Techniques for Probabilistic Analysis and Randomness-Efficient Computation

Aravind Srinivasan
Ph.D Thesis

93-1378
August 1993

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

TECHNIQUES FOR PROBABILISTIC ANALYSIS AND RANDOMNESS-EFFICIENT COMPUTATION

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Aravind Srinivasan

August 1993

TECHNIQUES FOR PROBABILISTIC ANALYSIS AND RANDOMNESS-EFFICIENT COMPUTATION

Aravind Srinivasan, Ph.D.

Cornell University 1993

Randomness is well-recognized as an important computational resource in theoretical computer science. Not only are there classical problems for which the only known "efficient" solutions are randomized, but there are problems for which randomness can be used to circumvent deterministic lower bounds. However, standard tools available for probabilistic analysis such as the Chernoff–Hoeffding bounds are not sufficiently powerful in all situations; second, since there are no known "true" sources of randomness, there is a need to develop general techniques for reducing/removing randomness from randomized algorithms. This thesis addresses these two issues.

Certain types of scheduling and resource allocation problems in a distributed setting can be modeled as edge coloring problems. In Chapter 2, we present a fast and simple randomized algorithm for edge coloring a graph, in the synchronous distributed point–to–point model of computation. Our algorithm computes an edge–coloring of a graph G with n nodes and maximum degree Δ with at most $(1.6 + \epsilon)\Delta + O(\log^{2+\delta} n)$ colors with high probability, for any fixed $\epsilon, \delta > 0$. To analyze

our algorithm, we introduce techniques for proving upper bounds on the tails of random variables, extending the Chernoff–Hoeffding (CH) bounds to some types of dependent random variables. This is joint work with A. Panconesi [91].

In Chapter 3, we show that the CH bounds for the tails of the sums of bounded and independent random variables $X_1, \ldots, X_n$ require only *limited independence* among the X_is. We show that if $X_1, \ldots, X_n$ lie in $[0,1]$ and are k–wise independent, then $Pr(X \geq E[\sum_i X_i](1+\delta))$ can be upper bounded by the CH bound, if $k \geq \mu \cdot min\{\delta, \delta^2\}$. This leads to algorithms that require a reduced amount of randomness for any analysis which uses the CH bounds. This is joint work with J. P. Schmidt and A. Siegel [108].

In Chapter 4, we present an RNC^2 algorithm for the perfect matching problem which uses $O(\log Z)$ random bits where Z is any given upper bound on the number of perfect matchings in the graph, generalizing results of Grigoriev & Karpinski. Underlying our algorithm is a randomness–optimal generalization of the Isolating Lemma of Mulmuley, Vazirani & Vazirani, which also leads to other applications. This is joint work with S. Chari and P. Rohatgi [26].

Biographical Sketch

Aravind was born to Srimati. Kalyani Srinivasan and Shri. P. R. Srinivasan on the 20th of August, 1968, in Madras, India. After studying at M. Ct. M. Chidambaram Chettiar Memorial Preparatory School, Sindhi Model Senior Secondary School, and M. Ct. M. Higher Secondary School, all in Madras, he joined the Indian Institute of Technology, Madras, in July 1985. He graduated from IIT with a Bachelor of Technology degree in Computer Science in August 1989, and joined Cornell in Fall 1989. He received a Master of Science degree in Computer Science from Cornell in January 1993. He spent Spring 1993 at Princeton University as an exchange student. He has accepted a postdoctoral fellowship offered jointly by the Institute for Advanced Study and the Center for Discrete Mathematics and Computer Science (DIMACS).

A small step for a man, and

An even smaller step for mankind.

– Alessandro Panconesi, circa 1991

Acknowledgements

It is with immense love that I offer my heartfelt thanks to my parents- Srimati. Kalyani Srinivasan (Amma) and Shri. P. R. Srinivasan (Appa), and to my Guru, Shri. Annapurnamba Sahita Shri. Amritananda Natha Saraswati. My parents showered me with love and made several sacrifices to ensure that I get the best education possible and over the past four years, Guruji has been a fount of unconditional love and grace during many a period of confusion.

The influence of my Guru at Cornell, my advisor David Shmoys, has been tremendous- in addition to his sound technical insights and inputs, he was ever-supportive. It is difficult to reconcile his young age with his wisdom in guiding a student- when we started with him, he would guide us every week over even small points, while later on, he gave us full freedom to pursue whatever we liked. Many of the problems and approaches considered here became clear due to discussions with him, with his intuition and wide knowledge helping a lot. He has been my model of a complete academic, in the accent he places on teaching, and in his superb teaching, writing, and presentation skills. He has been a great influence with his style of presentation, and indeed has spent so much time in teaching me this skill. He never scrimped with wise advice, and was always generous with his time. I truly feel indebted to David.

I have had fruitful discussions with Éva Tardos- in particular, some important ideas used in Chapter 2 were suggested by her. Dexter Kozen and Sam Toueg were sources of suggestions on problems, encouragement and advice. I thank Dexter, Éva and Sam for serving on my committee. Bob Constable and Juris Hartmanis were always encouraging, and despite being senior professors with many responsibilities, were ever-willing to offer their kind advice; thank you! Thanks also to Paul Pedersen for his constant encouragement and for his infectious enthusiasm for mathematics. His knowledge of mathematics and spontaneity have been a big inspiration.

The work reported in Chapter 2 is joint work with Alessandro Panconesi, that of Chapter 3 is joint work with Jeanette Schmidt and Alan Siegel, and Chapter 4 is joint work with Suresh Chari and Pankaj Rohatgi. My sincere thanks to my co-authors for all the discussions and for generously sharing their ideas with me. I also thank Joseph Cheriyan, Devdatt Dubhashi, Radhakrishnan Jagadeesan, Prasad Jayanti, Steve Mitchell, Sendhil Mullainathan, Moni Naor, David Pearson, Desh Ranjan, Sridhar Sundaram, Steve Vavasis, and Vijay Vazirani for valuable discussions.

This work has been supported in parts by NSF PYI award CCR–89–96272 with matching support from UPS and Sun Microsystems, by the U. S. Army Research Office through the Mathematical Sciences Institute of Cornell University, and by an IBM Graduate Fellowship. I am grateful to these sources for their support.

I thank the faculty, staff, and students of the Cornell Computer Science department for maintaining such a friendly environment. Dexter Kozen and Sam Toueg were nothing but supportive in their roles as Graduate Field Representative, while in the midst of busy academic careers. Jan Batzer could be counted on to make every bureaucratic hassle vanish, while always maintaining a warm and sweet smile. The department offered me Teaching Assistantships for six semesters and a summer, and I have learnt a lot from this experience; thanks to Bob Constable, Bruce Donald, Doug Howe, Keshav Pingali, Amitabh Shah and Kay Wagner, for whom I have been a TA. My office-mates Prasad Jayanti, Shyam Kapur and Adam Webber were models of serenity, whenever I panicked over trivia. I am also thankful to Cornell University for offering such a nice environment for study, and to the staff of the Cornell International Students and Scholars Office for helping smoothen the path for international students and scholars in so many ways.

I spent Spring 1993 at Princeton as an exchange student; I would like to thank Princeton University in general, and the DIMACS Center, Bob Sedgewick and Bob Tarjan in particular, for making this possible. I was fortunate to take a wonderful course on "Communication and Computation" taught by Andy Yao at Princeton. My heartfelt thanks to Sharon Rodgers, for being so kind and helpful at Princeton.

In addition to my parents, my entire family has been so loving and supportive of me. Right from childhood, my sisters Smt. Maithreyi Sunderrajan (Maith Akka) and Smt. Sowmya Sridharan (Chamakka), and my brother Prof. P. S. Giridharan (Giri Anna) have tirelessly impressed upon me the value of education; I cannot adequately measure their affection for me. Though he would never accept it, my brother has always been a major source of inspiration for me- in his academic excellence and diligence when I was young, and in the way he conducts his life, now. Above all, he was the one who introduced me to Guruji. My sincere thanks are also due to my brothers-in-law Shri. C. P. Sunderrajan (Sunder Athimbare) and Shri. P. Sridharan (Sridhar Athimbare), and to my sister-in-law Smt. Asha Giridharan (Asha) for their

encouragement.

There have been many who have been sources of good fun, and have offered solace at times of confusion and depression. In particular, I would like to thank ...

Prasad Jayanti, for the discussions on India, movies and much more, and Shanta and him for kindly putting me up in their house when I was down with pneumonia during one of Ithaca's cold and desolate winter breaks and for always keeping a warm family atmosphere open for me;

Alessandro Panconesi, whose spontaneity, wit, and several hilarious e-mail messages made joint work delightful, and who offered me a broader picture of life and never allowed me to take myself seriously;

Pal and Sridhar, for the continued friendship since our undergrad days and for the good times;

Alok Baveja, for being a kind friend whose frank criticism I have always benefited from and who is a source of inspiration in many ways;

Mike Kalantar the Ever-helpful, for tirelessly inviting me to every Bahá'ícelebration in town and for teaching me the basics of the Bahá'ífaith;

Dr. Thangavelu for re-kindling my interest in Thamizh and for introducing me to many authors, and to Smt. Uma and him for keeping their house always open to me, and

Venky, for the memorable long walks in scenic Ithaca, our friendship, and for his timely help in various matters.

I must have certainly forgotten to thank here the many other people who have influenced me, but my gratitude for them will certainly remain in my heart.

Table of Contents

Chapter 1

Introduction

Randomness is well-recognized as an important computational resource in theoretical computer science now. Not only are there classical problems such as primality testing for which the only known "efficient" solutions are randomized, but there are problems for which randomness can be used to circumvent deterministic lower bounds, *e.g.*, approximating the volume of a convex body. The theoretical advances are also slowly being realized in practice, *e.g.*, the CM-5 parallel computer uses randomized routing for communication among its processors. However, standard tools available for probabilistic analysis such as the Chernoff–Hoeffding bounds are not sufficiently powerful in all situations; second, since there are no known "true" sources of randomness, there is a need to develop general techniques for reducing/removing randomness from randomized algorithms. This thesis addresses these two issues.

