Derandomization Via Matroidal Sample Spaces

K. Gopalakrishnan
Department of Computer Science
East Carolina University
Greenville, NC 27858

May 17, 2004

Abstract

The amount of random bits used in a probabilistic algorithm can be considered as a complexity measure just like time and space. Hence, in recent years, many researchers have developed techniques to reduce the number of random bits needed in computation and to eliminate completely, if possible, the use of random bits. In this paper, we propose a new technique in its abstract form for derandomization using matroidal sample spaces. However, we are not able to identify any concrete algorithm where these techniques might be useful.

keywords: Derandomization, Probabilistic Algorithms, Binary Matroids.

1 An overview of previous work

Any probabilistic algorithm typically uses random bits in the course of its computation. However, it is difficult, if not impossible, to generate random bits. Hence one can view random bits as a scarce resource (such as time and space), to be used as sparingly as possible. Hence many researchers have developed techniques to reduce the number of random bits needed in computation and to eliminate completely, if possible, the use of random bits. In this section we present a brief overview of derandomization techniques using which randomized algorithms can be converted into deterministic algorithms. In section 2, a concrete example of derandomization using k-wise independence method is presented for illustration. In section 3 motivation for our new technique and an introduction to matroids is provided. In section 4, we present our derandomization technique which involves matroidal sample spaces. Finally in section 5, our technique is compared with the previous ones and some concluding remarks are made.

There are basically four major techniques for derandomization namely the method of conditional probabilities, k-wise independence, $\log n$-wise independence and almost k-wise independence. We present a very brief overview of all these techniques with special emphasis on the method based on k-wise independence. For an excellent survey of the first two methods the reader is referred to the recent book by Alon and Spencer [3].

1.1 The method of conditional probabilities

The method of conditional probabilities is due to J. Spencer [14] and P. Raghavan [12], although it was implicitly known before. Suppose we have a probability space containing 2^l points, denoted by binary vectors of length l. Let $A_1, A_2, \ldots A_s$ be a collection of events and suppose that $\sum_{i=1}^{s} Pr(A_i) = k$. Thus k is the expected value of the number of events A_i that hold, and hence there is a point $(\epsilon_1, \ldots, \epsilon_l)$ in the space in which at most k events hold. The objective is to find such a point. There is a trivial randomized algorithm to do so, namely randomly generate a vector of length l. Using the method of conditional probabilities it is sometimes possible to derandomize the abovementioned randomized algorithm. It is easy to verify that if the values of ϵ_j are chosen, each one in its turn, so as to minimize the value of $\sum_{i=1}^{s} Pr(A_i|\epsilon_1, \ldots, \epsilon_j)$, then the value of this sum cannot increase. Since this sum is k at the beginning, it follows that it is at most k at the end. But at the end, each ϵ_j is fixed, and hence the value of this sum is precisely the number of events A_i that hold at the point $(\epsilon_1, \ldots, \epsilon_l)$ and hence the objective is attained in a deterministic way.

Note that the basic idea is akin to conducting a binary search of the sample space for a "good" point. The method of conditional probabilities works fine provided the conditional probabilities $Pr(A_i|\epsilon_1, \ldots, \epsilon_j)$ can be computed efficiently. However, there are many interesting instances where this is not the case. A trick that can be useful in such cases is the concept of *pessimistic estimators*. We will not delve into the details of *pessimistic estimators* except to say that they provide an upper bound for the conditional probabilities and can be efficiently computed.

1.2 k-wise independence method

This method is a general technique for converting randomized algorithms whose analysis depends only on k-wise independence (rather than complete independence) of random variables involved, into deterministic ones. This method was first developed by M. Luby for the case $k = 2$ in the context of maximal independent set problem [7] and later generalized to arbitrary constant k by Alon, Babai and Itai [1].

The basic idea is to replace an exponentially large sample space by one of polynomial size. If a random variable on such a space takes a certain value with positive probability, then we can find a point in the sample space in which this happens simply by deterministically checking all the points. This can be done with no loss of time by using a polynomial number of parallel processors.

Small sample spaces with many k-wise independent random variables in them can be constructed from any linear error correcting code with appropriate parameters. The construction described here is based on the binary BCH codes [8].

