Constructing Small Sample Spaces for De-Randomization of Algorithms

Daphne Koller* Nimrod Megiddo†

September 1993

The subject of this paper is finding small sample spaces for joint distributions of n Bernoulli random variables where the probabilities of some events are prescribed. The problem of recognizing whether the prescribed probabilities are consistent is NP-hard. It is shown, however, that if the probabilities are consistent, then there exists a sample space that supports them whose cardinality does not exceed the number of events with prescribed probabilities. It is also shown that if the probabilities are consistent with a joint distribution of n independent Bernoulli variables, then a small sample space can be constructed in polynomial time. This last result is useful for converting randomized algorithms into deterministic ones. We demonstrate this technique by an application to the problem of finding large independent sets in sparse hypergraphs.

1. Introduction

The probabilistic method of proving existence of combinatorial objects has been very successful (see, for example, [?; ?]). The underlying idea is as follows. Suppose we wish to prove the existence of at least one element of a certain type within a finite set Ω. In other words, the elements of Ω are classified as "good" and "bad" and we wish to prove the existence of at least one good element. The proof then goes by constructing a probability distribution f over Ω and showing that the probability of picking a good element is positive. Probabilistic proofs often yield probabilistic algorithms for constructing a good element. In particular, many randomized algorithms are special cases of this technique, where the "good" elements are those sequences of random bits leading to a correct answer.

*Department of Computer Science, Stanford University, Stanford, California 94305, and IBM Almaden Research Center, 650 Harry Road, San Jose, California 95120-6099. Research supported in part by ONR Contract N00014-91-C-0026.

†IBM Almaden Research Center, 650 Harry Road, San Jose, California 95120-6099, and School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel. Research supported in part by ONR Contract N00014-91-C-0026.

For various reasons, it is often desirable to find a deterministic construction to replace the probabilistic one, or to convert a randomized algorithm into a deterministic algorithm. Obviously, this can be done by complete enumeration of the sample space Ω. Unfortunately, the cardinality of the sample space is typically exponential, *e.g.*, the obvious sample space of n independent Bernoulli random variables contains 2^n points. Thus, we are interested in constructing a smaller sample space which supports the distribution that is actually required by the algorithm.

Adleman [**?**] shows that for any randomized polynomial-time algorithm, there exists a sample space 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 for de-randomizing algorithms. In some cases, however, sample spaces of polynomial size are known. In particular, for a joint distribution f of n d-wise independent random variables, Alon, Babai, and Itai [**?**] construct a sample space of cardinality $O(n^d)$ with a distribution f' that approximates f (see also [**?**]). They also show that the cardinality of a sample space of a joint distribution of n d-wise independent Bernoulli variables must be at least $\Omega(n^{\lfloor d/2 \rfloor})$; this is also shown in [**?**]. Therefore, their construction is close to optimal for this case of n d-wise independent variables.

In order to improve on their results, we take a somewhat different approach. Instead of placing an upper bound on the degree of independence required by the algorithm, we examine what are the characteristics that are required in order for the algorithm to work. We then construct a distribution satisfying these requirements. In many cases, this approach yields a sample space smaller than the one obtained by placing an upper bound on the degree of independence. The distribution produced by our technique is, however, not necessarily uniform. Therefore, although our sample space is small, the number of uniform random variables required to generate is not necessarily small.

For the sake of simplicity, we restrict ourselves to joint distributions of n Bernoulli random variables $X_1, \ldots, X_n$ defined on $\Omega = \{0,1\}^n$. Our constructions can easily be extended to variables with larger ranges. 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 *essential sample space* of f.

The randomness requirements of an algorithm may be described in terms of constraints as follows. A *constraint* is an eqaulity of the form

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

where $E \subseteq \Omega$ is an *event* and $0 \leq \pi \leq 1$. The constraint $\Pr(\Omega) = 1$ always has to be satisfied. For example, the joint distribution of n independent uniform Bernoulli variables satisfies all the constraints of the form $\Pr(\{\boldsymbol{x}\}) = 1/2^n$ $(\boldsymbol{x} \in \Omega)$.