There are at least three striking advantages offered by randomness, *i.e.*, the ability to "flip a fair coin" any number of times independently, to computation. First, there are classical problems such as primality testing and other number-theoretic problems which are basic, for instance, to cryptographic protocols, for which the only known polynomial-time algorithms are randomized (see, for example, Solovay & Strassen [116], Rabin [98], and Adleman & Huang [2]). Furthermore, there are classical problems for which the best known deterministic algorithms start with a randomized construction and then derandomize it, *e.g.*, testing if an undirected graph is connected, within very limited space bounds (Nisan [87,88], Nisan, Szemerédi & Wigderson [89]). Third and most strikingly, there are several areas in which randomization can be used to circumvent lower bounds for deterministic algorithms– fault-tolerant Byzantine agreement in asynchronous distributed systems, packet routing in interconnection networks, communication complexity theory etc. A recent important illustration of this is as follows. Given a convex body in n dimensions presented in a certain natural form, one can approximate its volume only to within an exponential factor of n, deterministically in time polynomial in n (Elekes [37], Bárány & Füredi [13]); however, given any $\epsilon > 0$, Dyer, Frieze & Kannan show how to approximate this quantity to within ϵ in time polynomial in n and $1/\epsilon$, via randomization [36].

The above reasons, combined with the general simplicity of, and the interesting mathematical techniques brought forth by, randomized algorithms, make randomized algorithms an interesting topic of study. See Gill [42] for a complexity-theoretic grounding for randomized algorithms, Kozen [66] for a proposed semantics for randomized algorithms, and Karp [58], Raghavan [101] and Tompa [125] for some recent surveys of issues in randomized computation.

Two orthogonal issues in randomized computation are studied in this thesis. On the one hand, standard tools available for probabilistic analysis such as the Chernoff–Hoeffding bounds [27,50], are not sufficiently powerful in all situations; more tools and techniques are needed to design and analyze efficient randomized algorithms. In particular, one can say informally that while the behavior of independent random variables is fairly well understood, there is a need to develop more tools for analyzing the behavior of random variables displaying various types of correlation. On the other hand, there are no known "perfect" sources of randomness, *i.e.*, those that can output any number of unbiased, and more importantly, *independent* random bits; thus, we need to develop general techniques for reducing and if possible, even removing the randomness in randomized algorithms, without significantly increasing their complexity. To quote Karp [58],

> It is clear that random bits, if they can be produced at all, will be slow and costly to generate. For this reason, there has been considerable interest in reducing the number of truly random bits that algorithms require, or else showing that imperfect sources of randomness are adequate.

We emphasize, however, that in practice, randomized algorithms seem to work quite well, even with the linear congruential generator as their source of randomness. The main motivations for studying randomness–efficient computation under this reality are to have a theoretically sound basis for randomized algorithms, and to make some advances towards settling the complexity-theoretic question of classifying problems according to the amount of randomness they need, to be solved efficiently. Furthermore, such research opens up the possibility of deriving efficient *deterministic* algorithms from their probabilistic counterparts, if the randomness requirement is made small enough.

Thus, this thesis investigates these two issues of developing new techniques for analyzing randomized algorithms, and inventing general methods to reduce/remove the randomness from randomized algorithms.

1.1 Related Work

We now give a quick and informal, far from comprehensive, survey of some important existing tools for probabilistic analysis and randomness-efficient computation. For the purposes of this thesis, we are interested in tools for probabilistic analysis of a certain type–upper bounds on the probabilities of deviation of certain random variables from their mean. The Chernoff–Hoeffding bounds are basic tools used in bounding the tail probabilities of the sums of bounded and independent random variables. They are commonly used in randomized algorithms and in derandomization to upper bound $Pr(|X - \mu| \geq a)$, where $X = \sum_{i=1}^{n} X_i$ and $\mu = E[X]$, with $X_1, X_2, \ldots, X_n$

being bounded and *independent* random variables. An informal scenario where they might typically be used is where X denotes the (random) value taken on by a computational resource, say time, in a randomized algorithm, and the algorithm performs "well" when X is "close to" its expected value μ; such bounds show that with high probability, the algorithm indeed performs well. In the case of martingales, *i.e.*, sequences of random variables $X_0, X_1, \ldots$ such that $\forall i \geq 0 \; E[X_{i+1}|X_i] = X_i$, ideas from the Chernoff–Hoeffding bounds have been used in Azuma's inequality to upper bound $Pr(X_i \geq \lambda\sqrt{i})$, when $X_0 \equiv 0$ and $|X_{i+1} - X_i| \leq c$ for some constant c (see Alon, Spencer & Erdős [8]). See Chapters 5-8 of [8] for some other useful probability inequalities for correlated random variables.

The following are some known general approaches towards randomness-efficient computation.

(i) *Pseudorandomness.* A general method for reducing randomness is the theory of *pseudorandom number generators* (PSRGs) initiated by Blum & Micali [18] and Yao [139]. A PSRG is a sequence of functions $\{g_n\}$, $g_n : \{0,1\}^n \rightarrow \{0,1\}^{p(n)}$, which *deterministically* "stretch" a given seed to a much longer seed; $p(n)$ could be n^2, for instance. A PSRG is such that no polynomial time test can "distinguish between" $g_n(x)$ and y, where x is a truly random string of length n and y is a truly random string of length $p(n)$, even with the help of true randomness. Thus, a polynomial-time implementable PSRG can be used to safely run a randomized algorithm which requires $p(n)$ random bits, using just n random bits. An intimate connection- equivalence- between PSRGs and one-way functions has been found (Impagliazzo, Levin & Luby [52], Håstad [49]); since the existence of one-way functions is only a conjecture so far, there are no known provable PSRGs. This idea has inspired the next method, which is more limited in purpose but which does not require any complexity-theoretic assumptions.

(ii) *Using available random number generators.* Bach pioneered the idea of running randomized algorithms judiciously by taking a very short truly random seed, and then stretching it using standard random number generators such as the linear congruential generator to still guarantee the performance of these algorithms [12]; his main focus was on certain number-theoretic algorithms. Karloff & Raghavan have used this idea to judiciously run a randomized version of Quicksort, routing algorithms on the hypercube etc. [56], and Mulmuley has recently used it for certain geometric algorithms [82].

Given a randomized algorithm which requires n random bits, suppose one could analyze it by some method which only requires these bits to have some weaker stochastic property than that of being unbiased and independent (which clearly requires n random bits). One can then hope for some savings in randomness, by implementing the weaker property. The following two approaches exploit this; see also Schulman [111] and Koller & Megiddo [65] for related work.

(iii) *Limited independence.* Random variables $X_1, X_2, \ldots, X_n$ are *k–wise independent* if every k (or fewer) of them are mutually independent. Note, for instance, that if a random variable X is such that $X = \sum_i f_i(X_1, X_2, \ldots, X_n)$ where each f_i is a function of *at most* k of $X_1, X_2, \ldots, X_n$, then $E[X]$ is the same if $X_1, X_2, \ldots, X_n$ are mutually independent or are k–wise independent; thus, if X denotes the running time of a randomized algorithm, then the expected running time will be the same under these two situations and hence, we just require k–wise independent $X_1, X_2, \ldots, X_n$.

A very important property of such random variables is that they can be generated from $O(k \log n)$ truly random bits, when, for instance, each X_i is an unbiased random bit (Joffe [54], Carter & Wegman [24]). So, if k is a constant, then the sample space is polynomial-sized in n ($n^{O(k)}$), and thus we can eliminate randomness via searching all the points in the sample space, incurring just a polynomial overhead in time. This is a very powerful, oft-used technique. It was used implicitly by Karp & Wigderson in their breakthrough result that computing a maximal independent set in a graph is in NC [62], and was made explicit later by Luby [75].

(iv) *Biased random variables.* A sample space X for n–bit vectors was defined to be k–wise ϵ–biased by Naor & Naor [85] (see also Vazirani [132]) if

$$\forall S \subseteq \{1,2,\ldots,n\}, 1 \leq |S| \leq k, |Pr(\bigoplus_{i\in S} x_i = 1) - Pr(\bigoplus_{i\in S} x_i = 0)| \leq \epsilon,$$

where $\bigoplus$ denotes the XOR function and x_i denotes the ith bit of an n-bit string x picked uniformly at random from X. One important property of such a sample space is that $\forall \ell, \ell = 1,2,\ldots,k, \forall \{i_1, i_2, \ldots i_\ell\} \subseteq \{1,2,\ldots,n\}, \forall b_1 b_2 \ldots b_\ell \in \{0,1\}^\ell$,

$$|Pr(x_{i_1} = b_1, x_{i_2} = b_2, \ldots, x_{i_\ell} = b_\ell) - \frac{1}{2^\ell}| \leq \epsilon.$$

Constructions of k–wise ϵ–biased sources of size $poly(k, \log n, \frac{1}{\epsilon})$ were presented in [85]. Such sample spaces have been shown to have several applications to explicit constructions and to derandomization, mainly since probabilistic analyses may be expected to be robust under small perturbations of the probabilities. These results have been extended by Azar, Motwani & Naor [11], Alon, Goldreich, Håstad & Peralta [6], Even, Goldreich, Luby, Nisan & Veličković [41] and Chari, Rohatgi & Srinivasan [25].

(v) *Deterministic amplification.* Given a randomized algorithm $\mathcal{A}$ which uses r random bits and succeeds with probability p, one can easily boost its success probability to $q > p$, by just running it $\log_{(1-p)}(1-q)$ times independently (assuming that we can detect the algorithm's "success"). However, this requires $r \cdot \log_{(1-p)}(1-q)$ random bits; can we do better? Karp, Pippenger & Sipser [59] and Sipser [115] reduce this problem to that of the explicit construction of certain types of graphs (dispersers and expanders). Ajtai, Komlos & Szemerédi show how to do this via random walks on expanders [3], a connection further explored by Cohen & Wigderson [31] and

$Pr(X \geq \mu(1+\delta))$ can be upper bounded by the Chernoff-Hoeffding upper bound, even if $k \geq \mu \cdot \delta$. Additional methods are also presented, and the aggregate results are very sharp and provide a better understanding of the proof techniques behind these bounds. They also yield improved bounds for various tail probability distributions and enable improved approximation algorithms for jobshop scheduling. The "limited independence" result implies that weaker sources of randomness are sufficient for randomized algorithms whose analyses use the Chernoff–Hoeffding bounds; further, it leads to algorithms that require a reduced amount of randomness for any analysis which uses the Chernoff–Hoeffding bounds, *e.g.*, the analysis of randomized algorithms for random sampling and oblivious packet routing. This is joint work with J. P. Schmidt and A. Siegel [108].

