Constructing Small Sample Spaces Satisfying Given Constraints*

Daphne Koller†
e-mail: daphne@cs.berkeley.edu

Nimrod Megiddo‡
e-mail: megiddo@almaden.ibm.com

Abstract

The subject of this paper is finding small sample spaces for joint distributions of n discrete random variables. Such distributions are often only required to obey a certain limited set of constraints of the form $\Pr(Event) = \pi$. We show that the problem of deciding whether there exists any distribution satisfying a given set of constraints is NP-hard. However, if the constraints are consistent, then there exists a distribution satisfying them which is supported by a "small" sample space (one whose cardinality is equal to the number of constraints). For the important case of *independence constraints*, where the constraints have a certain form and are consistent with a joint distribution of *independent* random variables, a small sample space can be constructed in polynomial time. This last result can be used to derandomize algorithms; we demonstrate this by an application to the problem of finding large independent sets in sparse hypergraphs.

AMS subject classification: 60E05, 68R99.

Keywords: discrete probability distribution, linear programming, algorithm, sample space, derandomization, hypergraph, independent set, probabilistic constraint satisfaction, independent random variables.

1. Introduction

The probabilistic method of proving existence of combinatorial objects has been very successful (see, for example, Raghavan [16] and Spencer [18]). The underlying idea is as follows. Consider a finite set Ω whose elements are classified as "good" and "bad". Suppose we wish to prove existence of at least one "good" element within Ω. The proof proceeds by constructing a probability distribution f over Ω and showing that the probability of picking a good element is positive. Probabilistic proofs often yield randomized algorithms for constructing a good

*A preliminary version of this paper appeared in *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, 1993. Research supported in part by ONR Contract N00014-91-C-0026 and by the Air Force Office of Scientific Research (AFSC), under Contract F49620-91-C-0080. The United States Government is authorized to reproduce and distribute reprints for governmental purposes.

†Computer Science Division, University of California, Berkeley, CA 94720; and IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120.

‡IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120; and School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel.

element. In particular, many randomized algorithms are a special case of this technique, where the "good" elements are those sequences of random bits leading to a good answer.

It is often desirable to replace the probabilistic construction by a deterministic one, or to *derandomize* an algorithm. Obviously, this can be done by completely enumerating the sample space Ω until a good element is found.[1] Unfortunately, the sample space is typically exponential in the size of the problem; for example, the sample space of n independent random bits[2] contains 2^n points.

Let $X_1, \ldots, X_n$ be discrete random variables with a finite range. For simplicity, we assume that $X_1, \ldots, X_n$ all have the same range $\{0, \ldots, r-1\}$ (although not necessarily the same distribution); our constructions can easily be extended to variables with different ranges. The *probability space* associated with these variables is $\Omega = \{0, \ldots, r-1\}^n$. A *distribution* is a map $f : \Omega \to [0, 1]$ such that $\sum_{\boldsymbol{x} \in \Omega} f(\boldsymbol{x}) = 1$. We define the set $S(f) = \{\boldsymbol{x} \in \Omega \mid f(\boldsymbol{x}) > 0\}$ to be the *sample space of* f.

Given a distribution f involved in a probabilistic proof, only the points in $S(f)$ need to be considered in our search for a good point in Ω. Moreover, it suffices to search any subset of $S(f)$ that is guaranteed to contain a good point for each possible input. Adleman [1] shows that for any distribution f used in an algorithm in RP, there exists a space $S' \subseteq S(f)$ of polynomial size that contains a good point for every possible input. The proof of this fact is not constructive, and therefore cannot be used to derandomize algorithms.

A common technique for constructing a feasible search-space is to find a different distribution with a "small" (polynomial) sample space that can be searched exhaustively, as outlined above. The new distribution must agree with the original one sufficiently so that the correctness proof of the algorithm remains valid. The correctness proof often relies on certain assumptions about the distribution; that is, the distribution is assumed to satisfy certain constraints. A *constraint* is an equality of the form

$$\Pr(Q) = \sum_{\boldsymbol{x} \in Q} f(\boldsymbol{x}) = \pi \ ,$$

where $Q \subseteq \Omega$ is an *event* and $0 \leq \pi \leq 1$. If the randomness requirements of an algorithm are completely describable as a set of constraints, and the new distribution satisfies all of them, then the algorithm remains correct under the new distribution; no new analysis is needed. In other cases, the new distribution may only approximately satisfy the constraints, and it is necessary to verify that the analysis still holds.

The original distribution is almost always constructed based on *independent* random variables $X_1, \ldots, X_n$. Thus, all the constraints are satisfied by such a distribution. In many cases, however, full independence is not necessary. In particular, quite often the constraints are satisfied by a *d-wise independent distribution*—a distribution where each *neighborhood* of d variables

[1] This can be done if we assume that good elements are easy to recognize. In decision problems, this is usually the case. In optimization problems, we may be able to prove that a random element is optimal or close to optimal with a certain probability. In those cases, although we may not be able to tell by looking at an element if it is "good", we can often compare elements and decide which is "better". We can therefore derandomize such an algorithm by enumerating the sample space and choosing the "best" element in it. The techniques of this paper also apply to problems of this type.

[2] We use the term *random bits* to denote binary-valued uniformly-distributed random variables.

behaves as if it were independent. That is, it suffices for the distribution to satisfy the *independence constraints* that state that every event defined over a neighborhood of size d has the same probability as if the variables were independent. Most of the previous work has focussed on constructing approximations to such distributions.

Joffe [10] first demonstrated a construction of a joint distribution of n d-wise independent uniformly-distributed random variables with a sample space of cardinality $O((2n)^d)$. Luby [13] and Alon, Babai, and Itai [3] show how Joffe's construction can be generalized to allow for non-uniform distributions using sample spaces of essentially the same cardinality. In many cases, the resulting distributions only approximately satisfy the required constraints; that is, the distributions are d-wise independent, but the probabilities $\Pr(X_i = b)$ may differ from the corresponding probabilities in the original distribution.[3] These constructions result in sample spaces of polynomial size for any fixed d. Chor *et. al* [8] showed that any sample space of n d-wise independent random bits has cardinality $\Omega(n^{\lfloor d/2 \rfloor})$. Thus, these constructions are close to optimal in this case. Moreover, sample spaces of polynomial size exist for d-wise independent distributions only if d is fixed.

Naor and Naor [15] showed how to circumvent this lower bound by observing that ϵ-*independent* (or *nearly independent*) distributions often suffice. In other words, it suffices that the *independence constraints* for the neighborhoods of size d be satisfied to within ϵ. We point out that this is also a form of approximation, as defined above. Naor and Naor demonstrated a construction of sample spaces for ϵ-independent distributions over random bits, whose size is polynomial in n and in $1/\epsilon$. These constructions are polynomial for $\epsilon = 1/poly(n)$; for such values of ϵ, the ϵ-independence constraints are meaningful[4] for subsets of size up to $O(\log n)$. Therefore, we obtain a polynomial-size sample space that is nearly d-wise independent for $d = O(\log n)$ (as compared to the lower bound of $\Omega(n^{\log n})$ for truly d-wise independent sample spaces). Simplified constructions with similar properties were provided by Alon *et. al* [4]. Azar, Motwani, and Naor [5] later generalized these techniques to uniform distributions over non-binary random variables. Finally, Even *et. al* [9] presented constructions for nearly independent distributions over non-uniform non-binary random variables.