We say that a constraint $\Pr(E) = \pi$ is *k-simple* if there exist $\{i_1, \ldots, i_k\} \subseteq N =$

$\{1,\ldots,n\}$ and $\{b_1,\ldots,b_k\} \subseteq \{0,1\}$ such that

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

We say that a constraint is *simple* if it is k-simple for some k. Note that this 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. We will always assume that simple constraints are represented in linear space. Denote

$$\begin{aligned} \alpha(p;0) &= 1-p \\ \alpha(p;1) &= p \\ p_i &= \Pr(X_i = 1) \quad (i = 1,\ldots,n)\ . \end{aligned}$$

We say that a simple constraint

$$\Pr(\{X_{i_1} = b_1\ ,\ldots,\ X_{i_k} = b_k\}) = \pi$$

is an *independence constraint* if

$$\pi = \prod_{j=1}^{k} \alpha(p_{i_j}; b_j)\ .$$

For a fixed set of p_i's, we call this constraint the independence constraint corresponding to the event $E = \{X_{i_1} = b_1\ ,\ldots,\ X_{i_k} = b_k\}$, and denote it by $I(E)$. Obviously, if $X_1,\ldots,X_n$ are independent Bernoulli variables, then their joint distribution satisfies all the independence constraints. Moreover, 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 $i_1,\ldots,i_d \in N$ are distinct indices and $b_1,\ldots,b_d \in \{0,1\}$. In other words, every event defined in terms of only d variables has the same probability as if the variables were independent.

In Section 2 we show that for any set $\mathcal{C}$ of constraints, if $\mathcal{C}$ is consistent, *i.e.*, $\mathcal{C}$ is satisfiable by some distribution f, then there exists a distribution f' also satisfying $\mathcal{C}$ such that $|S(f')| \leq |\mathcal{C}|$. Thus, the cardinality of the essential sample space is not more than the number of constraints. The proof of this theorem includes a technique for constructing a small sample space, but this technique is not useful for de-randomizing algorithms since it requires exponential time.

In Section 3 we show that for a set $\mathcal{C}$ of 2-simple constraints, the problem of recognizing whether there exists a distribution f satisfying $\mathcal{C}$ is NP-complete.

The special case of independence constraints is important and interesting. Randomized algorithms and probabilistic proofs are usually first shown to work when the underlying random variables are independent. If the degree of independence can be bounded,

then the cardinality of the sample space can be reduced. Most of the constraints which occur in de-randomization of algorithms are independence constraints. Let $\mathcal{C}$ be a set of independence constraints defined using a set of p_i's as above. In Section 4 we show that for this $\mathcal{C}$, a distribution with a sample space of cardinality $|\mathcal{C}|$ can be constructed in polynomial time. In particular, for any fixed d, sample spaces of cardinality $O((2n)^d)$ can be constructed in polynomial time for any set of n d-wise independent Bernoulli variable. To our knowledge, this is the first technique which allows the construction of exact distributions of d-wise independent variable with arbitrary p_i's. Previously, except for certain special cases, only approximate distributions were constructed (see [?]).

Our technique allows us to create a sample space with the minimum degree of independence required. In Section 5 we show how the technique can be applied to de-randomization of algorithms. We discuss the problem of finding a large independent set in a d-uniform hypergraph. The underlying randomized algorithm, described in [?], was de-randomized in the same paper for a fixed values of d. It was later de-randomized also for $d = O((\log n)^k)$ in [?] and [?], using different techniques. We show how this algorithm can be de-randomized for any d. We note, however, that the algorithms in [?], [?], and [?] are NC algorithms, whereas our technique does not seem to be efficiently parallelizable. Moreover, a sequential polynomial time solution for the independent set problem in hypergraphs is known[1] We present our derandomization process on the randomized algorithm for this problem just to demonstrate the power of our technique.

2. Existence of small essential sample spaces

Let $\mathcal{C}$ be a set of constraints which includes the obvious one $\Pr(\Omega) = 1$.