Theorem 1.1 *Suppose* $n = 2^m - 1$ *and* $k = 2t + 1$. *Then there exists a symmetric probability space* Ω *of size* $2(n + 1)^t$ *and* k*-wise independent random variables* $y_1, y_2, \ldots y_n$ *over* Ω*, each of which takes the values 0 and 1 with equal probability.*

Proof: Consider the $1 + mt$ by n parity check matrix H of the extended binary BCH code of length n and designed distance $2t+2$. It is well known that any $2t + 1$ columns of H are linearly independent over $GF(2)$. Now form a new matrix A whose rows are all the linear combinations of the rows of H. Then A forms an orthogonal array of strength k. The points in the new sample space are the rows of A and the n columns correspond to the n random variables. It is trivial to verify that the random variables are k-wise independent and the new sample space is polynomial in size. Further details of the proof can be found in [1] and [3]. $\square$

While this method is simple and powerful one, it does not extend to the cases where $f(n)$-wise independence of the random variables is needed, where $f(n)$ is an increasing (however small the growth rate may be) function of n, as shown by the following theorem.

Theorem 1.2 *The size of the sample space constructed in Theorem 1.1 is asymptotically optimal and hence the size of a sample space with n k-wise independent non-trivial random variables can be polynomial in n only when k is a constant.*

Proof: The proof is a straightforward application of Rao bounds [13] for orthogonal arrays and can be found in [1] and [6]. $\Box$

1.3 $\log^c n$-wise independence method

When $\log n$ wise independence of the random variables are needed, then the previous method does not work. Hence a new technique was developed independently by Berger and Rompel [4] and Motwani, J. Naor and M. Naor [9]. This technique works even when polylogarithmic independence is required. This can be thought of as a combination of the previous two methods or as a parallelization of the method of conditional probabilities.

The basic idea behind the technique is outlined in the following. Suppose we have a sample space Ω such that $\log |\Omega|$ is polylogarithmic in the input size. Clearly, the method of conditional probabilities, if applicable, would find a good point in Ω in poly-logarithmic time. First it is shown that in certain algorithms it is sufficient to assume that the random choices are k-wise independent only, where k is polylogarithmic in the input size. Then a sample space Ω is constructed, with $\log |\Omega|$ being poly-logarithmic, such that sampling from it gives k-wise independence. This construction is essentially the previous method. However it is important that the construction permits computation of relevant conditional probabilities, or estimators thereof, in NC. Finally, the method of conditional probabilities is applied to this new sample space to obtain a good point in NC.

However this method, as is the method of conditional probabilities, is not completely general and will work only in some cases.

1.4 Almost k-wise independence method

This is a completely different method in competition with the previous method. This technique is due to J. Naor and M. Naor [11]. The equivalence between the following two conditions can be found in [6] and [15].

1. The n random variables $x_1, \ldots, x_n$ are independent and for all i, $Pr(x_i = 0) = Pr(x_i = 1)$.
2. For every subset $S \in \{1, \ldots, n\}$, it is equally likely that the parity of the subset is either zero or one.

The second condition is relaxed and a sample space is constructed such that for every subset $S \in \{1, \ldots, n\}$, it is "almost" equiprobable that the parity of the subset is zero or one. To be more precise, we require that for every subset S,

$$\left| Pr\left(\sum_{i \in S} x_i = 0\right) - Pr\left(\sum_{i \in S} x_i = 1\right) \right| \leq \epsilon.$$

This quantity is called the *bias* of the subset S and the sample spaces constructed are called small-bias probability spaces or alternatively almost independent probability spaces. The logarithm of the cardinality of the sample space constructed is $O(\log \frac{1}{\epsilon} + \log n)$. Hence, if ϵ is typically small, then the size of the sample space is also polynomial.

For almost k-wise independent sample spaces or k-wise independent ϵ-biased sample spaces, only the bias of subsets that are smaller than k is guaranteed to be bounded by ϵ. In this case the logarithm of the cardinality of the sample space constructed is $O(\log k + \log\log n + \log \frac{1}{\epsilon})$.

ϵ-biased sample spaces and almost k-wise independent sample spaces have application in derandomization of some randomized algorithms. Obviously using these techniques we can achieve poly-logarithmic independence, but only if we are content with the small bias. Some improvement in the constructions of this kind of sample spaces can be found in [2].