In Chapter 4, we study the *parallel randomness complexity* of perfect matching and related problems. We present a randomness–efficient RNC^2 algorithm for the perfect matching problem which uses $O(\log Z + \log n)$ random bits where Z is any given upper bound on the number of perfect matchings in the graph. This improves and generalizes the results of Grigoriev & Karpinski [47] who present an NC^3 algorithm when Z is polynomially bounded. In the worst case the algorithm uses $O(n \log(m/n))$ random bits, improving on the previously best-known $\Theta(m \log n)$ randomness complexity of Mulmuley, Vazirani & Vazirani. When no such upper bound is known, this gives an RNC^3 algorithm to find a perfect matching which uses at most $O(\log M + \log n)$ random bits with high probability, provided that M, the number of perfect matchings, is nonzero. See Karp & Ramachandran [60] for a study of the complexity class NC. The core of our algorithms is a randomness–optimal generalization of the Isolating Lemma of Mulmuley, Vazirani & Vazirani [83]. Given a set S and an unknown family $\mathcal{F} \subseteq 2^S$ with $|\mathcal{F}| \leq Z$, we outline a scheme to assign polynomially bounded weights to the elements of S using only $O(\log Z + \log |S|)$ random bits, such that the minimum weight set in $\mathcal{F}$ is unique with high probability. We also prove a matching lower bound for the randomness

Impagliazzo & Zuckerman [53], who show, when p is at least some constant, how to do this using $O(r + \log_{(1-p)}(1-q))$ random bits. Nisan shows, using very different techniques, how this can be approximated very well, using $O(r \log(\log_{(1-p)}(1-q)))$ random bits [87].

The final technique we sketch is of a very different flavor, and is an important tool for derandomization.

(vi) *The method of conditional probabilities.* This method is implicit in the work of Erdős & Selfridge [40], and has been systematized by Raghavan [100] and Spencer [117]. Given a joint distribution D on n (say, binary) random variables $X_1, \ldots, X_n$ such that

$$E_D[f(X_1, X_2, \ldots, X_n)] \leq a,$$

for some function f and some constant a, call a point $\vec{x} = (x_1, x_2, \ldots, x_n) \in \{0, 1\}^n$ *good* if $f(\vec{x}) \leq a$, and suppose we want to find any good point $\vec{g} = (g_1, g_2, \ldots, g_n)$. Clearly, some good point exists; how to find it efficiently? The following is one such method.

for i:= 1 **to** n **do**

if

$$E_D[f(\vec{X}) \mid X_i = 0, \wedge_{j=1}^{i-1}(X_j = g_j)] \;\leq\; E_D[f(\vec{X}) \mid X_i = 1, \wedge_{j=1}^{i-1}(X_j = g_j)]$$

then $g_i := 0$

else $g_i := 1$.

The key claim is that

$$E_D[f(\vec{X}) \mid \bigwedge_{j=1}^{i+1}(X_j = g_j)] \leq E_D[f(\vec{X}) \mid \bigwedge_{j=1}^{i}(X_j = g_j)], \; i = 0, \ldots, n-1,$$

which implies that

$$f(\vec{g}) = E_D[f(\vec{X}) \mid X_j = g_j, j = 1, 2, \ldots, n] \leq E_D[f(\vec{X})] \leq a,$$

i.e, that $\vec{g}$ is good. Thus, if we can compute the above conditional probabilities efficiently, we can zoom in on a good point fast. Typically, we cannot compute these conditional probabilities exactly in an efficient manner, but Raghavan shows how certain approximations to them (which he calls *pessimistic estimators*) are sufficient [100].

1.2 Contributions and organization of the thesis

Certain types of routing, scheduling and resource allocation problems in a distributed setting can be modeled as edge coloring problems; an edge coloring of a graph is an assignment of colors to its edges such that no two edges incident at the same vertex get the same color. In Chapter 2, we present a fast and simple randomized algorithm for edge coloring a graph, in the synchronous distributed point–to–point model of computation. Our algorithm computes an edge–coloring of a graph G with n nodes and maximum degree Δ with at most $(1.6 + \epsilon)\Delta + O(\log^{2+\delta} n)$ colors with high probability (arbitrarily close to 1), for any fixed $\epsilon, \delta > 0$. To analyze the performance of our algorithm, we introduce new techniques for proving upper bounds on the tail probabilities of certain random variables. Our results extend the Chernoff–Hoeffding bounds to certain types of random variables which are not stochastically independent. This is joint work with A. Panconesi [91]. This technique has been recently used by us [118] to improve certain known results on the distribution of the number of prime factors of integers (Hardy & Ramanujan [103], Erdős & Kac [39], Turán [126]), and has also served in part as motivation for the material presented in Chapter 3.

In Chapter 3, we present a simple technique which gives slightly better bounds than the Chernoff-Hoeffding bounds for the tail probabilities of the sums of bounded and independent random variables, and which more importantly requires only *limited independence* among the random variables, thereby importing a variety of standard results to the case of limited independence "for free". In particular, we show that if $X_1, X_2, \ldots, X_n$ are bounded between 0 and 1 and are k–wise independent, then complexity of this problem.

Due to the wide applicability of the Isolating Lemma, this generalization yields randomness–efficient RNC algorithms for several other problems such as variants on matching and basic problems on linearly representable matroids such as matroid intersection and matroid matching. This also gives a randomness–efficient version of the Valiant-Vazirani reduction from SAT to $USAT$ [131], which yields new proofs that $FewP \subseteq \oplus P$ (Cai & Hemachandra [22]) and $FewP \subseteq C_{=}P$ (Köbler, Schöning, Toda & Toran [64]). This is joint work with S. Chari and P. Rohatgi [26].

Chapter 5 concludes and presents some open problems.

We have attempted to keep each chapter self-contained. For the reader in a hurry, we recommend Sections 2.2.1, 2.3, 3.1.1 and 4.1 to collect the essence of this work.

Chapter 2

An Extension of the Chernoff-Hoeffding Bounds with an Application to Distributed Edge Coloring

An important limitation for a distributed network without global memory in computing any function is that for an efficient algorithm, each processor can communicate only with those processors that are within a small radius around it. Models of parallel computation like the PRAM abstract this problem of locality away by assuming the existence of a global shared memory with fast concurrent access. We are interested in studying how fast individual processors can compute their portion of the output in a message–passing distributed system, with such "local" information alone. The model we study in this chapter is the synchronous distributed point–to–point model, in which the processors are arranged as the vertices of an n–vertex graph $G = (V, E)$, and where all communication is via the edges of G alone. This chapter describes joint work with A. Panconesi [91].

An edge coloring of a graph is an assignment of colors to its edges such that no two edges incident at the same vertex get the same color. We present a fast and simple randomized distributed algorithm to edge color G with at most $(1.6 + \epsilon)\Delta + O(\log^{2+\delta} n)$ colors with high probability for any fixed $\epsilon, \delta > 0$, where Δ is the maximum degree of the vertices of G.

One main contribution of this chapter is the tools developed to analyze our algorithm. The *Chernoff–Hoeffding* (henceforth CH) bounds [27,50] are fundamental tools used in bounding the tail probabilities of the sum of bounded and independent random variables. The most frequent form in which these bounds are used in computer science is when there are n *independent* random bits $X_1, X_2, \ldots, X_n$, $X = \sum_{i=1}^{n} X_i$, and $\mu = E[X]$; in this case, these bounds show that $Pr(X \geq (1+\delta)\mu)$ is inverse exponential in μ and δ^2, for $0 < \delta \leq 1$. We introduce a new way of looking at the CH bounds and prove that CH-type bounds can be used for the sums of certain types of dependent random variables too. A generalization to a related form of dependence is known (Raman [102]), but it is not strong enough to be used in our applications. We also prove similar results to certain types of dependent non-binary random variables. These new extensions of the CH bounds are used crucially in the analysis of our algorithms. Furthermore, these methods have served partly as motivation for recent work of Schmidt, Siegel & Srinivasan [108] and Srinivasan [118].

The edge coloring problem can be used to model certain types of jobshop scheduling, packet routing, and resource allocation problems in a distributed setting. For example, given a set of processes $\mathcal{P}$ and a set of resources $\mathcal{R}$ such that each process $p \in \mathcal{P}$ needs a subset $f(p) \subseteq \mathcal{R}$ of the resources where: (i) each process p needs every resource in $f(p)$ for a unit of time each, and (ii) p can use the resources in $f(p)$ in any order, we can construct a bipartite graph $G_{\mathcal{P},\mathcal{R}} = (\mathcal{P}, \mathcal{R}, E_{\mathcal{P},\mathcal{R}})$ where $E_{\mathcal{P},\mathcal{R}} = \{(p, r) | p \in \mathcal{P} \wedge r \in f(p)\}$, and an edge coloring of $G_{\mathcal{P},\mathcal{R}}$ with c colors yields a schedule for the processes to use the resources within c time units. It has also been used in resource allocation problems such as the Dining Philosophers problem

(Lynch [77], Styer & Peterson [120]). Edge coloring can also be used in distributed models in situations where broadcasts are infeasible or undesirable: an edge coloring of the network results in a schedule for each processor to communicate with at most one neighbor at every step (at time step i, processors communicate via the edges colored i only), and using a "small" number of colors reduces the wastage of time in this schedule.