Definition 2.1. A distribution f that satisfies $\mathcal{C}$ is said to be *manageable* if $|S(f)| \leq |\mathcal{C}|$.

Theorem 2.2. *If $\mathcal{C}$ is consistent, then $\mathcal{C}$ is satisfied by a manageable distribution.*

Proof: Suppose $|\mathcal{C}| = c$ and $\mathcal{C}$ includes the obvious constraint $\Pr(\Omega) = 1$. Let $\boldsymbol{\pi} \in R^c$ denote a vector containing the values π of the constraints in $\mathcal{C}$. Let $m = 2^n$, $v_{\boldsymbol{x}} = f(\boldsymbol{x})$, and $\boldsymbol{v} = (v_{\boldsymbol{x}})_{\boldsymbol{x}\in\Omega} \in R^m$. The set $\mathcal{C}$ can be represented by a system $\boldsymbol{A}\boldsymbol{v} = \boldsymbol{\pi}$ of linear equations, where $\boldsymbol{A} \in R^{c\times m}$. Since $\mathcal{C}$ is assumed to be consistent, we know that this system has a nonnegative solution. A classical theorem in linear programming asserts under these conditions, there exists a vector $\boldsymbol{v'} \geq \boldsymbol{0}$ such that $\boldsymbol{A}\boldsymbol{v'} = \boldsymbol{b}$ and the columns of $\boldsymbol{A}$ with indices j such that $v'_j > 0$ are linearly independent. Let f' be this solution vector $\boldsymbol{v'}$. Since the number of constraints is $|\mathcal{C}|$, it follows that the number of positive

[1] Private communication with Noga Alon.

indices in f', which is precisely the cardinality of the essential sample space of f', is at most $|\mathcal{C}|$. ∎

The theorem quoted in this proof can be proven constructively. The underlying procedure is as follows.

Algorithm 2.3.

If the set of columns of $\boldsymbol{A}$ with indices j such that $v_j > 0$ are linearly independent, stop; otherwise, find a vector $\boldsymbol{u} \in R^m$, such that $u_j = 0$ for every j such that $v_j = 0$, and $\boldsymbol{Au} = \boldsymbol{0}$.

2. Find a real number t such that $\boldsymbol{v} + t\boldsymbol{u} \geq \boldsymbol{0}$, and for at least one index j such that $v_j > 0$, $v_j + tu_j = 0$.

3. Replace $\boldsymbol{v} \leftarrow \boldsymbol{v} + t\boldsymbol{u}$, and go to 1.

Thus, dependent columns can be removed from a solution one at a time. This suggests a technique for computing a manageable distribution f' from any given distribution f. Alternately, the manageable distribution can be computed directly from the constraints using a linear programming algorithm which computes basic solutions. Unfortunately, since Algorithm 2.3 handles the variables one at a time, it runs in time which is exponential in n, and often also in $|\mathcal{C}|$. Similarly, a linear programming algorithm runs in time polynomial in $m = 2^n$. Our goal is to find a manageable distribution directly from the constraints in polynomial time.

3. Arbitrary sets of constraints

It is often desirable to know, for an arbitrary given set of constraints $\mathcal{C}$, whether or not there exists a distribution f satisfying these constraints. For arbitrary constraints, the representation of the event can be very long, causing the input size to be unreasonably large. We, therefore, restrict attention to simple constraints whose events can be represented in space polynomial in n. It turns out that this problem is NP-hard even when restricted to 2-simple constraints:

Proposition 3.1. *It is* NP*-hard to recognize whether a set $\mathcal{C}$ of 2-simple constraints is consistent.*

Proof: We present only the outline of the proof. For the details, see [**?**]. We begin by viewing the problem as a linear programming problem with an exponential number of variables ($m = 2^n$). The dual of this problem has a variable for each constraint in $\mathcal{C}$ and an exponential number of constraints. It falls into the framework of Grötschel, Lovász, and Schrijver [**?**]. We reduce the problem of minimizing a general quadratic

form in $\{0,1\}$-variables to the separation oracle of the dual problem, showing that it is NP-hard. We then use the framework of [?] to deduce the NP-hardness of the original problem. ∎

An event E is said to be *polynomially checkable* if membership of any $\boldsymbol{x} \in \Omega$ in E can be checked in time polynomial in n.