2 A Concrete Example

In this section, we illustrate derandomization using k-wise independence method by providing a simple concrete example. The problem that we will use is the "Monochromatic K_4's" problem and can be found in [3].

For any integer $n \geq 4$, there is an edge-colouring of K_n with two colours such that the number of monochromatic induced K_4's is at most $\binom{n}{4}/32$. This could be seen easily using a probabilistic argument as follows. For a $2-$edge-colouring c of K_n, let $M(c)$ denote the number of monochromatic induced K_4's. Let $\overline{M}$ denote the average value of $M(c)$ over all colourings c. Now

$$\begin{aligned}
\overline{M}2^{\binom{n}{2}} &= \sum_{c} M(c) \\
&= \sum_{A \subseteq V, |A|=4} |\{c : |c(A)| = 1\}| \\
&= \binom{n}{4} \times 2 \times 2^{\binom{n}{2}-6} \\
&= \binom{n}{4} \times 2^{\binom{n}{2}-5}
\end{aligned}$$

Hence $\overline{M} = \binom{n}{4}/32$. So there must exist a colouring with at most $\overline{M} = \binom{n}{4}/32$ monochromatic induced K_4's. A randomized algorithm to find such a coloring is to simply generate a random coloring and check whether it has the desired property. Note that the total number of possible colorings is $2^{\binom{n}{2}}$. As this is exponential in the size of the input, it is not possible to test every coloring in polynomial time.

However, it is *not* necessary that the color of every edge be chosen independently at random. It suffices to consider a sample space where every subset of six edges has colors chosen independently at random. As we need only $6-$wise independent sample space, we can use the rows of an $\mathrm{OA}_\lambda\left(6, \binom{n}{2}, 2\right)$ as our sample space. Within this smaller sample space, we once again have

$$\begin{aligned}
\overline{M}\lambda 2^6 &= \sum_{c \in \mathbf{S}} M(c) \\
&= \sum_{A \subseteq V, |A|=4} |\{c \in \mathbf{S} : |c(A)| = 1\}| \\
&= \binom{n}{4} \times 2\lambda
\end{aligned}$$

and so $\overline{M} = \binom{n}{4}/32$.

Let m be such that $\binom{n}{2} \leq 2^m - 1$. Then we can construct orthogonal arrays $\mathrm{OA}_{2^{3(m-2)}}\left(6, \binom{n}{2}, 2\right)$ from BCH codes with designed distance 7. The size of the sample space would then be $O(n^6)$ and so it is a $poly(n)$ sized sample space. Now it is possible to deterministically check each coloring in the sample space. This would yield both a $poly(n)$ time sequential algorithm and a $polylog(n)$ time parallel algorithm with $poly(n)$ processors.

Several examples where derandomization is achieved (using different methods) in concrete algorithms can be found in literature (for example, see [10]).

3 Motivation

In this section motivation for the new technique that we propose and an introduction to matroids is provided. The technique that we propose can be thought of as a generalization of the k-wise independence method. In the k-wise independence method the exponentially large sample space is replaced by a polyonomially sized sample space by exploiting the fact that the analysis of some randomized algorithms requires only k-wise independence and not full independence of the n random variables. The basic idea is to provide only the amount of independence really required.

Let $S = \{1, 2, \ldots, n\}$ represent the set of n random variables. Let

$$\begin{aligned} \mathcal{I} &= \{A : A \subseteq S, \text{the independence of events in } A \text{ is required} \\ &\qquad \text{in the analysis of the randomized algorithm}\}. \end{aligned}$$

We could get maximum advantage by constructing a sample space that provides for mutual independence of only those subsets of random variables for which it is really required. It is clear that if $A \in \mathcal{I}$ and $B \subseteq A$, then $B \in \mathcal{I}$. Let

$$\mathcal{B} = \{B : B \in \mathcal{I}, B \subset A \Rightarrow A \notin \mathcal{I}\}.$$

In the k-wise independence method, only the situation where $\mathcal{B}$ consists of all subsets of size k is considered. In our new technique we consider several other situations as well. To be precise our technique addresses the question whenever $(S, \mathcal{I})$ is a binary matroid. Matroids are well-studied combinatorial objects. A good general reference for matroid theory is the book by Welsh [16]. A brief introduction to matroid theory is provided in what follows.