A different type of technique was introduced by Berger and Rompel [7] and by Motwani, Naor, and Naor [14]. This technique can be used to derandomize certain RNC algorithms where d, the degree of independence required, is polylogarithmic in n. The technique works, however, only for certain types of problems, and does not seem to generalize to larger degrees of independence.

Schulman [17] took a different approach towards the construction of sample spaces that require $O(\log n)$-wise independence. He observed that in many cases, only certain *d-neighborhoods* (sets of d variables) must be independent. Schulman constructs sample spaces satisfying this property whose size is 2^d times the greatest number of neighborhoods to which any variable belongs. In particular, for polynomially many neighborhoods each of size $O(\log n)$, this con-

[3] In fact, these distributions all have a sample space of cardinality $O(p^d)$ for some prime $p \geq n$. The approximation is better for larger p's.

[4] Consider a distribution over random bits, and some subset of k of the variables. The "correct" probability of any event prescribing values to all the variables in this subset is $1/2^k$. For $k = \log(1/\epsilon) = \Theta(\log n)$, this probability is $\leq \epsilon$. For larger k, all such constraints are therefore subsumed by constraints corresponding to smaller subsets of the variables.

struction results in a polynomial-size sample space. His construction works only for random bits, and is polynomial for a maximum neighborhood size $O(\log n)$.

In order to improve on these results, we view the problem from a somewhat different perspective. Instead of placing upper bounds on the degree of independence required by the algorithm, we examine the set of precise constraints that are required in order for the algorithm to work. We then construct a distribution satisfying these constraints exactly. In many cases, this approach yields a much smaller sample space.

We begin by showing a connection between the number of constraints and the size of the resulting sample space. We show in Section 2 that for any set $\mathcal{C}$ of such constraints, if $\mathcal{C}$ is *consistent*, *i.e.*, $\mathcal{C}$ is satisfied by some distribution f, then there exists a distribution f' also satisfying $\mathcal{C}$ such that $|S(f')| \leq |\mathcal{C}|$. That is, there exists a distribution for which the cardinality of the sample space is not more than the number of constraints. Note that f' precisely satisfies the constraints in $\mathcal{C}$, so that if $\mathcal{C}$ represents all the assumptions about f made by a proof, the proof will also hold for f'. The proof of the existence theorem includes an algorithm for constructing f'; however, the algorithm takes exponential time and is thus not useful. We justify this exponential behavior by showing that even for a set of very simple constraints, the problem of recognizing whether there exists a distribution satisfying them is NP-complete.

We can, however, define a type of constraint for which a small sample space can be constructed directly from the constraints in polynomial time. As we observed, the distributions that are most often used in probabilistic proofs are ones where $X_1, \ldots, X_n$ are independent random variables. Such a distribution is determined entirely by the probabilities $\{p_{ib} = \Pr(X_i = b) \ : \ i = 1, \ldots, n; \ b = 0, \ldots, r-1\}$. In the course of such a probabilistic proof, the distribution is assumed to satisfy various constraints. Above, we observed that in many cases, and in particular in all cases for which existing constructions work, these constraints are independence constraints. More formally, an independence constraint is one that forces the probability of a certain assignment of values to some subset of the variables to be as if the variables are independent. That is, for a fixed set of p_{ib}'s, a sequence of indices $i_1, \ldots, i_k$ in $\{1, \ldots, n\}$, and $b_1, \ldots, b_k \in \{0, \ldots, r-1\}$, the constraint

$$\left[\Pr(\{X_{i_1} = b_1, \ldots, X_{i_k} = b_k\}) = \prod_{j=1}^{k} p_{i_j b_j} \right]$$

is the *independence constraint* $I(Q)$ corresponding to the event[5] $Q = \{X_{i_1} = b_1, \ldots, X_{i_k} = b_k\}$. Obviously, if $X_1, \ldots, X_n$ are independent random variables then their joint distribution satisfies all the independence constraints. Note that d-wise independence can easily be represented in terms of constraints of this type: the variables $X_1, \ldots, X_n$ are d-wise independent if and only if all the independence constraints $I(\{X_{i_1} = b_1, \ldots, X_{i_d} = b_d\})$ are satisfied, where $i_1, \ldots, i_d \in \{1, \ldots, n\}$ and $b_1, \ldots, b_d \in \{0, \ldots, r-1\}$.

Let $\mathcal{C}$ be a set of independence constraints defined using a fixed set of p_{ib}'s as above. In Section 3 we present the main result of this paper, that shows how to construct in strongly polynomial time a distribution satisfying $\mathcal{C}$ with a sample space of cardinality $|\mathcal{C}|$. We note

[5] Throughout this paper, we assume without loss of generality that $i_1 < i_2 < \ldots < i_k$, and regard this notation as a shorthand for the event $\{(x_1, \ldots, x_n) \ : \ x_{i_1} = b_1, \ldots, x_{i_k} = b_k\}$.

that the distribution f produced by our technique is typically not the uniform distribution over $S(f)$. Therefore, we cannot in general use our construction to reduce the number of random bits required to generate the desired distribution.

Our construction has a number of advantages. First, the distributions generated always satisfy the constraints precisely. Thus, the correctness proof of the algorithm need not be modified. Moreover, the size of the sample space in all the nearly independent constructions [4, 5, 9, 15] depends polynomially on $1/\epsilon$ (where ϵ is the approximation factor). Our precise construction does not have this term. Previously, precise distributions were unavailable for many interesting distributions. In particular, our approach can construct sample spaces of cardinality $O((rn)^d)$ for any set of n r-valued, d-wise independent random variables (not necessarily uniformly distributed). For fixed d, this construction requires polynomial time. It has been argued by Even *et. al* [9] that probability distributions over non-uniform non-binary random variables are important. To our knowledge, this is the first technique that allows the construction of exact distributions of d-wise independent variables with arbitrary p_{ib}'s.

The main advantage of our construction is that the size of the sample space depends only on the number of constraints actually used. Except for Schulman's approach [17], all other sample spaces are limited by requiring that all neighborhoods of a particular size be independent (or nearly independent). As Schulman points out, in many cases only certain neighborhoods are ever relevant, thus enabling a further reduction in the size of the sample space. However, Schulman's approach still requires the sample space to satisfy all the independence constraints associated with the relevant neighborhoods.[6] This restricts his construction to neighborhoods of maximal size $O(\log n)$. With our construction we can deal with neighborhoods of any size, as long as the number of relevant constraints is limited.

For example, an algorithm may randomly choose edges in a graph by associating a binary random variable with each edge. An event whose probability may be relevant to the analysis of this algorithm is "no edge adjacent to a node v is chosen". Using the other approaches (even Schulman's), the neighborhood size would be the maximum degree Δ of a node in the graph; the relevant sample space would then grow as 2^{Δ}. Using our approach, there is only one event per node, resulting in a sample space of size n (the number of nodes in the graph).