Lemma 3.2. *If all the constraints in $\mathcal{C}$ pertain to polynomially checkable events, then the consistency of $\mathcal{C}$ can be recognized in non-deterministic time polynomial in terms of $|\mathcal{C}|$ and n.*

Proof: The algorithm guesses a subset $I \subset \Omega$ of cardinality $|\mathcal{C}|$. It then solves in polynomial time the system of equations consisting of the constraints in $\mathcal{C}$, restricted to the variables in I (the other variables are set to 0). Note that, given the initial guess, this system can be constructed in polynomial time, since for each constraint and each variable in I, it takes polynomial time to check whether the variable appears in the constraint. A nonnegative solution to this system exists if and only if there exists a manageable distribution whose essential sample space is I. By Theorem 2.2, we know that a set of constraints is consistent if and only if it is satisfiable by a manageable distribution. 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 deduce the following theorem.

Theorem 3.3. *For an arbitrary set $\mathcal{C}$ of simple constraints, the problem of recognizing the consistency of $\mathcal{C}$ is* NP*-complete.*

4. Quasi-independent variables

An important special case was already discussed in the introduction. Suppose all the constraints in $\mathcal{C}$ are independence constraints arising from a known set of values $p_i = \Pr(X_i = 1)$ $(i = 1, \ldots, n)$. In this case we can construct a small sample space in polynomial time.

The assumption that the p_i's are known is important in view of the following theorem:

Theorem 4.1. *It is* NP*-hard to recognize whether for a given set of simple constraints $\mathcal{C}$ there exist $p_1, \ldots, p_n \in [0,1]$ such that all the members of $\mathcal{C}$ are independence constraints relative to the p_i's.*

Proof: We prove the theorem by reduction from the validity problem of boolean formulas in disjunctive normal form (DNF). Given a DNF formula φ in the variables $u_1, \ldots, u_n$, we build a set $\mathcal{C}$ as follows. Let $\psi = y_1 \wedge \cdots \wedge y_k$ be a disjunct in φ, where $y_j \in \{u_{i_j}, \bar{u}_{i_j}\}$ $(j = 1, \ldots, k)$, and $i_1 < \cdots < i_k$. We associate with ψ a constraint

$$(\{X_{i_1} = b_1 \ , \ldots, \ X_{i_k} = b_k\}, 0) \ ,$$

where $b_j = 1$ if $y_j = u_{i_j}$ and $b_j = 0$ if $y_j = \bar{u}_{i_j}$. It is clear that for any assignment θ of truth-values to the variables $u_1, \ldots, u_n$ (1 for *true* and 0 for *false*)

$$\theta(\psi) = \prod_{j=1}^{k} \alpha(\theta(u_{i_j}); b_j) \ , \tag{1}$$

where $\theta(\psi)$ is the truth-value of ψ under θ.
We claim that φ is valid if and only if $\mathcal{C}$ is not a set of independence constraints for some set of p_i's. For the proof of the 'if' direction, suppose that φ is not valid, and consider an assignment θ of truth-values under which φ is false. Define $p_i = \theta(u_i)$. Since φ is false, every disjunct ψ in φ is false. Due to (1), this implies that every constraint in $\mathcal{C}$ is satisfied relative to these p_i's. For the proof of the 'only if' direction, suppose that $\mathcal{C}$ is a set of independence constraints relative to some set of p_i's. Consider the assignment of *true* to u_i for i such that $p_i > 0$ and, *false* otherwise. Let ψ be any disjunct in φ. Since the corresponding constraint is satisfied, the product is zero, so there must be some index i_j appearing in the constraint for which $\alpha(p_{i_j}; b_j) = 0$. Thus, by (1), ψ is falsified under this assignment θ. Since this holds for every ψ, it follows that φ is falsified under θ, and is therefore not valid. ∎