A matroid M is a pair $(S, \mathcal{I})$, where S is a finite set and $\mathcal{I}$ is a collection of subsets (called *independent* sets) of S satisfying the following three axioms.

1. $\phi \in \mathcal{I}$.
2. If $X \in \mathcal{I}$ and $Y \subseteq X$ then $Y \in \mathcal{I}$.
3. If U, V are members of $\mathcal{I}$ with $|U| = |V| + 1$ there exists $x \in U \backslash V$ such that $V \cup \{x\} \in \mathcal{I}$.

A subset of S not belonging to $\mathcal{I}$ is called *dependent*. A *base* of M is a maximal independent subset of S, the collection of bases is denoted by $\mathcal{B}$. A *circuit* of M is a minimal dependent set, the collection of circuits is denoted by $\mathcal{C}$. It is well known that all bases of a matroid have the same cardinality and this cardinality is called the *rank* of the matroid, usually denoted by r.

Let S be a set of cardinality n and let $\mathcal{I}$ be all subsets of S of cardinality $\leq k$. This is a matroid on S, called the *uniform matroid* of rank k and denoted by $U_{k,n}$. Under this terminology, it is clear that the k-wise independence method deals only with uniform matroids. It is so primarily because uniform matroids have a nice structure and are easy to deal with.

A matroid M on S is *representable over the field* F if there exists a vector space V over F and a map $\phi : S \to V$, such that independent subsets of S are mapped onto linearly independent vectors and dependent subsets of S are mapped onto linearly dependent vectors. Such a map ϕ is called a *representation* of M and a matroid M is said to be *representable* if it is representable over some field. A matroid M is called a *binary matroid* if it is representable over the field $GF(2)$. Our technique will deal with binary matroids.

Note that the collection of subsets whose independence is required in the analysis of a randomized algorithm need not even form a matroid in the first place. In that case, we need to embed that collection into a binary matroid before applying our technique.

4 Matroidal Sample Spaces

In this section, we construct the desired optimal sample spaces. For obvious reasons, we call them *matroidal sample spaces*.

It is trivial to observe that if a matroid M of rank r is representable over a field F, then there exists a dependence preserving mapping $\phi : S \rightarrow V$, where V is a vector space of dimension r over F. We now state and prove our main theorem. Actually the construction used in the proof of the theorem is studied before by Brickell and Davenport [5] in a different context (ideal secret sharing schemes).

Theorem 4.1 *Let $M = (S, \mathcal{I})$ be a binary matroid of rank r. Then there exists a symmetric probability space Ω of size 2^r and random variables $y_1, y_2, \ldots, y_n$ over Ω, each of which takes the values 0 and 1 with equal probabilities and the subsets of random variables corresponding to the elements of $\mathcal{I}$ are mutually independent. Moreover the size of the sample space Ω is optimal (not just asymptotically but in absolute sense).*

Proof: $M = (S, \mathcal{I})$ is a binary matroid and hence is representable over $GF(2)$. Let V be the vector space of dimension r over $GF(2)$. Then there exists a dependence preserving mapping $\phi : S \rightarrow V$. We shall construct a matrix A of size 2^r by n over $GF(2)$. The rows of A will be indexed by the elements of V and the columns will be indexed by elements of S. To be precise, let f be a one to one correspondence from $\{1, 2, \ldots, 2^r\}$ to V. The matrix A is defined as follows:

$$A[i, j] = f(i) \circ \phi(j),$$

where the operation $\circ$ is the usual inner product. The points in the new sample space Ω are the rows of A and the n columns correspond to the n random variables. It is clear that $|\Omega| = 2^r$.

Let y_j be an arbitrary random variable. The number of points in the sample space at which y_j assumes a value of 0 is simply the number of vectors $x = (x_1, x_2, \ldots, x_r)$ in $[GF(2)]^r$ such that $x \circ \phi(y_j) = 0$. This is a system of 1 equation in the r unknowns $x_1, \ldots, x_r$ and hence the dimension of the solution space is $r - 1$. Hence the number of points at which y_j assumes a value of 0 is 2^{r-1}. Since there are only 2^r points in the sample space, the number of points at which y_j assumes a value of 1 is also 2^{r-1}. Thus each of the random variables $y_1, y_2, \ldots, y_n$ takes the values 0 and 1 with equal probability.

