018), "Redesigning the Israeli Medical Internship Match", *ACM Transactions on Economics and Computation (TEAC)*, **6**, 2167–8375.
BROOME, J. (1972), "Approximate Equilibrium in Economies with Indivisible Commodities", *Journal of Economic Theory*, **5**, 224–249.
BUDISH, E. (2011), "The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes", *Journal of Political Economy*, **119**, 1061–1103.
BUDISH, E., CACHON, G. P., KESSLER, J. B. *et al.* (2016), "Course Match: A Large-Scale Implementation of Approximate Competitive Equilibrium from Equal Incomes for Combinatorial Allocation", *Operations Research*, **65**, 314–336.
BUDISH, E., CHE, Y.-K., KOJIMA, F., and MILGROM, P. (2013), "Designing Random Allocation Mechanisms: Theory and Applications", *American Economic Review* **103**, 585–623.
CHE, Y.-K., KIM, J., and KOJIMA, F. (2019), "Stable Matching in Large Economies", *Econometrica*, **87**, 65–110.
CHE, Y.-K., KIM, J., and MIERENDORFF, K. (2013), "Generalized Reduced-form Auctions: A Network-flow Approach", *Econometrica*, **81**, 2487–2520.
CHE, Y.-K. and KOJIMA, F. (2010), "Asymptotic Equivalence of Probabilistic Serial and Random Priority Mechanisms", *Econometrica*, **78**, 1625–1672.
CHE, Y.-K. and TERCIEUX, O. (2019), "Efficiency and Stability in Large Matching Markets", *Journal of Political Economy*, **127**, 2301–2342.
CHEKURI, C., VONDRAK, J., and ZENKLUSEN, R. (2010), "Dependent Randomized Rounding via Exchange Properties of Combinatorial Structures", *Foundations of Computer Science (FOCS)*, 2010:51.
CHEN, Y. and SONMEZ, T. (2002), "Improving Efficiency of On-campus Housing: An Experimental Study", *The American Economic Review*, **92**, 1669–1686.
DAS, S. (2016), "A Brief Note on Estimates of Binomial Coefficients" http://page.mi.fu-berlin.de/shagnik/notes/binomials.pdf.
DIERKER, E. (1971), "Equilibrium Analysis of Exchange Economies with Indivisible Commodities", *Econometrica: Journal of the Econometric Society*, **39**, 997–1008.
EDMONDS, J. (2003), *Submodular Functions, Matroids, and Certain Polyhedra* (Berlin Heidelberg: Springer Berlin, Heidelberg) 11–26.
EHLERS, L., HAFALIR, I. E., YENMEZ, M. B. *et al.* (2014), "School Choice with Controlled Choice Constraints: Hard Bounds versus Soft Bounds", *Journal of Economic Theory*, **153**, 648–683.
FRAGIADAKIS, D. and TROYAN, P. (2017), "Improving Matching under Hard Distributional Constraints", *Theoretical Economics*, **12**, 863–908.
GANDHI, R., KHULLER, S., PARTHASARATHY, S., and SRINIVASAN, A. (2006), "Dependent Rounding and Its Applications to Approximation Algorithms", *Journal of the ACM (JACM)*, **53**, 324–360.
HAFALIR, I. E., YENMEZ, M. B., and YILDIRIM, M. A. (2013), "Effective Affirmative Action in School Choice", *Theoretical Economics*, **8**, 325–363.
HASHIMOTO, T. (2018), "The Generalized Random Priority Mechanism with Budgets", *Journal of Economic Theory*, **177**, 708–733.
HATFIELD, J. W. (2009), "Strategy-proof, Efficient, and Nonbossy Quota Allocations", *Social Choice and Welfare*, **33**, 505–515.
HENRY, C. (1970), "Indivisibilités Dans Une Economie D'Echanges", *Econometrica*, **38**, 542–558.
HOU, M., SODOMKA, E., and STIER-MOSES, N. (2016), "Game Abstractions for Counterfactual Prediction in Online Markets", (Mimeo),
HYLLAND, A. and ZECKHAUSER, R. (1979), "The Efficient Allocation of Individuals to Positions", *Journal of Political Economy*, **87**, 293–314.