It is not clear that the problem of Theorem 4.1 is in NP. The set of p_i's, relative to which a given set of constraints is a set of independence constraints, might contain irrational numbers even if all the input numbers are rational, as we show in the following example:

Example 4.2. Consider the problem of constructing a sample space for Bernoulli variables X_1, X_2, and X_3 so that

$$\Pr(\{X_1 = X_2 = 1\}) = \Pr(\{X_1 = X_3 = 1\}) = \Pr(\{X_2 = X_3 = 1\}) = \tfrac{1}{2} \ .$$

The latter are independence constraints only with respect to

$$p_1 = p_2 = p_3 = \frac{1}{\sqrt{2}} \ .$$

Nevertheless, in many cases, the p_i's are either known or easily computable from the constraints. In the following discussion, let $p_1, \ldots, p_n \in [0, 1]$ be fixed and known, and

let $\mathcal{C}$ be a set of independence constraints with respect to these p_i's. Let $c = |\mathcal{C}|$ be the number of constraints. We can now find a sample space of cardinality c satisfying the constraints in $\mathcal{C}$ in time polynomial in cn.

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

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

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

$$\pi_\ell(E) = \{X_{i_1} = b_1 \ , \ldots , \ X_{i_k} = b_r\} \ .$$

Analogously, we call $I(\pi_\ell(E))$ the ℓ-projection of the constraint $I(E)$. We now define, recursively, a sequence of distributions $f_1, \ldots, f_n$, such that for each ℓ $(\ell = 1, \ldots, n)$, the following conditions hold:

(i) f_ℓ is a distribution on $\{0,1\}^\ell$.
(ii) f_ℓ satisfies the ℓ-projections of the constraints in $\mathcal{C}$.
(iii) The cardinality of the essential sample space, $S(f_\ell)$, of f_ℓ is at most c.

The distribution f_n will be the desired distribution. We begin by defining

$$\begin{aligned} f_1((0)) &= 1 - p_1 \\ f_1((1)) &= p_1 \ . \end{aligned}$$

This clearly satisfies all the requirements. Now, assume that f_ℓ satisfies the above requirements, and define an intermediate distribution $g_{\ell+1}$ by

$$g_{\ell+1}(x_1, \ldots, x_\ell, b) = f_\ell(x_1, \ldots, x_\ell) \cdot \alpha(p_{\ell+1}; b) \quad (b \in \{0,1\}) \ . \tag{2}$$

Lemma 4.3. *If f_ℓ satisfies the ℓ-projections of the constraints in $\mathcal{C}$, then $g_{\ell+1}$ satisfies the $(\ell+1)$-projections of the constraints in $\mathcal{C}$.*

Proof: Suppose $I(E)$ is an arbitrary constraint in $\mathcal{C}$, where $E = \{X_{i_1} = b_1 \ , \ldots , \ X_{i_k} = b_k\}$. For simplicity, denote $E_j = \pi_j(E)$ $(j = 1, \ldots, n)$. Let r be the maximal index such that $i_r \leq \ell$. By the assumption,

$$f_\ell(E_\ell) = \prod_{j=1}^{r} \alpha(p_{i_j}; b_j) \ .$$

We distinguish two cases:
Case I: $i_{r+1} = \ell + 1$. In this case

$$E_{\ell+1} = \{(x_1, \ldots, x_{\ell+1}) | (x_1, \ldots, x_\ell) \in E_\ell, \ x_{\ell+1} = b_{r+1}\} \ ,$$

and therefore

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

Thus, $g_{\ell+1}$ satisfies the independence constraint $I(E_{\ell+1})$.
Case II: $i_{r+1} \neq \ell+1$. In this case

$$E_{\ell+1} = \{(x_1,\ldots,x_{\ell+1}) | (x_1,\ldots,x_\ell) \in E_\ell, x_{\ell+1} \in \{0,1\}\} \ ,$$