In this example, the constraints depend on the edge structure of the input graph. In general, our construction depends on the specific constraints derived from the particular input. Therefore, unlike most sample space constructions, our construction cannot be prepared in advance. This property, combined with the fact that our algorithm is sequential, means that it cannot be used to derandomize parallel (RNC) algorithms.

In Section 4 we show an example of how our technique can be applied to derandomization of algorithms. We discuss the problem of finding a large independent set in a d-uniform hypergraph. The underlying randomized algorithm, described by Alon, Babai, and Itai [3], was derandomized in the same paper for fixed values of d. It was later derandomized also for $d = O(\mathit{polylog}\ n)$ by Berger and Rompel [7] and Motwani, Naor, and Naor [14]. We show how this algorithm can be derandomized for any d. A sequential deterministic polynomial time solution for the independent set problem in hypergraphs exists [2]. However, the derandomization of this algorithm using our technique serves to demonstrate its unique power.

[6] Moreover, as we have observed, Schulman's construction works only for random bits.

Algorithm 1: Reduction to Basic Solutions

While $\{\boldsymbol{A}_{*j} : j \in S(\boldsymbol{v})\}$ are linearly dependent:

1. Find a nonzero vector $\boldsymbol{u} \in \mathbb{R}^m$ such that:
   $u_j = 0$ for every $j \notin S(\boldsymbol{v})$, and
   $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{0}$.
2. Find some $t \in \mathbb{R}$ such that:
   $\boldsymbol{v} + t\boldsymbol{u} \geq \boldsymbol{0}$, and
   $v_j + tu_j = 0$ for some $j \in S(\boldsymbol{v})$.
3. Replace $\boldsymbol{v} \leftarrow \boldsymbol{v} + t\boldsymbol{u}$.

2. Existence of small sample spaces

Let $\mathcal{C} = \{[\Pr(Q_k) = \pi_k] \ : \ k = 1, \ldots, c\}$ be a set of constraints such that $[\Pr(\Omega) = 1] \in \mathcal{C}$. From here on, the term "polynomial" means polynomial in terms of $n, r, |\mathcal{C}|$, and the bit lengths of the π_k's.

Definition 2..1. A set $\mathcal{C}$ of constraints is *consistent* if there exists some distribution f satisfying all the members of $\mathcal{C}$.

Definition 2..2. A distribution f that satisfies $\mathcal{C}$ is said to be *manageable* if $|S(f)| \leq c = |\mathcal{C}|$.

Theorem 2..3. *If $\mathcal{C}$ is consistent, then $\mathcal{C}$ is satisfied by a manageable distribution.*

Proof: Let $\mathcal{C}$ be as above, and recall that $c = |\mathcal{C}|$. We describe a distribution f satisfying $\mathcal{C}$ as a non-negative solution to a set of linear equations. Let $\boldsymbol{\pi} \in \mathbb{R}^c$ denote the vector $(\pi_k)_{k=1,\ldots,c}$. Recall that $\Omega = \{0, \ldots, r-1\}^n$; let $m = |\Omega| = r^n$, and let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ denote the points of Ω. The variable v_ℓ will represent the probability $f(\boldsymbol{x}_\ell)$. Let $\boldsymbol{v}$ be the vector $(v_\ell)_{\ell=1,\ldots,m}$. A constraint $\Pr(Q_k) = \pi_k$ can be represented as the linear equation

$$\sum_{\ell=1}^{m} a_{k\ell} v_\ell = \pi_k \ ,$$

where

$$a_{k\ell} = \begin{cases} 1 & \text{if } \boldsymbol{x}_\ell \in Q_k \\ 0 & \text{otherwise .} \end{cases}$$

Thus, the constraints in $\mathcal{C}$ can be represented by a system $\boldsymbol{A}\boldsymbol{v} = \boldsymbol{\pi}$ of linear equations (where $\boldsymbol{A}$ is the matrix $(a_{k\ell})$). Since $\mathcal{C}$ is assumed to be consistent, there is a distribution f satisfying $\mathcal{C}$. Therefore, for $v_\ell = f(\boldsymbol{x}_\ell)$, the vector $\boldsymbol{v}$ is a nonnegative solution to this system. A classical theorem in linear programming asserts that under these conditions, there exists a basic solution to this system. That is, there exists a vector $\boldsymbol{v}' \geq \boldsymbol{0}$ such that $\boldsymbol{A}\boldsymbol{v}' = \boldsymbol{\pi}$ and the columns $\boldsymbol{A}_{*j}$ such that $v'_j > 0$ are linearly independent. Let f' be the distribution corresponding to this solution vector $\boldsymbol{v}'$. Since the number of rows in the matrix is c, the number of linearly independent columns is also at most c. Therefore, the number of positive indices in $\boldsymbol{v}'$, which is precisely $|S(f')|$, is at most $c = |\mathcal{C}|$. ∎

This theorem can be proven constructively based on the standard algorithm outlined above. This algorithm begins with a distribution vector $\boldsymbol{v}$, and removes points from the sample space one at a time. The removal is done while keeping all variables non-negative, so that the truth of

the equations is maintained. This results in a manageable distribution vector $\boldsymbol{v}'$. Throughout the algorithm, $S(\boldsymbol{v})$ denotes the set of indices $\{j : v_j > 0\}$. Intuitively, these indices represent points in the sample space of the distribution represented by $\boldsymbol{v}$.

Algorithm 1 is described in full detail by Beling and Megiddo [6]. They show that it requires $O(|S(f)| \cdot c^2)$ arithmetic operations, assuming that f is represented sparsely (so that points not in $S(f)$ need not be considered at all).[7] However, Beling and Megiddo also present a faster algorithm for the same problem, based on fast matrix multiplication. Given a matrix multiplication algorithm that multiplies two $k \times k$ matrices using $O(k^{2+\delta})$ arithmetic operations, the algorithm of Beling and Megiddo finds a basic solution in $O(c^{\frac{3-\delta}{2-\delta}}|S(f)|)$ arithmetic operations. Using the best known algorithm for matrix multiplication, their algorithm allows us to prove the following:

Theorem 2..4. *Given a distribution f in sparse representation that satisfies the constraints in $\mathcal{C}$, it is possible to construct a manageable distribution f' satisfying the same constraints using $O(|S(f)| \cdot c^{1.62})$ arithmetic operations.*

Unfortunately, the complexity of this approach is linear in $|S(f)|$, which can be as large as $m = r^n$. The algorithm is therefore exponential in n in the worst case.[8]

The exponential behavior of these algorithms can be justified by considering the problem of deciding whether a given set of constraints $\mathcal{C}$ is consistent; that is, does there exist a distribution f satisfying the constraints in $\mathcal{C}$? For arbitrary constraints, the representation of the events can be very long, causing the input size to be unreasonably large. We therefore restrict attention to *simple constraints*.