JACOBS, L. A. (2004), *Pursuing Equal Opportunities: The Theory and Practice of Egalitarian Justice* (Cambridge University Press).
KAMADA, Y. and KOJIMA, F. (2015), "Efficient Matching under Distributional Constraints: Theory and Applications", *American Economic Review*, **105**, 67–99.
KAMADA, Y. and KOJIMA, F. (2019), "Fair Matching under Constraints: Theory and Applications" (Working Paper).
KESTEN, O., KURINO, M., and NESTEROV, A. S. (2017), "Efficient Lottery Design", *Social Choice and Welfare*, **48**, 31–57.
KHULLER, S., PARTHASARATHY, S., and SRINIVASAN, A. (2006), "Dependent Rounding and Its Applications to Approximation Algorithms", *Journal of the ACM*, **53**, 324–360.
KOJIMA, F. (2009), "Random Assignment of Multiple Indivisible Objects", *Mathematical Social Sciences*, **57**, 134–142.
KOJIMA, F. and MANEA, M. (2010), "Incentives in the Probabilistic Serial Mechanism", *Journal of Economic Theory*, **145**, 106–123.

KOMINERS, S. D. and SÖNMEZ, T. (2016), "Matching with Slot-łspecific Priorities: Theory", *Theoretical Economics* **11**, 683–710.
LAU, L. C., RAVI, R., and SINGH, M. (2011), *Iterative Methods in Combinatorial Optimization* (Vol. 46) (Cambridge: Cambridge University Press).
LIU, Q., and PYCIA, M. (2016) *Ordinal Efficiency, Fairness, and Incentives in Large Markets (August 1, 2016)*. https://ssrn.com/abstract=1872713.
MANEA, M. (2009), "Asymptotic Ordinal Inefficiency of Random Serial Dictatorship", *Theoretical Economics*, **4**, 165–197.
MAS-COLELL, A. (1977), "Indivisible Commodities and General Equilibrium Theory", *Journal of Economic Theory*, **16**, 443–456.
MATTHEWS, S. A. (1984), "On the Implementability of Reduced Form Auctions", *Econometrica*, **52**, 1519–1522.
NGUYEN, T., PEIVANDI, A., and VOHRA, R. (2016), "Assignment Problems with Complementarities", *Journal of Economic Theory*, **165**, 209–241
NGUYEN, T. and VOHRA, R. (2018), "Near-Feasible Stable Matchings with Couples", *American Economic Review*, **108**, 3154–3169.
NGUYEN, T. and VOHRA, R. (2019), "Stable Matching with Proportionality Constraints", *Operations Research* **67**, 1503–1519.
NYCDOE (2019), "New York City Department of Education Reports" https://ibo.nyc.ny.us/iboreports/. Accessed: 2019-05-11.
PATHAK, P. A. and SETHURAMAN, J. (2011), "Lotteries in Student Assignment: An Equivalence Result", *Theoretical Economics*, **6**, 1–17.
POPOVICI, E. (2014), "Anne Auger and Benjamin Doerr (eds): Theory of Randomized Search Heuristics: Foundations and Recent Developments", *Genetic Programming and Evolvable Machines*, **15**, 111–112.
PYCIA, M. and ÜNVER, M. U. (2015), "Decomposing Random Mechanisms", *Journal of Mathematical Economics*, **61**, 21–33.
RUBINSTEIN, A. (2014), "Inapproximability of Nash Equilibrium" Preprint, arXiv.
SCHRIJVER, A. and COOK, W. J. (1997), *Combinatorial Optimization.* https://onlinelibrary.wiley.com/doi/book/10.1002/9781118033142.
STANFORD ENCYCLOPEDIA OF PHILOSOPHY (2013). *Affirmative action*. https://plato.stanford.edu/entries/affirmative-action/.
VON NEUMANN, J. (1953), "A Certain Zero-Sum Two-Person Game Equivalent to the Optimal Assignment Problem", *Contributions to the Theory of Games*, **2**, 5–12.