Note that Δ colors are necessary to edge color a graph with maximum degree Δ. Vizing showed that it is always possible to edge color a graph with $\Delta + 1$ colors and gave a polynomial time algorithm [135] (see, for example, Bollobás [19]). Karloff & Shmoys gave an RNC algorithm for this problem that uses $\Delta + \Delta^{0.5+\epsilon}$ colors, which was derandomized in NC by Berger & Rompel and Motwani, Naor & Naor [17,81]. In the distributed model, a straightforward edge coloring algorithm is to apply a vertex coloring algorithm to the line graph of G. There are fast (polylogarithmic) randomized vertex coloring algorithms that use $(\Delta+1)$ and Δ colors, which translate to $(2\Delta-1)$- and $(2\Delta-2)$- edge coloring algorithms respectively (Luby [76], Panconesi & Srinivasan [92]). There are no known polylogarithmic deterministic algorithms in the distributed setting for $(2\Delta - 1)$–edge coloring; moreover, distributed Δ–edge coloring requires $\Omega(diameter(G))$ time, even with randomization [92].

It has recently come to our attention through Alon & Spencer [7], that methods of Rödl [106] and Pippenger & Spencer [96] can be used to derive randomized polylogarithmic distributed algorithms to edge color using $\Delta(1 + o(1))$ many colors. However, we present the material in this chapter with the view that the analysis is of independent interest, and also in view of the new extension of the CH bounds and its applications.

This work also belongs to the realm of studying how a distributed system can solve some problem (related to scheduling and synchronization) on its own topology. The main problems that have been studied in this regard are computing a maximal independent set (MIS) in G (Alon, Babai & Itai [5], Luby [75]), a $(\Delta + 1)$–vertex coloring of G (Awerbuch, Goldberg, Luby & Plotkin [10], Luby [76]), and a Δ–vertex coloring of G [92]. An MIS captures the idea of a set of processors working in parallel without interfering with the decisions made by their neighbors, and a vertex coloring is a partition of the processors into independent sets, yielding a schedule for the processors to work in parallel.

An important point about all of these problems is that they need just a *local search* for an *incremental update*, in the following sense. Consider the simple sequential algorithm to compute an MIS– pick any vertex v, add it to the partial MIS computed so far, remove v and its neighbors from G, and repeat. Thus, a partial MIS can be updated incrementally with a local search of radius 1. Similar "local search" results for incremental updates are known for $(\Delta+1)$– and Δ–vertex coloring (and hence for $(2\Delta-1)$– and $(2\Delta-2)$–edge coloring): local searches of radius 1 and radius $O(\log_\Delta n)$ respectively [92]. Hence, the key problem in implementing these in a parallel setting is *symmetry breaking*: parallelizing the incremental sequential algorithm (implied by such a local result) by somehow breaking the symmetrical effort of all the processors to execute one incremental step of the sequential algorithm. Two key tools for symmetry breaking are randomization [5,75,76] and *network decomposition* (Awerbuch, Goldberg, Luby & Plotkin [10], Linial & Saks [72], Berger & Cowen [16], Panconesi & Srinivasan [92]).

However, such a local search result is not known for edge coloring with less than $2\Delta-2$ colors. Note that if we are allowed $2\Delta-1$ colors, an uncolored edge is always surrounded by at most $2\Delta-2$ colors, and a local search of radius 1 is sufficient for the update. It is not clear that any local result should hold when we have at most $2\Delta-3$ colors. Hence, another main contribution of this chapter is a fast and simple distributed algorithm for a problem not falling under the hitherto dominant paradigm of "symmetry breaking and local search". Our algorithm is also very simple; we first present an algorithm for edge coloring bipartite graphs, and then extend it to general graphs using an idea of Karloff & Shmoys [57].

2.1 Notation

Given an undirected graph $G = (V, E)$ we denote by Δ its maximum degree, *i.e.*, the maximum number of edges incident at any node; by d_u we denote the degree of vertex u, and by $\delta(u)$ we denote the set of edges incident at vertex u.

Given a positive integer n, $[n]$ denotes the set $\{1, 2, \dots, n\}$. The permanent of a (possibly non–square) matrix $M \in \Re^{r \times c}$ with $c \leq r$ is defined as the natural extension of the permanent of square matrices. Let $\mathcal{P} = \{\pi \mid \pi : [c] \to [r],\ \pi \text{ is one–to–one}\}$. Then,

$$perm(M) \doteq \sum_{\pi \in \mathcal{P}} \prod_{i=1}^{c} M_{\pi(i),i}.$$

An event $\mathcal{A}$ is said to happen *with high probability* (*w.h.p.*) if $Pr(\mathcal{A}) \geq 1 - 1/f(n)$ for some superpolynomial function $f(n)$ (*i.e.*, $n^c = o(f(n))$ for all $c > 0$, and the failure probability of $\mathcal{A}$ is at most $1/f(n)$).

2.2 The Chernoff-Hoeffding Bounds Extension

In this section we introduce our extension of the Chernoff-Hoeffding bounds, which are important tools used in estimating the tail probabilities of random variables. Given n *independent* and bounded random variables $X_1, X_2, \dots, X_n$, these bounds are used in deriving an upper bound on the upper tail probability $Pr(X \geq (1+\epsilon)\mu)$, where $X = \sum_{i=1}^{n} X_i$, $\mu = E[X]$, and $\epsilon > 0$. We extend these bounds to a certain case of dependency among the X_i's, which we call λ–correlation. We first introduce this notion in the case where the X_i's are 0-1 random variables and prove our extension of the CH bounds. This special case is simpler to handle and more intuitive but contains all the essential difficulties of the general case. After the special case is dealt with, we will take care of the more general case.

Let us review Chernoff's classical approach to upper bound the upper tail probability of a random variable X, which is the sum of *independent* binary random variables $X_1, X_2, \ldots, X_n$ [27] (this approach apparently dates back to S. N. Bernstein). Let $Pr(X_i = 1) = p_i)$ and $\mu \doteq E[X] = \sum_{i=1}^{n} p_i$. We want good upper bounds on $Pr(X \geq \mu(1+\epsilon))$, for $\epsilon > 0$. Chernoff implicitly showed that for identically distributed 0–1 variables $X_1, X_2, \ldots X_n$ and for $a > \mu$,

$$Pr(X \geq a) \leq min_t \frac{E[e^{tX}]}{e^{at}} \leq L(n, \mu, a) = (\frac{\mu}{a})^a (\frac{n-\mu}{n-a})^{n-a}.$$

Hoeffding [50] extended this by showing that $L(n, \mu, a)$ is an upper bound for the above minimum even if the X_i's are not identically distributed and range between 0 and 1. Replacing a by $\mu(1+\epsilon)$ in the Hoeffding estimate $L(\cdot,\cdot,\cdot)$ gives, for $\epsilon > 0$,

$$Pr(X \geq \mu(1+\epsilon)) \leq F(n, \mu, \epsilon) \doteq \frac{(1 + \frac{\mu\epsilon}{(n-\mu(1+\epsilon))})^{n-\mu(1+\epsilon)}}{(1+\epsilon)^{\mu(1+\epsilon)}}.$$

Since $L(n, \mu, a)$ is symmetric with respect to (a, μ) and $(n-a, n-\mu)$, the Hoeffding estimate also shows that

$$Pr(X \leq \mu(1-\epsilon)) = Pr(n-X \geq n-\mu(1-\epsilon)) \leq F(n, \mu, -\epsilon) \doteq \frac{(1 - \frac{\mu\epsilon}{(n-\mu(1-\epsilon))})^{n-\mu(1-\epsilon)}}{(1-\epsilon)^{\mu(1-\epsilon)}}.$$

The following simple upper bounds for $F(n, \mu, \epsilon)$ and $F(n, \mu, -\epsilon)$ are sufficient to derive most of the useful approximations that have appeared in the literature [50,9, 101,8].

$$F(n, \mu, \epsilon) \leq G(\mu, \epsilon) \doteq (\frac{e^\epsilon}{(1+\epsilon)^{1+\epsilon}})^\mu \text{ (see, for example, [101]);} \tag{2.1}$$

for $\epsilon < 1$, $\quad G(\mu, \epsilon) \leq e^{-\epsilon^2 \mu/3}$ [9]; $\qquad$ for $\epsilon > 2e-1$, $\quad G(\mu, \epsilon) \leq 2^{-(1+\epsilon)\mu}$ [101], and

$$F(n, \mu, -\epsilon) \leq G(\mu, -\epsilon) \leq e^{-\mu\epsilon^2/2} \quad [50, 9, 101, 8].$$

Hoeffding also considers the more general case where X is the sum of n independent and *bounded* random variables $X_i \in [a_i, b_i]$, and uses the above approach to show that if $E[X] = \mu$, then for $\epsilon > 0$,

$$\min_{t>0} \frac{E[e^{tX}]}{e^{t(1+\epsilon)\mu}} \leq H(\mu, \epsilon, \vec{a}, \vec{b}) \doteq \exp\left[-\frac{2\,\mu^2\epsilon^2}{\sum_{i\in[n]}(b_i - a_i)^2}\right]. \tag{2.2}$$

Inequalities (2.1) and (2.2) will be used in our proofs. If ϵ is a fixed positive quantity no greater than 1 (which will be true in all our applications), then $G(\mu, \epsilon) \leq e^{-\epsilon^2\mu/3}$. Hence, if $\mu = \Omega(\log^{1+\delta} n)$, then $G(\mu, \epsilon)$ is the inverse of a superpolynomial function of n, for any fixed $\delta > 0$ (similar considerations apply to $H(\mu, \epsilon, \vec{a}, \vec{b})$, when each $b_i - a_i$ is bounded). This fact makes the CH-bounds a powerful tool for deriving strong performance guarantees for randomized algorithms and will be used repeatedly in this chapter.

2.2.1 Binary Random Variables

We now present our extension of the CH-bounds to the case of 0-1 λ–correlated random variables. We begin by defining λ–correlation (this was defined as "self–weakening with parameter λ" in [91]) for 0-1 random variables.