Definition 2..5. We say that a constraint $\Pr(Q) = \pi$ is *k-simple* if there exist $i_1, \ldots, i_k \in \{1, \ldots, n\}$ and $b_1, \ldots, b_k \in \{0, \ldots, r-1\}$ such that $Q = \{X_{i_1} = b_1, \ldots, X_{i_k} = b_k\}$. A constraint is *simple* if it is k-simple for some k.

Note that the natural representation of the event as a simple constraint requires space which is at most linear in n, whereas the number of points in the event is often exponential in n (for example, a 1-simple constraint contains r^{n-1} points). We assume throughout that simple constraints are represented compactly (in linear space). Under this assumption, we can show that the consistency problem is NP-hard, even when restricted to 2-simple constraints over binary-valued random variables.

Proposition 2..6. *The problem of recognizing whether a set $\mathcal{C}$ of 2-simple constraints is consistent is* NP*-hard, even if the variables constrained by $\mathcal{C}$ are binary-valued.*

Proof: The proof uses a reduction from the 3-colorability problem: given a graph $G = (V, E)$, decide if there exists a legal coloring $\gamma : V \to \{1, 2, 3\}$. Let G be a graph, and assume that $V = \{v_1, \ldots, v_n\}$. We define a set of $3n$ binary-valued variables $\{X_{i,1}, X_{i,2}, X_{i,3} : i = 1, \ldots, n\}$. Intuitively, we would like it to be the case that $\gamma(v_i) = c$ iff $X_{i,c} = 1$ and $X_{i,b} = 0$ for $b \neq c$; for example, $\gamma(v_i) = 2$ iff $X_{i,1} = X_{i,3} = 0$ and $X_{i,2} = 1$. We will construct $\mathcal{C}$ so that the constraints enforce this relationship. The set $\mathcal{C}$ contains constraints of two types:

- For each $i = 1, \ldots, n$ and $b \neq c \in \{1, 2, 3\}$, $\mathcal{C}$ contains the constraints:

$$\Pr(\{X_{i,b} = 0, X_{i,c} = 0\}) \quad = \quad 1/3$$

[7] If f is not represented sparsely, it obviously requires exponential time simply to read it in.

[8] The manageable distribution can also be computed directly from the constraints using a linear programming algorithm that computes basic solutions. The running time of such an algorithm will also be exponential in n.

$$\begin{aligned} \Pr(\{X_{i,b} = 0, X_{i,c} = 1\}) &= 1/3 \\ \Pr(\{X_{i,b} = 1, X_{i,c} = 0\}) &= 1/3 \ . \end{aligned}$$

Intuitively, these disallow illegal colorings, where the same node gets two colors.

- For each $(v_i, v_j) \in E$ and each $b \in \{1, 2, 3\}$, $\mathcal{C}$ contains the constraints:

$$\begin{aligned} \Pr(\{X_{i,b} = 0, X_{j,b} = 0\}) &= 1/3 \\ \Pr(\{X_{i,b} = 0, X_{j,b} = 1\}) &= 1/3 \\ \Pr(\{X_{i,b} = 1, X_{j,b} = 0\}) &= 1/3 \ . \end{aligned}$$

Intuitively, these disallow colorings where two adjacent nodes get the same color.

All the constraints in $\mathcal{C}$ are clearly 2-simple. We now prove that $\mathcal{C}$ is consistent iff G is 3-colorable.

Assume that $\mathcal{C}$ is consistent, and let f be some distribution satisfying $\mathcal{C}$. Consider the probability

$$f(\{X_{i,1} = 1, X_{i,2} = 0, X_{i,3} = 0\}) = f(\{X_{i,1} = 1, X_{i,2} = 0\}) - f(\{X_{i,1} = 1, X_{i,2} = 0, X_{i,3} = 1\}).$$

The latter probability is at most $f(\{X_{i,1} = 1, X_{i,3} = 1\})$, which by the constraints of the first type is 0. Therefore,

$$f(\{X_{i,1} = 1, X_{i,2} = 0, X_{i,3} = 0\}) = f(\{X_{i,1} = 1, X_{i,2} = 0\}) = 1/3 \ .$$

Similar reasoning allows us to conclude that $f(\{X_{i,1} = 0, X_{i,2} = 1, X_{i,3} = 0\}) = f(\{X_{i,1} = 0, X_{i,2} = 0, X_{i,3} = 1\}) = 1/3$, so that $f(\{X_{i,1} = 0, X_{i,2} = 0, X_{i,3} = 0\}) = 0$. Now, pick some arbitrary point $\boldsymbol{x} \in S(f)$, and define $\gamma(v_i)$ to be b iff $x_{i,b} = 1$. Due to the reasoning above, there is a unique such b for every i, so that this defines a coloring of the graph. Now, consider any edge $(v_i, v_j) \in E$, and assume by contradiction that $\gamma(v_i) = \gamma(v_j) = b$. Then, $x_{i,b} = x_{j,b} = 1$, so that $f(\{X_{i,b} = 1, X_{j,b} = 1\}) \geq f(\boldsymbol{x}) > 0$, violating a constraint of the second type. Therefore, γ is a well-defined legal coloring.

Now, assume that there exists a legal coloring γ of G. Let $\pi^1, \ldots, \pi^6$ be the six permutations of $\{1, 2, 3\}$. We define f to be the uniform distribution over six points $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^6$: for $k = 1, \ldots, 6$, $i = 1, \ldots, n$, and $b = 1, 2, 3$, we define $x^k_{i,b} = 1$ iff $\pi^6(\gamma(v_i)) = b$. It is simple to verify, by straightforward symmetry considerations, that the resulting distribution f satisfies all the constraints in $\mathcal{C}$. ∎

In order to prove a matching upper bound, we again need to make a simple assumption about the representation of the input.

Definition 2..7. An event Q is said to be *polynomially checkable* if membership of any point $\boldsymbol{x} \in \Omega$ in Q can be checked in polynomial time.

Proposition 2..8. *If all the constraints in $\mathcal{C}$ pertain to polynomially checkable events, then the consistency of $\mathcal{C}$ can be decided in non-deterministic polynomial time.*

Proof: The algorithm guesses a subset $T \subseteq \Omega$ of cardinality $|\mathcal{C}|$. It then constructs in polynomial time a system of equations corresponding to the constraints in $\mathcal{C}$ restricted to the variables in T (the other variables are set to 0). Given the initial guess, this system can be constructed in polynomial time, since for each constraint and each point in T it takes polynomial time to check whether the point appears in the constraint. The algorithm then attempts to find a nonnegative solution to this system. Such a solution exists if and only if there exists a manageable distribution whose sample space is (contained in) T. By Theorem 2..3, we know that a set of constraints is consistent if and only if it is satisfied by a manageable distribution; that is, a distribution over some sample space T of cardinality not greater than $|\mathcal{C}|$. Therefore, $\mathcal{C}$ is consistent if and only if one of these subsystems has a nonnegative solution. ∎

Since simple constraints are always polynomially checkable (using the appropriate representation), we obtain the following corollary.