and thus

$$\begin{aligned}
g_{\ell+1}(E_{\ell+1}) &= \sum_{(x_1,\ldots,x_{\ell+1})\in E_{\ell+1}} g_{\ell+1}(x_1,\ldots,x_{\ell+1}) \\
&= \sum_{(x_1,\ldots,x_\ell)\in E_\ell, x_{\ell+1}\in\{0,1\}} g_{\ell+1}(x_1,\ldots,x_{\ell+1}) \\
&= \sum_{(x_1,\ldots,x_\ell)\in E_\ell, x_{\ell+1}=0} g_{\ell+1}(x_1,\ldots,x_{\ell+1}) + \sum_{(x_1,\ldots,x_\ell)\in E_\ell, x_{\ell+1}=1} g_{\ell+1}(x_1,\ldots,x_{\ell+1}) \\
&= \alpha(p_{\ell+1};0) \prod_{j=1}^{r} \alpha(p_{i_j};b_j) + \alpha(p_{\ell+1};1) \prod_{j=1}^{r} \alpha(p_{i_j};b_j) \\
&= \prod_{j=1}^{r} \alpha(p_{i_j};b_j) \ .
\end{aligned}$$

Thus, $g_{\ell+1}$ satisfies the constraint $I(E_{\ell+1})$. ∎

If $|S(f_\ell)| \leq c$, then $|S(g_{\ell+1})| \leq 2c$, since each point with positive probability in $S(f_\ell)$ yields at most two points with positive probabilities in $S(g_{\ell+1})$. Therefore, $g_{\ell+1}$ does not satisfy requirement (iii). But $g_{\ell+1}$ is a nonnegative solution to the system of linear equations defined by the $(\ell+1)$-projections of the constraints in $\mathcal{C}$. Therefore, we may use Algorithm 2.3 to reduce the cardinality of the sample space to c. Let $f_{\ell+1}$ be the resulting distribution. It clearly satisfies all three requirements.

This procedure takes $n-1$ iterations. Each iteration requires at most $O(m^3)$ arithmetic operations for running Algorithm 2.3 to reduce the cardinality of the sample space. Therefore, the entire algorithm runs in $O(nm^3)$ operations. Note that the number of operations does not depend on the magnitudes of the numbers in the input.

Our algorithm can easily be extended to operate on variables with more than two possible values. Let r_i be the number of values in the range of X_i. The sample space of $g_{\ell+1}$ will consist of vectors $(x_1, \ldots, x_\ell, \xi)$ where $(x_1, \ldots, x_\ell) \in S(f_\ell)$ and ξ is in the range of X_i. Then,

$$|S(g_{\ell+1})| \leq r_{\ell+1}|\mathcal{C}| \ .$$

The proof goes through as above but the number of operations in one iteration is $O(r_i m^3)$. The total number of operations is $O((\sum_{i=1}^{n} r_i)m^3) = O(rnm^3)$, where $r = \max\{r_1, \ldots, r_n\}$. The cardinality of the resulting sample space is still $|\mathcal{C}|$.

5. De-randomizing algorithms

In this section we demonstrate how the technique of Section 4 can be used to de-randomize 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 [**?**]. Noga Alon pointed out to us that a sequential polynomial time algorithm for this problem was known.

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, which are called *edges*. A subset $U \subseteq V$ is said to be *independent* if it contains no edges. In this section we choose to restrict ourselves to d-uniform hypergraphs. This is only for the sake of simplicity; a similar process goes through for non-uniform hypergraphs.

Proposition 5.1 (Alon, Babai, and Itai) : *If $\mathcal{H} = (V, \mathcal{E})$ is a d-uniform hypergraph with n vertices and m edges, then for $k = (1/18)(n^d/m)^{1/(d-1)}$, there exists a randomized algorithm that finds an independent set of cardinality exceeding k with probability greater than $\frac{1}{2} - \frac{3}{k}$.*

Proof: The algorithm follows.

Algorithm 5.2.

Construct a random subset R of V so that for each $i \in V$, the probability that $i \in R$ is $p = 3k/n$.

(ii) Construct from R an independent set U as follows. For each edge E_j such that $E_j \subseteq R$, remove from R one arbitrary vertex $i \in E_j$ from R.