Definition 1 *Let $X_1, X_2, \ldots, X_n$ be 0-1 random variables. The X_i's are λ*–correlated *if, for all nonempty $I \subseteq [n]$,*

$$Pr\left(\bigwedge_{i\in I} X_i = 1\right) \leq \lambda \prod_{i\in I} Pr(X_i = 1).$$

Before proving the extension of the CH-bounds, let us develop some intuition by giving an example of λ–correlation. Suppose Δ balls are thrown uniformly at random and independently into Δ bins. Let B_i be an indicator random variable that is 1 if bin i is empty and 0 otherwise. These variables are 1–correlated. To see this, consider any nonempty subset $J \subseteq [\Delta]$ of the bins and suppose that all these bins are empty. Then, given this, the probability that some other bin, say i, remains empty decreases. That is, for all $J \subseteq [\Delta]$ and $i \notin J$,

$$Pr(B_i = 1 \mid \bigwedge_{j\in J} B_j = 1) = \left(1 - \frac{1}{\Delta - |J|}\right)^{\Delta} \leq \left(1 - \frac{1}{\Delta}\right)^{\Delta} = Pr(B_i = 1).$$

By straightforward induction, this implies that for all nonempty $I \subseteq [\Delta]$,

$$Pr\left(\bigwedge_{i\in I} B_i = 1\right) \leq \prod_{i\in I} Pr(B_i = 1).$$

The CH-bounds say that if X is the sum of n *independent* 0-1 random variables then $Pr(X > (1+\epsilon)\mu) \leq G(\mu,\epsilon)$. The next theorem shows that if the X_i's are λ–correlated then $Pr(X > (1+\epsilon)\mu) \leq \lambda\, G(\mu,\epsilon)$. In the statement of the following theorem, μ is an upper bound on $E[X]$; this is sufficient because we are upper bounding the upper tail probability of X.

Theorem 1 *Let X be the sum of λ-correlated 0-1 random variables, and let $E[X] \leq \mu$. Then,*

$$Pr(X > (1+\epsilon)\mu) \leq \lambda\, G(\mu,\epsilon).$$

PROOF. As in the classical proof, we start by introducing a positive parameter t and by applying Markov's inequality to the variable e^{tX},

$$Pr(X > (1+\epsilon)\mu) = Pr(e^{tX} > e^{t(1+\epsilon)\mu}) \leq \frac{E[e^{tX}]}{e^{t(1+\epsilon)\mu}}.$$

To find a good upper bound on the numerator (we cannot use independence of the X_i's) we write down its Maclaurin expansion, and use the fact that e^{tX} is bounded by e^{tn} to apply linearity of expectation to the infinite series:

$$E[e^{tX}] = E\left[\sum_{k=0}^{\infty} \frac{t^k X^k}{k!}\right] = \sum_{k=0}^{\infty} \frac{t^k E[X^k]}{k!}.$$

Focus on the generic term $E[X^k]$ of this series. From the facts that $X = \sum X_i$ and that the X_i's are 0-1 random variables it follows that $E[X^k]$ is a linear combination of the form

$$E[X^k] = \sum_{I \subseteq [n]} \alpha_I \; Pr\left(\bigwedge_{i \in I} X_i = 1\right)$$

for $I \neq \emptyset$, and where all α_I's are non-negative (this can be verified by unfolding X^k and by observing that $X_i^c = X_i$ for all i and all $c > 0$). In order to upper bound the term $E[X^k]$ we introduce n twin 0-1 random variables $\hat{X}_i$ which have the same marginal (individual) distributions as the X_i's except that they are independent. From the hypothesis of λ–correlation and from the fact the twin variables are independent it follows that

$$Pr\left(\bigwedge_{i\in I} X_i = 1\right) \leq \lambda \prod_{i\in I} Pr(X_i = 1) = \lambda \prod_{i\in I} Pr(\hat{X}_i = 1) \ = \lambda\ Pr\left(\bigwedge_{i\in I} \hat{X}_i = 1\right).$$

Hence, $E[X^k] \leq \lambda\ E[\hat{X}^k]$. By upper bounding each term of the series expansion of $E[e^{tX}]$, we get

$$E[e^{tX}] \leq \lambda\ E[e^{t\hat{X}}].$$

Note that $\hat{X}$ is the sum of n *independent* random variables and $E[\hat{X}] = E[X] \leq \mu$. Hence, by inequality (2.1),

$$Pr(X > (1+\epsilon)\mu) \leq \frac{E[e^{tX}]}{e^{t(1+\epsilon)\mu}} \leq \lambda\ \frac{E[e^{t\hat{X}}]}{e^{t(1+\epsilon)\mu}} \leq \lambda\ G(\mu, \epsilon).$$

□

To see an application of this theorem, let us go back to the balls-and-bins experiment, and suppose we are interested in estimating the number of empty bins after the balls are thrown; this number is given by the random variable $B = \sum_i B_i$. By linearity of expectation and by the fact that the balls are thrown at random independently of each other,

$$E[B] = \sum_{i\in[\Delta]} E[B_i] = \sum_{i\in[\Delta]} \left(1 - \frac{1}{\Delta}\right)^{\Delta} \leq \frac{\Delta}{e}.$$

We already saw that the B_i's are 1–correlated. Hence, by Theorem 1,

$$Pr(B > (1+\epsilon)\Delta/e) \leq G(\Delta/e, \epsilon).$$

A very different proof of this result also appears in ([8], page 91), using the theory of martingales.

Remark. Jain has proved the following lemma [102]: *Let $a_1, a_2, \ldots, a_n$ be n random trials (not necessarily independent) such that the probability that trial a_i 'succeeds'* *is bounded above by p regardless of the outcomes of the other trials. Then if X is the random variable that represents the number of 'successes' in these n trials, and Y is a binomial variable with parameters (n,p), then:* $Pr[X \geq k] \leq Pr[Y \geq k]$, $0 \leq k \leq n$.

The assumptions of Jain's lemma are strictly stronger than those of 1–correlation. For instance, in the balls-and-bins example,

$$Pr\left(B_\Delta = 1 \mid \bigwedge_{i\in[\Delta-1]} B_1 = 0\right) = \frac{\Delta-1}{\Delta+1}.$$

which, for $\Delta \geq 3$, is greater than $Pr(B_\Delta = 1)$ $(\approx 1/e)$.

2.2.2 The General Case

We now introduce the more general definition of λ-correlation and prove the corresponding extension of the CH-bounds.

The proof of Theorem 1 is based on the observation that if we can upper bound each term $E[X^k]$ of the Taylor expansion of $E[e^{tX}]$ by λ $E[\hat{X}^k]$ where $\hat{X}$ is the sum of independent random variables, and if $E[e^{t\hat{X}}] \leq B$, then $E[e^{tX}] \leq \lambda B$. This motivates the following definition.

Definition 2 *Let $X_1, X_2, \ldots, X_n$ be bounded random variables such that $X_i \in [a_i, b_i]$ and let $X = \sum_{i\in[n]} X_i$. The X_i's are λ-correlated if there exists a collection of independent twin random variables $\hat{X}_i \in [a_i, b_i]$ such that,*

- *$E[X] \leq E[\hat{X}]$, where $\hat{X} = \sum_{i\in[n]} \hat{X}_i$, and*
- *for all nonempty $I \subseteq [n]$ and positive integers $s_1, s_2, \ldots, s_{|I|}$,*

$$E[\prod_{i\in I} X_i^{s_i}] \leq \lambda \prod_{i\in I} E[\hat{X}_i^{s_i}].$$

If we apply this definition to the case of 0-1 random variables we see that to have λ–correlated 0-1 random variables, it suffices to find twin variables $\hat{X}_i$ such that $E[X] \leq E[\hat{X}]$ and, for all nonempty $I \subseteq [n]$,

$$Pr\left(\bigwedge_{i\in I} X_i = 1\right) \leq \lambda \prod_{i\in I} Pr(\hat{X}_i = 1)$$

which reduces to Definition 1 when $Pr(X_i = 1) = Pr(\hat{X}_i = 1)$. Notice that for this to hold, it is not necessary that $Pr(X_i = 1) = Pr(\hat{X}_i = 1)$; in typical applications we do not know the exact value of $Pr(X_i = 1)$ but only an approximation (usually an upper bound; we will see examples in later sections), but this is good enough to apply the CH-bounds extension.

Theorem 2 *Let X be the sum of λ–correlated random variables $X_1, X_2, \ldots, X_n$, where $X_i \in [a_i, b_i]$ and let $\hat{X}$ be the sum of the n twin variables $\hat{X}_i$. Then,*

$$Pr(X > (1+\epsilon)\, E[\hat{X}]) \leq \lambda\; H(\mu, \epsilon, \vec{a}, \vec{b}).$$

PROOF. Let $\mu = E[\hat{X}]$. As in the proof of Theorem 1, we start with the inequality

$$\begin{aligned} Pr(X > (1+\epsilon)\mu) &= Pr(e^{tX} > e^{t(1+\epsilon)\mu}) \\ &\leq \frac{E[e^{tX}]}{e^{t(1+\epsilon)\mu}}. \end{aligned}$$

By the hypotheses of boundedness and of λ–correlation it follows that

$$E[e^{tX}] = E\left[\sum_{k=0}^{\infty} \frac{t^k X^k}{k!}\right] = \sum_{k=0}^{\infty} \frac{t^k E[X^k]}{k!} \leq \lambda \sum_{k=0}^{\infty} \frac{t^k E[\hat{X}^k]}{k!} = \lambda\; E[e^{t\hat{X}}].$$

By the already discussed result of Hoeffding, when $\hat{X}$ is the sum of n independent bounded random variables $\hat{X}_i \in [a_i, b_i]$,

$$\min_{t>0} \frac{E[e^{t\hat{X}}]}{e^{t(1+\epsilon)\mu}} \leq H(n, \epsilon, \vec{a}, \vec{b}).$$

□

In this chapter, we will use the special case of Theorem 2 where $X_i \in [0, 1]$, $i \in [n]$. In this case, $G(\mu, \epsilon)$ is also an upper bound for the upper tail of X.