Corollary 2..9. *For an arbitrary set $\mathcal{C}$ of simple constraints, the problem of recognizing the consistency of $\mathcal{C}$ is* NP*-complete.*

3. Independence constraints

An important special case was already discussed in the introduction. Suppose all the members of $\mathcal{C}$ are independence constraints arising from a known[9] fixed set of values

$$\{p_{ib} \ : \ i = 1, \ldots, n; \ b = 0, \ldots, r-1\},$$

where p_{ib} represents $\Pr(\{X_i = b\})$, and therefore $\sum_{b=0}^{r-1} p_{ib} = 1$ for all i and $p_{ib} \geq 0$ for all i, b. In this case, we can construct in strongly polynomial time a manageable distribution satisfying $\mathcal{C}$. We note that the distribution we construct does not necessarily satisfy the additional constraints that $\Pr(\{X_i = b\}) = p_{ib}$. If it is necessary that these constraints be satisfied, they must be put explicitly into $\mathcal{C}$.

We first define the concept of a *projected event*. Consider an event

$$Q = \{X_{i_1} = b_1, \ldots, X_{i_k} = b_k\} \ .$$

Let ℓ $(1 \leq \ell \leq n)$ be an integer and denote by $q = q(\ell)$ the maximal index such that $i_q \leq \ell$. The *ℓ-projection* of Q is defined as

$$\Pi_\ell(Q) = \{X_{i_1} = b_1, \ldots, X_{i_k} = b_q\} \ .$$

Intuitively, the ℓ-projection of a constraint is its restriction to the variables $X_1, \ldots, X_\ell$. For example, if Q is $\{X_1 = 0, X_4 = 1, X_7 = 1\}$, then $\Pi_3(Q) = \{X_1 = 0\}$ and $\Pi_4(Q) = \{X_1 = 0, X_4 = 1\}$. Analogously, we call $I(\Pi_\ell(Q))$ the ℓ-projection of the constraint $I(Q)$. Finally, for a set of independence constraints $\mathcal{C}$, $\Pi_\ell(\mathcal{C})$ is the set of the ℓ-projections of the constraints in $\mathcal{C}$.

We now recursively define a sequence of distributions $f_0, f_1, \ldots, f_n$, such that for each ℓ $(\ell = 0, \ldots, n)$, the following conditions hold:

[9] The assumption that the p_{ib}'s are known is a necessary one; see Theorem 3..3.

1. f_ℓ is a distribution on $\{0,\ldots,r-1\}^\ell$,
2. f_ℓ satisfies $\Pi_\ell(\mathcal{C})$,
3. $|S(f_\ell)| \leq c$.

The distribution f_n is clearly the desired one.

We begin by defining f_0, which is a distribution on $\{0,\ldots,r-1\}^0 = \{()\}$ (the singleton set containing the empty sequence). The only possible definition is:

$$f_0(()) = 1 \ .$$

This clearly satisfies all the requirements.

Now, assume that $f_{\ell-1}$ (for $\ell \geq 1$) satisfies the above requirements, and define an intermediate distribution g_ℓ by:

$$g_\ell(x_1,\ldots,x_{\ell-1},b) = f_{\ell-1}(x_1,\ldots,x_{\ell-1}) \cdot p_{\ell b}$$

for $b = 0,\ldots,r-1$.

Lemma 3..1. *If $f_{\ell-1}$ satisfies $\Pi_{\ell-1}(\mathcal{C})$, then g_ℓ satisfies $\Pi_\ell(\mathcal{C})$.*

Proof: We will prove that g_ℓ satisfies every constraint in $\Pi_\ell(\mathcal{C})$. Let $I(Q)$ be an arbitrary constraint in $\mathcal{C}$, and suppose that $Q = \{X_{i_1} = b_1,\ldots,X_{i_k} = b_k\}$. For simplicity, we denote $Q^j = \Pi_j(Q)$ $(j = 1,\ldots,n)$. Let r be the maximal index such that $i_r \leq \ell - 1$. By the assumption,

$$f_{\ell-1}(Q^{\ell-1}) = \prod_{j=1}^{r} p_{i_j b_j} \ .$$

We distinguish two cases:

Case I: Q mentions the variable X_ℓ. In this case, $i_{r+1} = \ell$, and

$$\begin{aligned} Q^\ell &= \{X_{i_1} = b_1,\ldots,X_{i_r} = b_r, X_{i_{r+1}} = b_{r+1}\} \\ &= \{(x_1,\ldots,x_{\ell-1},b_{r+1}) \ : \ (x_1,\ldots,x_{\ell-1}) \in Q^{\ell-1}\} \ . \end{aligned}$$

Therefore:

$$\begin{aligned} g_\ell(Q^\ell) &= \sum_{(x_1,\ldots,x_{\ell-1})\in Q^{\ell-1}} g_\ell(x_1,\ldots,x_{\ell-1},b_{r+1}) \\ &= \sum_{(x_1,\ldots,x_{\ell-1})\in Q^{\ell-1}} f_{\ell-1}(x_1,\ldots,x_{\ell-1}) \cdot p_{\ell b_{r+1}} \\ &= f_{\ell-1}(Q^{\ell-1}) \cdot p_{\ell b_{r+1}} \\ &= \prod_{j=1}^{r} p_{i_j b_j} \cdot p_{\ell b_{r+1}} = \prod_{j=1}^{r+1} p_{i_j b_j} \ . \end{aligned}$$

Thus, g_ℓ satisfies the constraint $I(Q^\ell)$.

Case II: Q does not mention the variable X_ℓ. In this case,

$$\begin{aligned} Q^\ell &= \{X_{i_1} = b_1,\ldots,X_{i_r} = b_r\} \\ &= \{(x_1,\ldots,x_\ell) \ : \ (x_1,\ldots,x_{\ell-1}) \in Q^{\ell-1}, x_\ell \in \{0,\ldots,r-1\}\} \ . \end{aligned}$$

Therefore:

$$\begin{aligned} g_\ell(Q^\ell) &= \sum_{b\in\{0,\ldots,r-1\}} \sum_{(x_1,\ldots,x_{\ell-1})\in Q^{\ell-1}} g_\ell(x_1,\ldots,x_{\ell-1},b) \\ &= \sum_{b\in\{0,\ldots,r-1\}} \sum_{(x_1,\ldots,x_{\ell-1})\in Q^{\ell-1}} f_{\ell-1}(x_1,\ldots,x_{\ell-1})\cdot p_{\ell b} \\ &= \sum_{b\in\{0,\ldots,r-1\}} p_{\ell b}\cdot f_{\ell-1}(Q^{\ell-1}) \\ &= f_{\ell-1}(Q^{\ell-1}) = \prod_{j=1}^{r} p_{i_j b_j} \ . \end{aligned}$$

Again, g_ℓ satisfies the constraint $I(Q^\ell)$. ∎

If $|S(f_{\ell-1})| \le c$, then $|S(g_\ell)| \le rc$, since each point with positive probability in $S(f_{\ell-1})$ yields at most r points with positive probabilities in $S(g_\ell)$. Thus, g_ℓ satisfies requirements 1 and 2, but may not satisfy requirement 3. But g_ℓ is a nonnegative solution to the system of linear equations defined by $\Pi_\ell(\mathcal{C})$. Therefore, we may use Algorithm 1 or the algorithm of Beling and Megiddo [6] to reduce the cardinality of the sample space to c, as described in Section 2. Let f_ℓ be the resulting distribution. It clearly satisfies all three requirements. We thus obtain the following theorem:

Theorem 3..2. *Given a set of independence constraints, we can construct a manageable distribution f satisfying $\mathcal{C}$ in strongly polynomial time using $O(rnc^{2.62})$ arithmetic operations.*

Proof: The distribution f_n constructed as above is clearly a manageable distribution satisfying $\mathcal{C}$. The construction takes n iterations. Iteration ℓ requires at most $O(rc)$ operations to create g_ℓ from $f_{\ell-1}$. It requires at most $O(|S(g_\ell)|c^{1.62}) = O(rc\cdot c^{1.62}) = O(rc^{2.62})$ arithmetic operations for running the algorithm of Beling and Megiddo to reduce g_ℓ to f_ℓ, as in Theorem 2..4. Therefore, the entire algorithm runs in $O(rnc^{2.62})$ arithmetic operations. The number of operations does not depend on the magnitudes of the numbers in the input. In order to prove that the algorithm is strongly polynomial, it remains to show that the magnitudes of the numbers used in the algorithm are polynomial in the input size. Each distribution f_ℓ is a basic solution to the system of linear equations defined by $\Pi_\ell(\mathcal{C})$. The numbers used in describing this system are 1's, 0's, and products of polynomially many p_{ib}'s. Hence, their magnitudes are all polynomial in the size of the input. Since the numbers in a basic solution to a system always have polynomial length in the size of the system, we conclude that the magnitudes of the numbers in each f_ℓ are polynomial in the size of the input. The intermediate phases—creating $g_{\ell+1}$ and running the algorithm of Beling and Megiddo—do not cause blowup, since the latter is known to be strongly polynomial. ∎

As we mentioned, our algorithm can easily be extended to operate on random variables with ranges of different sizes. Let r_i be the number of values in the range of X_i. The sample space of g_ℓ will consist of vectors $(x_1,\ldots,x_{\ell-1},b)$ where $(x_1,\ldots,x_{\ell-1}) \in S(f_{\ell-1})$ and $b \in \{0,\ldots,r_\ell-1\}$. Then $|S(g_\ell)| \le r_\ell|\mathcal{C}|$. The proof goes through as before, but the number of operations in iteration ℓ is $O(r_\ell c^{2.62})$. The total number of operations is $O((\sum_{\ell=1}^{n} r_\ell)c^{2.62}) = O(rnc^{2.62})$, where $r = \max\{r_1,\ldots,r_n\}$. The cardinality of the resulting sample space is still $|\mathcal{C}|$.

The algorithm can also deal with more general constraints with no change. In particular, it can deal with *combinatorial rectangles*, as described by Even *et. al* [9] and by Linial *et. al* [12].

A combinatorial rectangle is an independence constraint over an event of the form

$$\{X_{i_1} \in R_1, \ldots, X_{i_k} \in R_k\} \ ,$$

where R_j is a subset of $\{0, \ldots, r-1\}$ (or of $\{0, \ldots, r_{i_j} - 1\}$ in the more general case). The proof remains essentially unchanged, except for minor modifications to deal with the fact that the "right" probabilities for the events (their probabilities under the assumption of independence) are different. For example, the probability of the event above would be

$$\prod_{j=1}^{k} (\sum_{b \in R_j} p_{i_j b}) \ .$$

The complexity of the algorithm for this case, and the size of the resulting sample space, remain as in Theorem 3..2. Karger and Koller [11] show that this construction can be further generalized to deal with a far more general class of constraints.

Throughout this section, we have assumed that the p_{ib}'s are known. This assumption is important in view of the following theorem, that states that if this is not the case, it is NP-hard to verify whether all of a given set of constraints are independence constraints.

Theorem 3..3. *It is* NP*-hard to recognize whether for a given set of 2-simple constraints $\mathcal{C}$ there exists a set $P = \{p_{ib}\}$ such that all the members of $\mathcal{C}$ are independence constraints relative to P.*

Proof: As in the proof of Proposition 2..6, we use a reduction from the problem of 3-colorability. In this proof, however, we use different variables and a different set of constraints $\mathcal{C}$. Let $G = (V, E)$ be a graph, with $V = \{v_1, \ldots, v_n\}$. We construct a set of 2-simple constraints over the 3-valued variables $X_1, \ldots, X_n$. These constraints essentially say that the probability that two neighboring vertices get the same color is 0:

$$\mathcal{C} = \{[\Pr(\{X_i = b, X_j = b\}) = 0] \ : \ (v_i, v_j) \in E; \ b \in \{0, 1, 2\}\} \ .$$

We claim that the constraints in $\mathcal{C}$ are independence constraints with respect to some P iff G is 3-colorable. Clearly, if the constraints in $\mathcal{C}_G$ are independence constraints relative to some P, then they are satisfiable. Let f be some distribution satisfying $\mathcal{C}$, and let $\boldsymbol{x}$ be some (arbitrary) point in $S(f)$. Define $\gamma(v_i) = x_i$. If γ is not a legal coloring, then there exists an edge $(v_i, v_j) \in E$ such that $x_i = x_j = a$. But since $f(\boldsymbol{x}) > 0$, necessarily $f(\{X_i = b, X_j = b\}) > 0$, contradicting the assumption that f satisfies $\mathcal{C}$. Assume, on the other hand, that G is 3-colorable, and let γ be an appropriate coloring. Define $p_{ib} = 1$ if $\gamma(v_i) = b$, and $p_{ib} = 0$ otherwise. We show that each constraint in $\mathcal{C}$ is an independence constraint relative to these p_{ib}'s. Each constraint in $\mathcal{C}$ is of the form $\Pr(Q^b_{(v_i,v_j)}) = 0$, for some edge $(v_i, v_j) \in E$, some $b \in \{0, 1, 2\}$ and $Q^b_{(v_i,v_j)} = \{X_i = b, X_j = b\}$. The independence constraint $I(Q^b_{(v_i,v_j)})$ relative to these p_{ib}'s is $\Pr(Q^b_{(v_i,v_j)}) = p_{ib} \cdot p_{jb}$. Since γ is a legal coloring, it is impossible that both $\gamma(v_i) = b$ and $\gamma(v_j) = b$. Therefore, either $p_{ib} = 0$ or $p_{jb} = 0$ and their product is necessarily 0, resulting in the desired constraint. ∎

This theorem can be interpreted as showing the NP-hardness of deciding whether a set of constraints is satisfied by an independent distribution. It shows that the problem is hard

for 2-simple constraints over 3-valued random variables. It is also possible to prove, using a reduction to 3-SAT, that this problem is hard for 3-simple constraints over binary-valued random variables. But, unlike the problem of deciding the satisfiability of a set of constraints by an *arbitrary* distribution (Proposition 2..6), the problem is *not* NP-hard for the case of 2-simple constraints over binary-valued random variables. In this case, a numeric variant of the standard algorithm for 2-SAT can be used to solve the problem in polynomial time.