Denote by X_i the random variable that equals 1 if $i \in R$, and 0 otherwise. The cardinality of R is $X = |R| = \sum_{i=1}^{n} X_i$, so $E(X) = np = 3k$. If the X_i's are pairwise independent, then the variance of X is

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

By Chebychev's inequality,

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

For each edge $E_j \in \mathcal{E}$, denote by Y_j the random variable that equals 1 if $E_j \subseteq R$ and 0 otherwise. If the X_i's are d-wise independent, then for every j $(j = 1, \ldots, m)$,

$$E(Y_j) = \Pr\left(\bigcap_{i \in E_j} \{X_i = 1\}\right) = p^d = \left(\frac{3k}{n}\right)^d = (3c)^d (n/m)^{d/(d-1)} \ . \tag{4}$$

Let $Y = \sum_{j=1}^{m} Y_j$ denote the number of edges contained in R. It follows that

$$E(Y) = mp^d = (3c)^d (n^d/m)^{1/(d-1)} = 3^d c^{(d-1)} k \ ,$$

so, for $c < 1/18$, $E(Y) < k/2$, and $\Pr(Y \geq k) < \frac{1}{2}$. Consequently,

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

When the event in the latter equality occurs, the independent set constructed by the algorithm has cardinality of at least k. ∎

The de-randomization procedure of Alon, Babai, and Itai [**?**] is based on constructing a joint distribution of d-wise independent variables which approximates the joint d-wise independent distribution of variables X_i for which $\Pr(X_i) = 3k/n$ $(i = 1, \ldots, n)$. It is then necessary to analyze this approximate distribution. 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 $\mathcal{C}$ of the constraints:

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

The number of these constraints is $|\mathcal{C}| = \binom{n}{d} 2^d = O((2n)^d)$. For any fixed d, this number is polynomial in d, resulting in a sample space of polynomial size. Therefore, the algorithm runs in polynomial time, including both the phase of constructing the sample

space and the phase running Step 2 of the algorithm to find an independent set at each point in the resulting space.

A closer examination of the proof reveals that not all the $\binom{n}{d}$ subsets of cardinality d have to be independent. In order for equation (4) to hold, it suffices that only the random choices for vertices in the same edge be independent. If $E_j = \{i_1, \ldots, i_d\}$, let $\mathcal{C}_j$ denote the set of 2^d independence constraints

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

On the other hand, in order for equation (3) to hold, the choices must still be pairwise independent. Denote by $\mathcal{C}^2$ the set of $4\binom{n}{2}$ constraints

$$I(\{X_{i_1} = b_1, X_{i_2} = b_2\}) \quad (i_1, i_2 \in N,\ b_1, b_2 \in \{0,1\} \ .$$

Thus, the following set of constraints suffices:

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

More precisely, if the set $\mathcal{C}'$ is satisfied, then the proof of Proposition 5.1 goes through, and the resulting sample space must contain a point which is good for this hypergraph. Since the number of constraints is

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

this results in a polynomial time algorithm for $d = O(\log n)$, which applies to a larger class of graphs than the one presented in [?]. We note that more recent results of Berger and Rompel [?] and of Motwani, Naor, and Naor [?] provide polynomial time algorithms for $d = O((\log n)^k)$ for any fixed k. These results use completely different techniques, and cannot be extended to handle larger values of d.

A yet closer examination of the proof of Proposition 5.1 reveals that equation 4 does not require full independence of the random variables associated with the set of vertices of the edge. It suffices that the probability of the event

$$Y_j = \{X_{i_1} = 1, \ldots, X_{i_d} = 1\} \ ,$$

defined on these variables, be the same as if they were independent. This is a simple event which defines an independence constraint of the type our technique applies to. Finally, the following set of constraints suffices and the analysis of Proposition 5.1 goes through:

$$\mathcal{C}^* = \mathcal{C}^2 \cup \bigcup_{E_j \in \mathcal{E}} I(Y_j) \ .$$

The number of constraints

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

is polynomial in the size of the problem (nm) regardless of d. Therefore, this results in a deterministic polynomial time algorithm for finding independent sets of cardinality greater than k for arbitrary uniform hypergraphs.