It is sufficient to prove the mutual independence of random variables corresponding to the bases of the matroid. Let B be a base of the matroid. Then $|B| = r$. Let $B = \{b_1, b_2, \ldots, b_r\}$. Let $(a_1, a_2, \ldots, a_r)$

be an arbitrary vector in $[GF(2)]^r$. Let us count the number of points in the sample space such that the random variable b_1 assumes a value of a_1, ..., b_r assumes a value of a_r. This is same as the number of vectors $x = (x_1, x_2, \ldots, x_r)$ in $[GF(2)]^r$ such that the following system of equations is satisfied.

$$\forall i, 1 \leq i \leq r, x \circ \phi(b_i) = a_i$$

This is a system of r equations in the r unknowns $x_1, \ldots, x_r$. All the equations are linearly independent as the mapping ϕ preserves independence and the random variables $b_1, b_2, \ldots, b_r$ form a base of the matroid. Hence the system has a unique solution. Since the number of sample points satisfying the constraints is independent of the vector $(a_1, a_2, \ldots, a_r)$ chosen, it is clear that the random variables $b_1, b_2, \ldots, b_r$ are mutually independent.

It is trivial to observe that any sample space which provides for mutual independence of the random variables $b_1, b_2, \ldots, b_r$ should have at least 2^r points in it. Since Ω has exactly 2^r points in it, the optimality of the size of Ω follows. $\Box$.

5 Concluding Remarks

In this section we compare the new technique with other methods available and identify the future work needed. Although we have presented the proposed technique, it is only in its abstract form. We are yet to identify concrete algorithms where the new technique might be profitably used. However, existence of such algorithms is not totally ruled out. In the following, we outline a scenario where the new technique might be useful.

Consider the following situation, which is not necessarily hypothetical. Suppose a randomized algorithm requires n random binary variables. But only for some subsets $\mathcal{I}$ of these n variables mutual independence is needed. Suppose that the largest set of random variables whose mutual independence is required for the analysis of the randomized algorithm is $\log n$. Suppose further that the conditional probabilities involved are difficult to compute or estimate. Then obviously the method of conditional probabilities and the $\log^c(n)$-wise independence method cannot be used. If we actually require absolute independence of the random variables concerned and approximate independence is not sufficient, then the almost k-wise independence method cannot be applied either. The only existing method that will be applicable to this situation is the k-wise independence method. To use the k-wise independence method, we have to provide for independence of all subsets of size $\log n$ of the random variables, even though they may not be actually needed for the analysis. In other words, we are embedding the sets of random variables for which independence is required into the uniform matroid $U_{\log n,n}$. Once we do that the size of the sample space constructed will be $O(n^{\frac{\log n}{2}})$ by theorem 1.1 and that size is asymptotically optimal in view of theorem 1.2. But $n^{\frac{\log n}{2}}$ is clearly not polynomial and hence we cannot exhaustively search the new sample space for a good point in polynomial time.

It is conceivable that the sets of random variables for which independence is required can be embedded into a binary matroid of rank $\log n$, as the size of the largest set of random variables for which independence is required is $\log n$. Once we can do this embedding, then we can apply theorem 4.1 and construct a sample space that provides independence of required sets of random variables. The size of the sample space constructed would be $2^{\log n} = n$. Thus the size of the transformed sample space is not only polynomial but in fact linear. It is clearly possible to search this new sample space in linear time for a good point.

The magical difference and the apparent paradoxical contradiction to theorem 1.2 is due to the fact the k-wise independence method provides for independence of many other subsets of random variables whose independence is really not required for the analysis of the randomized algorithm. The method of matroidal sample spaces might also provide independence for some subsets of random variables whose independence is not really required. However, such additional independence is provided without any extra cost, as the size of the sample space constructed in the matroidal sample spaces method is absolutely optimal.

The above statement tacitly assumes that we can successfully embed the independence system into a binary matroid without increasing the size of the maximal independent set. By an independence system, we mean a collection of subsets which satisfies only the first two axioms in the definition of a matroid. To a large extent the success of the proposed technique depends on the answer to the question: how often the independence systems that arise in practice can be embedded into a binary matroid without increasing the rank?