In general, it is not clear that the problem of Theorem 3..3 is even in NP. The set P relative to which a given $\mathcal{C}$ is a set of independence constraints might contain irrational numbers even if all the numbers in the input are rational.

Example 3..4. Consider the problem of constructing a distribution over the binary-valued variables X_1, X_2, and X_3 satisfying

$$\begin{aligned} \Pr(\{X_1 = 1, X_2 = 1\}) &= \tfrac{1}{2} \\ \Pr(\{X_1 = 1, X_3 = 1\}) &= \tfrac{1}{2} \\ \Pr(\{X_2 = 1, X_3 = 1\}) &= \tfrac{1}{2} \ . \end{aligned}$$

These are independence constraints only with respect to $p_{11} = p_{21} = p_{31} = \frac{1}{\sqrt{2}}$. ∎

In most practical cases, however, the p_{ib}'s are part of the specification of the algorithm. Thus, it is usually reasonable to assume that they are known.

4. Derandomizing algorithms

In this section we demonstrate how the technique of Section 3 can be used to derandomize algorithms. We present three progressively improving ways in which the technique can be applied. For the sake of simplicity and for ease of comparison, we will base our analysis on a single problem. This is the problem of finding large independent sets in sparse hypergraphs. The problem description and the randomized algorithm for its solution are taken from Alon, Babai, and Itai [3]. Note that a deterministic polynomial-time algorithm for this problem is known [2].

A *d-uniform hypergraph* is a pair $\mathcal{H} = (V, \mathcal{E})$ where $V = \{v_1, \ldots, v_n\}$ is a set of *vertices* and $\mathcal{E} = \{E_1, \ldots, E_m\}$ is a collection of subsets of V, each of cardinality d, that are called *edges*. For simplicity, we restrict attention to d-uniform hypergraphs; a similar analysis goes through in the general case. A subset $U \subseteq V$ is said to be *independent* if it contains no edge.

Consider the randomized Algorithm 2 (k will be defined later). The following theorem, due to Alon, Babai, and Itai [3], states that this algorithm finds "large" independent sets in hypergraphs with "high" probability. We only sketch the proof of this theorem, concentrating on the part that is relevant to this discussion: the constraints on the distribution assumed by the proof.

Proposition 4..1 (Alon, Babai, Itai) : *If $\mathcal{H} = (V, \mathcal{E})$ is a d-uniform hypergraph with n vertices and m edges, then for $k = (1/18)(n^d/m)^{1/(d-1)}$ Algorithm 2 finds an independent set of cardinality exceeding k with probability greater than $\frac{1}{2} - \frac{3}{k}$.*

Algorithm 2: Independent Sets in Hypergraphs

1. **Construct a random subset R of V.**
   For each vertex $v_i \in V$:
   put v_i in R with probability $p = 3k/n$.
2. **Modify R into an independent set U.**
   For each edge $E_j \in \mathcal{E}$ such that $E_j \subseteq R$:
   remove from R some arbitrary vertex $v_i \in E_j$.

Proof: For each vertex $v_i \in V$, let X_i be the random variable that equals 1 if $v_i \in R$ and 0 otherwise. For each edge $E_j \in \mathcal{E}$, let Y_j be the random variable that equals 1 if $E_j \subseteq R$ and 0 otherwise. The cardinality of R is $|R| = \sum_{i=1}^{n} X_i = X$, so $E(X) = np = 3k$.

- If the X_i's are pairwise independent, then the variance of X is

$$\sigma^2(X) = \sum_{i=1}^{n} \sigma^2(X_i) = np(1-p) < np = 3k \ . \tag{1}$$

  Thus, using Chebychev's inequality,

$$\Pr(X \leq 2k) \leq \frac{\sigma^2(X)}{k^2} < \frac{3}{k} \ .$$

- If the X_i's are d-wise independent then for every $j = 1, \ldots, m$,

$$E(Y_j) = \Pr\left(\bigcap_{i \in E_j} \{X_i = 1\} \right) = p^d \ . \tag{2}$$

  Let $Y = \sum_{j=1}^{m} Y_j$ denote the number of edges contained in R. Computation shows that $\Pr(Y \geq k) < \frac{1}{2}$.

If R contains at least $2k$ vertices after the first stage in the algorithm, and at most k vertices are removed in the second stage, then the independent set constructed by the algorithm has cardinality at least k. This has probability at least

$$\Pr\left(\{Y < k\} \cap \{X \geq 2k\}\right) > \frac{1}{2} - \frac{3}{k} \ ,$$

as desired. ∎

Derandomization I

The derandomization procedure of Alon, Babai, and Itai [3] is based on constructing a joint distribution of d-wise independent variables X_i that approximates the joint d-wise independent distribution for which $\Pr(X_i = 1) = 3k/n$ $(i = 1, \ldots, n)$. It is then necessary to analyze this approximate distribution in order to verify that the correctness proof above continues to

hold. Our technique provides exactly the required distribution, so that no further analysis is needed. As we explained in the introduction, this can be done by considering the set $\mathcal{C}^I$ of the constraints:[10]

$$I(\{X_{i_1} = b_1, \ldots, X_{i_d} = b_d\}) \ : \ i_1, \ldots, i_d \in \{1, \ldots, n\},\ b_1, \ldots, b_d \in \{0,1\}\} \ .$$

The number of these constraints is $|\mathcal{C}^I| = \binom{n}{d} 2^d = O((2n)^d)$. For fixed d, this number is polynomial in n, resulting in a sample space of polynomial size (in fact, the size of the sample space is comparable to the one achieved in [3]). Therefore, the algorithm runs in polynomial time, including both the phase of constructing the sample space and the phase of running Step 2 of Algorithm 2 on each point of this space until a sufficiently large independent set is found.

Derandomization II

A closer examination of the proof reveals that not all the $\binom{n}{d}$ neighborhoods of cardinality d have to be independent. In order for equation (2) to hold, it suffices that only the X_i's associated with vertices in the same edge be independent. If $E_j = \{v_{i_1}, \ldots, v_{i_d}\}$, let $\mathcal{C}_j$ denote the set of 2^d independence constraints

$$\{I(\{X_{i_1} = b_1, \ldots, X_{i_d} = b_d\}) \ : \ b_1, \ldots, b_d \in \{0,1\}\} \ .$$

On the other hand, in order for equation (1) to hold, the X_i's must still be pairwise independent. Let $\mathcal{C}^2$ denote the set of $4\binom{n}{2}$ constraints

$$\{I(\{X_{i_1} = b_1, X_{i_2} = b_2\}) \ : \ i_1, i_2 \in \{1, \ldots, n\},\ b_1, b_2 \in \{0,1\}\} \ .$$

Thus, the following set of constraints suffices:

$$\mathcal{C}^{II} = \mathcal{C}^2 \cup \bigcup_{E_j \in \mathcal{E}} \mathcal{C}_j \ .$$

More precisely, if the set $\mathcal{C}^{II}$ is satisfied then the proof of Proposition 4..1 goes through, and the resulting sample space must contain a point that is good for this hypergraph. Since the number of constraints is

$$|\mathcal{C}^{II}| = |\mathcal{C}^2| + \sum_{E_j \in \mathcal{E}} |\mathcal{C}_j| = 4\binom{n}{2} + m2^d \ ,$$

this results in a polynomial-time algorithm for $d = O(\log n)$. This algorithm therefore applies to a larger class of graphs than the one presented by Alon, Babai, and Itai [3]. At first glance, it seems that as we have polynomially many neighborhoods of logarithmic size, Schulman's technique [17] can also be used in this case. However, his approach is limited to (uniformly distributed) random bits so it does not apply to this algorithm. The results of Berger and Rompel [7] and of Motwani, Naor, and Naor [14], however, provide a polynomial-time algorithm for $d = O(\mathit{polylog}\ n)$. Their results use a completely different technique, and cannot be extended to handle larger values of d.