Corollary 1 *Let X be the sum of n λ-correlated random variables $X_i \in [0,1]$ and let $\hat{X}$ be the sum of the n twin variables $\hat{X}_i$. Then,*

$$Pr(X > (1+\epsilon)E[\hat{X}]) \leq \lambda\, G(\mu, \epsilon).$$

PROOF. Let $E[\hat{X}] = \mu$. When $\hat{X}$ is the sum of n independent random variables $\hat{X}_i \in [0,1]$ Hoeffding shows that, for $t \in (0, 1 - \mu/n)$ (cf. Theorem 1 of [50]),

$$Pr\left(\frac{\hat{X} - \mu}{n} \geq t\right) \leq \left(\frac{\mu}{\mu + nt}\right)^{\mu + nt} \left(1 + \frac{n\,t}{n - \mu - nt}\right)^{n - \mu - nt}.$$

By setting $\epsilon = nt/\mu$ and by applying the standard inequality $1 + x \leq e^x$ for $x = nt/(n - \mu - nt)$, we get

$$Pr(X \geq (1+\epsilon)\mu) \leq \left(\frac{e^{\epsilon}}{(1+\epsilon)^{(1+\epsilon)}}\right)^{\mu} = G(\mu, \epsilon).$$

□

2.3 The Edge Coloring Algorithm

We now present our randomized distributed edge coloring algorithm. The algorithm uses an idea of Karloff & Shmoys to reduce the problem of edge coloring general graphs to that of edge coloring a special class of bipartite graphs [57].

The Karloff & Shmoys algorithm was proposed for the PRAM model and is as follows. The input to the algorithm is a graph $G = (V, E)$ of maximum degree $\Delta(G) = \Delta$.

1. Compute a random partition of V into black and white vertices (all vertices flip a fair coin independently and in parallel). Let $G[B]$ be the subgraph induced by the black vertices, $G[W]$ be the subgraph induced by the white vertices, and $G[B, W]$ be the bipartite subgraph formed by the edges having endpoints of different colors.

2. Optimally edge color $G[B, W]$, *i.e.*, with $\Delta(G[B, W])$ colors, using the PRAM algorithm of Lev, Pippenger & Valiant [70].

3. Recurse on $G[B]$ and $G[W]$ using the same set of fresh new colors on both graphs.

The key fact is that as long as Δ is large enough, the maximum degree of the three subgraphs is at most $\Delta/2 + \Delta^{1/2+\epsilon} = \Delta/2 + o(\Delta)$ *w.h.p.*, for any fixed $\epsilon > 0$. (This can be proved by applying the standard CH-bounds.) Note that the *same* set of fresh new colors can be used on both $G[B]$ and $G[W]$ because $G[B, W]$ completely separates the two graphs. The recurrence for the number of colors used is

$$\begin{aligned} C(\Delta) &\leq \left(\frac{\Delta}{2} + \Delta^{1/2+\epsilon}\right) + C\left(\frac{\Delta}{2} + \Delta^{1/2+\epsilon}\right) \\ &\leq \Delta + o(\Delta) \end{aligned}$$

and holds *w.h.p.* The "high probability" statement holds as long as $\Delta = \Omega(\log^{1+\delta} n)$, for any fixed $\delta > 0$. In this case, the failure probability given by the CH-bounds is the inverse of a superpolynomial function. For the case when the degree goes below the $\Omega(\log^{1+\delta})$ threshold, Karloff & Shmoys give a PRAM algorithm to $(\Delta+1)$–edge color graphs of this low maximum degree.

The non–distributed parts of the Karloff & Shmoys scheme are the subroutine which colors the bipartite graph optimally (it can be shown that this is impossible in the distributed model of computation [92]), and the handling of the case $\Delta \leq \log^{1+\delta} n$. The latter can be handled by Luby's randomized distributed $(\Delta + 1)$–*vertex* coloring algorithm [76]. Luby's algorithm is simulated on the line graph of the given graph G thus giving a $2\Delta(G) - 1$ edge coloring. (When $\Delta(G) = O(polylog(n))$, we can compute a $2\Delta(G) - 1$ coloring *deterministically* in $O(polylog(n))$ time via an algorithm based on the idea of removing maximal matchings. We prefer to use Luby's algorithm here for conciseness.) Hence, if we could get a good distributed algorithm for bipartite edge coloring then we could use the Karloff & Shmoys scheme to get a good coloring of any graph. We now show how this can be achieved.

2.3.1 Distributed Edge Coloring of Bipartite Graphs

We now present a simple Monte Carlo algorithm for edge coloring the special type of bipartite graphs generated by the Karloff & Shmoys scheme.

Given a bipartite graph $G = (A, B, E)$ we assume that each vertex knows whether it belongs to A or B. This is an important assumption because such information cannot be computed fast distributively [92], but it holds for the bipartite graphs generated by the Karloff & Shmoys scheme. From now on, we will refer to vertices in A as the *top* vertices and to the vertices in B as the *bottom* vertices.

The algorithm takes $O(\log n)$ time and colors a bipartite graph G of maximum degree Δ with at most $1.6\Delta + O(\log^{2+\delta} n)$ colors *w.h.p.*, for any fixed $\delta > 0$ (the failure probability will depend on δ). In the algorithm we use a variable Δ_C that is initialized to $\Delta(G)$. During any iteration of the algorithm, Δ_C is meant to be an upper bound on the degree of the current graph; we will prove later that this holds *w.h.p.* Recall that $\delta(u)$ denotes the set of edges incident at vertex u, and let THRESHOLD $= \log^{2+\delta} n$.

The algorithm is given two parameters $\epsilon, \delta > 0$, and is as follows:

1. PART I: Repeat until $\Delta_C <$ THRESHOLD:

   Let G_C be the current graph. Pick a set χ of Δ_C fresh new colors.

   - *(Random proposal of bottom vertices)* In parallel and independently of the other vertices in B, each vertex $v \in B$ assigns a temporary color to each edge in $\delta(v)$ with uniform probability *without replacement, i.e.* edge e_1 is assigned color $\alpha \in \chi$ with probability $1/\Delta_C$, e_2 is assigned $\beta \in \chi - \{\alpha\}$ with probability $1/(\Delta_C - 1)$ and so on.

   - *(Lottery of top vertices)* (**Comment:** The coloring so far is consistent around any vertex $v \in B$ but can be inconsistent around a vertex $u \in A$.) For each $u \in A$, let $C_u(\alpha)$ be the set of edges in $\delta(u)$ with temporary color α. Each vertex $u \in A$ selects a *winner* uniformly at random in $C_u(\alpha)$, for each nonempty $C_u(\alpha)$. All other edges, the *losers*, are decolored and assigned "color" $\perp$.

- Set $\Delta_C := \Delta_C(1+\epsilon)/e$, where $e = 2.71\ldots$ is the base of natural logarithms. $G_\perp$, the subgraph of G_C induced by the losers (*i.e.*, by the $\perp$-edges), becomes the new current graph.

2. PART II: Let G_r be the remaining graph. Edge color G_r with $2\Delta(G_r) - 1 < 2 \cdot$ THRESHOLD colors by executing Luby's vertex coloring algorithm on the line graph of G_r [76].

The **key claim** is that in every iteration of part (I) above, the maximum degree of the graph shrinks by a factor of at least $(1+\epsilon)/e$ *w.h.p.*, as long as $\Delta \geq$ THRESHOLD. That is,

$$\Delta(G_\perp) \leq (1+\epsilon)\,\frac{\Delta(G_C)}{e}$$

with high probability, for any fixed $\epsilon > 0$. The condition $\Delta_C \geq$ THRESHOLD ensures that the failure probability given by the extension of the CH-bounds is the inverse of a superpolynomial function. The reason for setting THRESHOLD $= \log^{2+\delta} n$ will be apparent from the probabilistic analysis.

Once the key claim is established, we can bound both the total number of colors used and the running time of the algorithm. To bound the number of colors used, observe that if the degree of the graph shrinks at every iteration by at least an $(1+\epsilon)/e$ factor *w.h.p.* then the maximum degree of G_r is at most $\log^{2+\delta} n$ *w.h.p.* Hence, *w.h.p.*, the number of colors used is at most

$$C(\Delta) \leq \Delta + \frac{\Delta}{e}(1+\epsilon) + \ldots + \frac{\Delta}{e^k}(1+\epsilon)^k + 2\log^{2+\delta} n,$$

where k is the smallest integer such that $\Delta(1+\epsilon)^k/e^k \leq \log^{2+\delta} n$. The total number of colors used is at most (here, ϵ' depends on ϵ and can be made arbitrarily small)

$$C(\Delta) \;\leq\; \left(\frac{e}{e-1} + \epsilon'\right)\Delta + \left(2 - \frac{e}{e-1} - \epsilon'\right)\log^{2+\delta} n$$

$$< \; 1.585\,\Delta + 0.4\,\log^{2+\delta} n$$
$$< \; 1.59\,\Delta$$

when $\Delta > 80\,\log^{2+\delta} n$. The running time of the algorithm is $O(\log n)$ because part (I) takes $O(\log \Delta)$ time (by the key claim), and part (II), *i.e.*, Luby's algorithm, takes $O(\log n)$ time.

If $\Delta > 80\,\log^{2+\delta} n$ and we use our distributed subroutine for bipartite graphs in the Karloff & Shmoys algorithm the total number of colors is given by the recurrence

$$\begin{aligned} TC(\Delta) &\leq C\left(\frac{\Delta}{2} + \Delta^{1/2+\epsilon}\right) + TC\left(\frac{\Delta}{2} + \Delta^{1/2+\epsilon}\right) \\ &< 1.59\,\Delta + o(\Delta) \\ &< 1.6\,\Delta\,. \end{aligned}$$

If $\Delta \leq 80\,\log^{2+\delta} n$, we apply Luby's algorithm directly to get an edge coloring with $2\,\Delta - 1 \leq 160\,\log^{2+\delta} n$ colors. Hence, the total number of colors to color *any* graph is at most $1.6\,\Delta + O(\log^{2+\delta} n)$, for any $\delta > 0$.

The rest of this chapter is devoted to establishing the key claim.

2.4 Probabilistic Analysis

This section is devoted to proving the key claim of the preceding section, namely that given a graph G and Δ such that $\Delta \geq \Delta(G)$ and $\Delta \geq$ THRESHOLD then, after one iteration of Part (I) of the bipartite algorithm, the maximum degree of the new graph, $\Delta(G_\perp)$, is at most $(1+\epsilon)\Delta/e$ *w.h.p.*, for any fixed $\epsilon > 0$. It turns out that the analysis is considerably easier for the top vertices than for the bottom vertices. We begin with the easy part.