The insight needed to answer the above question can perhaps be gained by studying the characterization of binary matroids developed by Tutte. For the sake of completeness we record the characterization as the following theorem and the details can be found in [16, Chap. 10].

Theorem 5.1 *The following statements about a matroid M are equivalent.*

1. *M is binary.*

2. *For any circuit C and cocircuit C^*, $|C \cap C^*|$ is even.*

3. *The symmetric difference of any collection of distinct circuits of M is the union of disjoint circuits of M.*

4. *If C_1, C_2 are distinct circuits of M, the symmetric difference $C_1 \triangle C_2$ contains a circuit C.*

5. *For any base B and circuit C of M, if $C \backslash B = \{e_1, e_2, \ldots, e_q\}$ and if $C(e_i)$ is the fundamental circuit of e_i in the base B, then*

$$C = C(e_1) \triangle C(e_2) \triangle \ldots \triangle C(e_q).$$

6. *M has no minor isomorphic to $U_{2,4}$, the uniform matroid of rank 2 on 4 elements.*

It would be interesting to determine whether the proposed derandomization technique can be used to show a problem known to be in RNC but not known to be in NC to be in NC (or a problem known to be in RP but not known to be in P to be in P).

The results of theorem 4.1 can be easily extended to random variables assuming values from an arbitrary finite field $GF(p^n)$ with equal probability. However we did not do so in section 4 for the sake of simplicity.

It is interesting to observe that the proposed derandomization technique provides yet another application of matroids to algorithm design. It is well known that matroids form the abstract model for greedy algorithms.

Acknowledgments

The author would like to thank Sunil Shende and Douglas Stinson for several helpful discussions.

References

[1] Noga Alon, Laszlo Babai, and Alon Itai. A Fast and Simple Randomized Parallel Algorithm for the Maximal Independent Set Problem. *Journal of Algorithms*, 7:567–583, 1986.

[2] Noga Alon, Oded Goldreich, Johan Hastad, and Rene Peralta. Simple Constructions of Almost k-wise Independent Random Variables. *Random Structures and Algorithms*, 3(3):289–304, 1992.

[3] Noga Alon and Joel H. Spencer. *The Probabilistic Method*. John Wiley & Sons, Inc., 1992.

[4] Bonnie Berger and John Rompel. Simulating $(\log^c n)$-wise Independence in NC. *30th FOCS*, pages 2–7, 1989.

[5] E.F. Brickell and D.M. Davenport. On the Classification of Ideal Secret Sharing Schemes. *Journal of Cryptology*, 4:123–134, 1991.

[6] Benny Chor, Oded Goldreich, Johan Håstad, Joel Friedman, Steven Rudich, and Roman Smolensky. The Bit Extraction Problem or t-Resilient Functions. *IEEE Symposium on Foundations of Computer Science*, 26:396–407, 1985.

[7] Michael Luby. A Simple Parallel Algorithm for the Maximal Independent Set Problem. *SIAM J. Comput.*, 15(4):1036–1053, November 1986.

[8] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland, 1977.

[9] Rajeev Motwani, Joseph Naor, and Moni Naor. The Probabilistic Method Yields Deterministic Parallel Algorithms. *30th FOCS*, pages 8–13, 1989.

[10] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *19th STOC*, pages 345–354, 1987.

[11] Joseph Naor and Moni Naor. Small-bias Probability Spaces: Efficient Constructions and Applications. *22nd STOC*, pages 213–223, 1990.

[12] Prabhakar Raghavan. Probabilistic Construction of Deterministic Algorithms: Approximating Packing Integer Programs. *Journal of Computer and System Sciences*, 37:130–143, 1988.

[13] C. R. Rao. Factorial experiments derivable from combinatorial arrangements of arrays. *J. Royal Stat. Soc.*, 9:128–139, 1947.

[14] J. Spencer. *Ten Lectures on the Probabilistic Method.* SIAM, 1987.

[15] V. Vazirani. *Randomness, adversaries and computation*. PhD thesis, University of California, Berkeley, 1986.

[16] D. J. A. Welsh. *Matroid Theory*. L.M.S. Monographs. Academic Press, 1976.