[10] Theoretically, we also need to include the constraints $\Pr(\Omega) = 1$ and $\Pr(\{X_i = 1\}) = p$ for all i. However, these are implied by the other constraints in $\mathcal{C}^I$. This will also be the case for the later sets of constraints $\mathcal{C}^{II}$ and $\mathcal{C}^{III}$.

Derandomization III

A yet closer examination of the proof of Proposition 4..1 reveals that Equation (2) does not require complete independence of the neighborhood associated with the edge E_j. It suffices to constrain the probability of the event "all the vertices in E_j are in R" (the event corresponding to the random variable Y_j in the proof). That is, for $E_j = \{v_{i_1}, \ldots, v_{i_d}\}$, we need only the independence constraint over the event:

$$Q_j = \{X_{i_1} = 1, \ldots, X_{i_d} = 1\} .$$

This is a simple event that defines an independence constraint of the type to which our technique applies. We conclude that the following set of constraints suffices for the analysis of Proposition 4..1 to go through:

$$\mathcal{C}^{III} = \mathcal{C}^2 \cup \{I(Q_j) : E_j \in \mathcal{E}\} .$$

The number of constraints

$$|\mathcal{C}^{III}| = 4\binom{n}{2} + m$$

is polynomial in n and m regardless of d. Therefore, this results in a deterministic polynomial-time algorithm for finding large independent sets in arbitrary uniform hypergraphs.

5. Conclusions and open questions

We have presented a new approach to constructing distributions with small sample spaces. Our technique constructs a distribution tailored precisely to the required constraints. The construction is based on an explicit representation of the constraints as a set of linear equations over the distribution. It enables us to construct sample spaces for arbitrary distributions over discrete random variables, that are precise (not approximations) and sometimes considerably smaller than sample spaces constructed using previously known techniques. This construction can be done in polynomial time for a large class of practical problems—those problems that can be described using only independence constraints.

A number of open questions arise immediately from our results.

- Schulman's approach constructs a sample space whose size depends not on the total number of neighborhoods involved in constraints, but on the maximum number of such neighborhoods in which a particular variable appears. Perhaps the size of the sample space in our approach can similarly be reduced to depend on the maximum number of independence constraints in which a variable X_i participates.

- We mentioned in the introduction that the nature of our approach generally prevents a precomputation of the manageable distribution. However, our approach shows the existence of manageable distributions that are useful in general contexts. For example, for every n, d, and p, we show the existence of a d-wise independent distribution over n binary random variables such that $\Pr(X_i = 1) = p$ for all i. It would be useful to come up with an explicit construction for this class of distributions.

- Our technique constructs distributions that precisely satisfy a given set of arbitrary independence constraints. It is natural to ask if our results can be improved by only requiring the distribution to approximately satisfy these constraints. In particular, it may be possible to construct approximate distributions faster, or in parallel (see [11]), or over smaller sample spaces. We note that the original d-wise independent constructions [3, 10, 13] precisely satisfy the d-wise independence constraints but approximately satisfy the constraints on the value of $\Pr(X_i = b)$. In contrast, the nearly-independent constructions [4, 5, 9, 15] approximately satisfy the d-wise independence constraints. Thus, these results can all be viewed as providing an answer to this question for certain types of constraint-sets $\mathcal{C}$ and certain restrictions on which constraints can be approximated.

- Combined with our inability to precompute the distribution, the sequential nature of our construction prevents its use for derandomization of parallel algorithms. Parallelizing the construction could open up many application areas for this approach (see [11]).

Acknowledgements

The authors wish to thank Joe Kilian for suggesting a considerably simplified proof for Proposition 2..6. We would also like to thank Yossi Azar and David Karger for stimulating discussions, and Howard Karloff, Moni Naor, and Sundar Vishwanathan for useful comments on previous versions of this paper.

References

[1] L. Adleman. Two theorems on random polynomial time. In *Proceedings of the 19th Annual Symposium on Foundations of Computer Science*, pages 75–83, 1978.

[2] N. Alon. Private communication.

[3] N. Alon, L. Babai, and A. Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *Journal of Algorithms*, 7:567–583, 1986.

[4] N. Alon, O. Goldreich, J. Hastad, and R. Peralta. Simple constructions of almost k-wise independent random variables. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, pages 544–553, 1990.

[5] Y. Azar, R. Motwani, and J. Naor. Approximating arbitrary probability distributions using small sample spaces. Unpublished Manuscript.

[6] P. A. Beling and N. Megiddo. Using fast matrix multiplication to find basic solutions. Technical Report RJ 9234, IBM Research Division, 1993.

[7] B. Berger and J. Rompel. Simulating $(\log^c n)$-wise independence in NC. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 2–7, 1989.

[8] B. Chor, O. Goldreich, J. Hastad, J. Friedman, S. Rudich, and R. Smolensky. t-resilient functions. In *Proceedings of the 26th Annual Symposium on Foundations of Computer Science*, pages 396–407, 1985.

[9] G. Even, O. Goldreich, M. Luby, N. Nisan, and B. Veličković. Approximations of general independent distributions. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 10–16, 1992.

[10] A. Joffe. On a set of almost deterministic k-independent random variables. *Annals of Probability*, 2:161–162, 1974.

[11] D. R. Karger and D. Koller. A (de)randomized derandomization technique. Unpublished Manuscript.

[12] N. Linial, M. Luby, M. Saks, and D. Zuckerman. Efficient construction of a small hitting set for combinatorial rectangles in high dimension. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing, to appear*, 1993.

[13] M. Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM Journal on Computing*, 15(4):1036–1053, 1986.

[14] R. Motwani, J. Naor, and M. Naor. The probabilistic method yields deterministic parallel algorithms. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 8–13, 1989.

[15] J. Naor and M. Naor. Small-bias probability spaces: Efficient constructions and applications. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, pages 213–223, 1990.

[16] P. Raghavan. Probabilistic construction of deterministic algorithms: Approximating packing integer problems. *Journal of Computer and System Sciences*, 37:130–143, 1988.

[17] L. J. Schulman. Sample spaces uniform on neighborhoods. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 17–25, 1992.

[18] J. Spencer. *Ten Lectures on the Probabilistic Method.* Society for Industrial and Applied Mathematics, 1987.