2.4.1 Analysis of Top Vertices

Let u be a generic top vertex with neighbors v_i, $i \in [d_u]$, and incident edges $e_i = (u, v_i)$. Recall that a *loser* is an edge which, after having got a tentative color in the random proposal, lost the lottery and got decolored. So, the new degree of u is given by the number of losers incident at it.

From the point of view of a top vertex, the random proposal and the lottery are equivalent to the following random experiment. For each edge e_i incident at u we introduce a ball i, and for each color k we introduce a bin k; the assignment of a tentative color to an edge by the algorithm is equivalent to throwing each ball into one of the Δ bins independently and uniformly at random, since the bottom vertices assign tentative colors with uniform probability and independently of the other bottom vertices. Recalling that we have at most Δ balls and exactly Δ bins:

$$\begin{aligned} \sharp losers &= \sharp balls - \sharp winners \\ &\leq \sharp bins - \sharp nonempty\ bins \\ &= \sharp empty\ bins. \end{aligned}$$

Let X be a random variable denoting the number of losers. To estimate X and its tail distribution we will study the random variable $B = \sharp empty\ bins$. For this purpose, we introduce Δ many indicator random variables

$$B_k = \begin{cases} 1 & \text{bin } k \text{ is empty} \\ 0 & \text{otherwise} \end{cases}$$

and hence, $B = \sum_{k\in[\Delta]} B_k$. Note that $X \leq B$ always. The variable B was studied in Section 2.2.1 where it was shown that $E[B] \leq \Delta/e$ and that the B_k's are 1-correlated, which implies that $Pr(B > (1+\epsilon)\Delta/e) \leq G(\Delta/e, \epsilon)$. Since $E[X] \leq E[B]$ and $Pr(X > (1+\epsilon)\Delta/e) \leq Pr(B > (1+\epsilon)\Delta/e)$, we get

Theorem 3 *Let u be any top vertex. For any iteration of Part (I) of our algorithm, suppose the degree of u was at most Δ_C before that iteration, and let X be the random variable denoting the number of losers incident at u after the iteration, i.e., u's new degree. Then, $E[X] \leq \Delta_C/e$ and*

$$Pr(X > (1+\epsilon)\Delta_C/e) \leq G(\Delta_C/e, \epsilon).$$

2.4.2 Analysis of Bottom Vertices

In this section we analyze what happens to the new degree of a generic bottom vertex v_B. This case is considerably harder to handle than the previous one, because of the way in which the random variables describing the process are correlated. For a top vertex, the dependency among the variables was playing for us; given that some edges incident at a top vertex are losers, the probability of having another loser decreases. For a bottom vertex the situation is reversed: having some edges losing the lottery might even make the probability of having another loser increase. The problem can be seen in the following situation. Let $x_1 = v_B$ and x_2 be bottom vertices, and y_1 and y_2 be top vertices which induce a four-cycle, *i.e.*, there is an edge $e_{i,j} = (x_i, y_j)$ for $i, j = 1, 2$. Suppose we are given that $e_{1,1}$ got tentative color α and lost the lottery, and that $e_{1,2}$ got tentative color β; we will argue intuitively that given this, the probability of $e_{1,2}$ losing the lottery has increased. Since $e_{1,1}$ lost, the probability of $e_{2,1}$ getting tentative color α increases. which implies that the probability of $e_{2,2}$ getting tentative color β also increases, and this increases the probability of $e_{1,2}$ losing the lottery.

For the sake of the analysis we modify the algorithm as follows: instead of performing all random proposals in parallel, suppose that the bottom vertices perform their random proposals sequentially one after the other and independently, with v_B being the last. This does not modify the probability distributions because the choices are still done independently. We want to focus our attention on the last vertex v_B performing the random proposal. We will use the fact that when v_B performs its random proposal, all edges not incident at v_B already have a tentative color. By symmetry, any upper bound on the probabilities we can find for v_B will hold for all bottom vertices.

Let e_i, for $i \in [d_{v_B}]$, denote an edge incident at the bottom vertex v_B, with the other endpoint of e_i being u_i. We introduce the indicator random variables

$$X_i = \begin{cases} 1 & e_i \text{ loses the lottery} \\ 0 & \text{otherwise} \end{cases}$$

and want to study the expectation and tail probability distribution of $X = \sum_{i \in [d_{v_B}]} X_i$. Computing the expectation is easy.

Lemma 1 $E[X] \leq \Delta/e$.

PROOF. It is sufficient to show that $Pr(X_i = 1) \leq 1/e$, for all $i \in [d_{v_B}]$. From the analysis of the top vertices, we know that the expected number of losers incident at u_i is at most Δ/e and hence that $\sum_{e_j \in \delta(u_i)} Pr(e_j \text{ loses}) \leq \Delta/e$. By symmetry, $Pr(e_j \text{ loses}) \leq 1/e$ for all $j \in \delta(u_i)$, and hence $Pr(X_i = 1) \leq 1/e$. □

We now study the tail probability distribution of X. Our goal is to show that $X \leq (1+\epsilon)\Delta/e$ *w.h.p.*, for any fixed $\epsilon > 0$. Establishing this claim will take several lemmas.

For technical reasons that will be clear later, we subdivide the edges incident at v_B into $\sqrt{\Delta}$ groups, each of at most $\sqrt{\Delta}$ edges. Let A be any such group and define $X_A = \sum_{e_i \in A} X_i$; we would like to show that X_A is the sum of λ–correlated random variables, so that

$$\forall \epsilon > 0, \quad Pr(X_A > (1+\epsilon)\sqrt{\Delta}/e) \leq \lambda\, G(\sqrt{\Delta}/e, \epsilon).$$

Rather than proving this claim, we will prove a weaker one namely that X_A is the sum of λ–correlated random variables *with high probability!* More precisely, we will define an event $\mathcal{A}$ such that $\mathcal{A}$ happens *w.h.p.*, and such that, given that $\mathcal{A}$ occurs, then X_A is the sum of λ–correlated random variables. That is,

$$Pr(X_A > (1+\epsilon)\sqrt{\Delta}/e \mid \mathcal{A}) \leq \lambda\, G(\sqrt{\Delta}/e, \epsilon).$$

Showing this weaker claim is sufficient because

$$Pr(X > (1+\epsilon)\Delta/e) \quad \leq \quad Pr(\exists A\; X_A > (1+\epsilon)\sqrt{\Delta}/e)$$

$$
\begin{aligned}
&= Pr(\exists A\ X_A > (1+\epsilon)\sqrt{\Delta}/e \mid \mathcal{A})\ Pr(\mathcal{A})\ + \\
&\quad\ Pr(\exists A\ X_A > (1+\epsilon)\sqrt{\Delta}/e \mid \mathcal{A}^c)\ Pr(\mathcal{A}^c) \\
&\leq Pr(\exists A\ X_A > (1+\epsilon)\sqrt{\Delta}/e \mid \mathcal{A})\ +\ Pr(\mathcal{A}^c) \\
&\leq \sqrt{\Delta}\ Pr(X_A > (1+\epsilon)\sqrt{\Delta}/e \mid \mathcal{A})\ +\ Pr(\mathcal{A}^c) \\
&\leq \sqrt{\Delta}\ \lambda\ G(\sqrt{\Delta}/e, \epsilon)\ +\ Pr(\mathcal{A}^c).
\end{aligned}
$$

Another point in the analysis is that λ will not be a constant term but an exponentially growing function. However we will be able to show the rate of growth of λ is much slower than that of $1/G(\sqrt{\Delta}/e, \epsilon)$. In other words, we will show that $\lambda\ G(\sqrt{\Delta}/e, \epsilon)$ goes to zero superpolynomially fast.

We now turn to the task of showing that X_A is the sum of λ-correlated random variables *w.h.p.* Recall from the definition that the X_i's, $e_i \in A$, are λ-correlated if, for all nonempty $I \subseteq A$,

$$
Pr\left(\bigwedge_{e_i \in I} X_i = 1\right) \leq \lambda \prod_{i \in I} Pr(X_i = 1).
$$

Consider then a generic subset $I \subseteq A$ of edges incident at the bottom vertex v_B, and let us see how to compute $Pr\left(\bigwedge_{e_i \in I} X_i = 1\right)$. Recall that we are analyzing the situation where v_B is the last vertex to perform its random proposal. This means that prior to the assignment of a tentative color to edge $e_i = (v_B, u_i)$, all other edges incident at u_i already have their tentative color. Using the balls-and-bins language, we can say that prior to throwing ball e_i at random into one of the bins, all balls coming from the other neighbors of u_i have been thrown. We will think of e_i as a red ball and of the other edges at u_i as blue balls. Once the red ball is thrown in, say, bin $k \in [\Delta]$, a winner is selected uniformly at random among all (*i.e.*, red *and* blue) balls in bin k. All other balls, the losers, are discarded. Notice that the probability of discarding the red ball is itself a random variable which depends on the particular placement of the blue balls prior to throwing the red ball.

Given any placement of blue balls at u_i, we construct a vector of probabilities

C_i as follows. Let a_{ik} denote the number of blue balls in bin $k \in [\Delta]$ of vertex u_i, and let $p_{ik} = a_{ik}/(1 + a_{ik})$ denote the probability that the red ball loses the lottery given that it was thrown in bin k (equivalently, p_{ik} is the probability that edge i loses, given that it got tentative color k). For each neighbor u_i of our bottom vertex v_B, we construct the corresponding vector $C_i = (p_{i1}, p_{i2}, \ldots, p_{i\Delta})$; recall that all this is with respect to the given placement of blue balls at each u_i. Given a set $I = \{e_{j_1}, e_{j_2}, \ldots, e_{j_k}\}$ of edges incident at v_B, we construct the matrix M_I whose i-th column is the vector C_{j_i}. The next lemma explains why this matrix is relevant to us. Let $[\ell]_k \doteq \ell(\ell-1)\cdots(\ell-k+1)$.

Lemma 2 *Let $I = \{e_{j_1}, e_{j_2}, \ldots, e_{j_k}\} \subseteq A$, and fix some placement $\mathcal{PB}$ of blue balls at each u_{j_ℓ}. Then,*

$$Pr\left(\bigwedge_{e_i \in I} X_i = 1 \mid \mathcal{PB}\right) = \frac{perm(M_I)}{[\Delta]_k}.$$

PROOF. The random proposal of v_B, when restricted to I, is equivalent to choosing an one-to-one function $\pi : I \to [\Delta]$ uniformly at random among the set $\mathcal{P}$ of all such functions. Recall that the entry $M_{\ell i}$ of M_I is the conditional probability that edge e_{j_i} loses, given that it has temporary color ℓ. Hence,

$$\begin{aligned}
Pr(\bigwedge_{e_i \in I} X_i = 1 \mid \mathcal{PB}) &= \sum_{\pi \in \mathcal{P}} Pr(\bigwedge_{e_i \in I} X_i = 1 \mid \pi \textit{ is selected}, \mathcal{PB})\ Pr(\pi \textit{ is selected}) \\
&= \sum_{\pi \in \mathcal{P}} \left(M_{\pi(1),1}\ M_{\pi(2),2}\ \cdots\ M_{\pi(k),k}\right)\left(\prod_{j=0}^{k-1} \frac{1}{\Delta - j}\right) \\
&= \frac{perm(M_I)}{[\Delta]_k}.
\end{aligned}$$

□

We now want to find a good upper bound of $perm(M_I)$. The following lemma gives a simple upper bound that is sufficient for our purposes. Given a matrix M, let S_i denote the sum of the elements of the i-th column of M.

Lemma 3 *Let M be a matrix with c columns and r rows, $c \leq r$, and with non-negative entries. Then,* $\mathrm{perm}(M) \leq \prod_{i \in [c]} S_i$.

PROOF. Let $\mathcal{P} = \{\pi \mid \pi : [c] \to [r],\ \pi \text{ is one-to-one}\}$. Then,

$$\begin{aligned} perm(M) &= \sum_{\pi \in \mathcal{P}} M_{\pi(1),1}\, M_{\pi(2),2} \cdots M_{\pi(c),c} \\ &\leq (M_{1,1} + \cdots + M_{r,1})(M_{1,2} + \cdots + M_{r,2}) \ldots (M_{1,c} + \cdots + M_{r,c}) \\ &= \prod_{i \in [c]} S_i. \end{aligned}$$

□

The next lemma relates the value of S_i to that of $1/e \geq Pr(e_i \text{ loses})$, $e_i \in \delta(v)$. It is an application of the most general definition of λ–correlation. Before the proof of the lemma, we establish

Proposition 1 *If $0 \leq p \leq 1$, $q = 1 - p$ and m is a positive integer, then*

$$\sum_{r=1}^{m} \binom{m}{r} p^r q^{m-r} \frac{r}{r+1} = 1 - \frac{(1 - q^{m+1})}{p(m+1)}.$$

PROOF. Let

$$\begin{aligned} f(p) &= \sum_{r=1}^{m} \binom{m}{r} p^r q^{m-r} \frac{r}{r+1} \\ &= 1 - q^m - \sum_{r=1}^{m} \binom{m}{r} p^r q^{m-r} \frac{1}{r+1}. \end{aligned}$$

Integrating, with respect to x, both sides of the binomial expansion

$$(x+q)^m = \sum_{r=0}^{m} \binom{m}{r} x^r q^{m-r}$$

between 0 and p, we get

$$\frac{1 - q^{m+1}}{m+1} = p\,(1 - f(p)),$$

from which the proposition follows. □

We now return to our scenario where v_B is the last bottom vertex to pick tentative colors for its edges, with its edges having been split into groups. Recall that we are focusing on a particular group A, and that we want a good upper bound on $Pr\left(\bigwedge_{i=1,\dots,k} X_{j_i} = 1\right)$, for any nonempty $I = \{e_{j_1}, e_{j_2}, \dots, e_{j_k}\} \subseteq A$. Given any placement $\mathcal{PB}$ of blue balls at each u_i, Lemma 2 shows that

$$Pr\left(\bigwedge_{i=1,\dots,k} X_{j_i} = 1 \mid \mathcal{PB}\right) = perm(M_I)/[\Delta]_k,$$

and Lemma 3 gives the upper bound

$$perm(M_I) \leq \prod_{i=1}^{k} S_{j_i}.$$

So, a good upper bound on each S_{j_i} will hopefully translate into a good upper bound for $Pr\left(\bigwedge_{i=1,\dots,k} X_{j_i} = 1\right)$. The next lemma says that $S_{j_i} \leq \Delta(1+\epsilon_1)/e$ *w.h.p.* for any fixed $\epsilon_1 > 0$. Recall that we are analyzing the situation where v_B is to give tentative colors for its edges and where all other edges have already been assigned tentative colors. If we focus on a particular edge $e_{j_i} = (v_B, u_{j_i})$, the situation is equivalent to throwing a red ball e_{j_i} uniformly at random into Δ bins where each bin has some number of blue balls in it (possibly zero); the number of blue balls in each bin is itself a random variable.

Lemma 4 *Let $e_{j_i} = (v, u_{j_i})$ be any edge in $I = \{e_{j_1}, e_{j_2}, \dots, e_{j_k}\} \subseteq A$, and S_{j_i} be the sum of the elements of the i-th column of M_I. Then, $E[S_{j_i}] \leq \Delta/e = \mu$ and*

$$\forall \epsilon_1 > 0, \quad Pr(S_{j_i} > (1+\epsilon_1)\mu) \leq G(\mu, \epsilon_1).$$

PROOF. For vertex u_{j_i}, let Z_ℓ be the random variable denoting the number of blue balls in bin ℓ and $Y_\ell = Z_\ell/(Z_\ell + 1)$ be the random variable denoting the probability that the red ball loses the lottery given that it ends in bin ℓ. Then, $S_{j_i} = Y \doteq \sum_{\ell=1}^{\Delta} Y_\ell$. Note that the Y_ℓ's are bounded random variables with values in $[0,1]$. We will show that $E[Y] \leq \Delta/e$ and that the Y_ℓ's are 1–correlated (under the most general definition of λ–correlation), which will give our claim.

Firstly, we may assume that we have $d = \Delta - 1$ blue balls (*i.e.*, that the degree of u_i is Δ): $Pr(Y \geq (1+\epsilon_1)\Delta/e)$ is maximized at $d = \Delta - 1$, as d varies from 1 to $\Delta - 1$.

(To see this, assume $d = \Delta - 1 - \ell < \Delta - 1$. Add ℓ *yellow* balls to the blue balls, and run two experiments. In one experiment throw the blue and red balls and compute the probability that the red ball loses the lottery. In the other experiment, throw blue, yellow and red balls and again compute the probability that the red ball loses. In both experiments, let us look at the bin where the red ball fell. The probability that the red ball loses is $b/(b+1)$ for the first experiment, and $(b+y)/(b+y+1)$ for the second, where b and y are, respectively, the number of blue and yellow balls in the bin. Since $y \geq 0$, $b/(b+1) \leq (b+y)/(b+y+1)$. Thus, if $Y(d)$ indicates the variable Y when u_{j_i} has degree d, then $Y(d) \leq Y(\Delta)$ for all $d \in [\Delta]$.)

First, we will show that, for all $r \in \Delta$, $E[Y_r] \leq 1/e$ and then we will show that, for any set of $\ell \geq 1$ indices $L \subseteq [\Delta]$ and strictly positive integers s_r,

$$E[\prod_{r \in L} Y_r^{s_r}] \leq 1/e^{\ell}. \tag{2.3}$$

Given this we can apply Corollary 1 by introducing n independent twin 0-1 random variables $\hat{Y}_r$ such that $E[\hat{Y}_r] = Pr(\hat{Y}_r = 1) = 1/e$. Since the $\hat{Y}_r$'s are binary, inequality (2.3) is the same as

$$E[\prod_{r \in L} Y_r^{s_r}] \leq \prod_{r \in L} E[\hat{Y}_r] = \prod_{r \in L} E[\hat{Y}_r^{s_r}],$$

which is to say that the Y_r's are 1–correlated. Noting that $0 \leq Y_r \leq 1$, it suffices to show that

$$E[\prod_{r \in L} Y_r] \leq 1/e^{\ell}. \tag{2.4}$$

We assume w.l.o.g. that $L = [\ell]$, and prove inequality (2.4) by induction on $\ell \geq 1$; when $\ell = 1$,

$$\begin{aligned} E[Y_1] &= \sum_{r=0}^{\Delta-1} \binom{\Delta - 1}{r} \left(\frac{1}{\Delta}\right)^r \left(1 - \frac{1}{\Delta}\right)^{\Delta-1-r} \frac{r}{r+1} \\ &= (1 - 1/\Delta)^{\Delta} \leq 1/e, \end{aligned}$$

where the second equality follows from Proposition 1. Notice that for all $r \in [\Delta]$, $E[Y_r] = E[Y_1] \leq 1/e$. When $\ell > 1$, the law of conditional probabilities gives

$$E[\prod_{r\in[\ell]} Y_r] = E[Y_1 Y_2 \cdots Y_{\ell-1}\ E[Y_\ell \mid Y_1, Y_2, \ldots, Y_{\ell-1}]] \tag{2.5}$$

Suppose we show that for all *non-zero* $c_r \in [0,1]$ with $r \in [\ell-1]$,

$$E[Y_\ell \mid \bigwedge_{r=1}^{\ell-1} Y_r = c_r] \leq \frac{1}{e}; \tag{2.6}$$

then, since the product $Y_1 Y_2 \cdots Y_{\ell-1}$ in equation (2.5) is zero when any c_r is zero, we see by induction on ℓ that

$$\begin{aligned} E[\prod_{r=1}^{\ell} Y_r] &= E[Y_1 Y_2 \cdots Y_{\ell-1}\ E[Y_\ell \mid Y_1, Y_2, \ldots, Y_{\ell-1}]] \\ &\leq \frac{1}{e} E[\prod_{r=1}^{\ell-1} Y_r] \\ &\leq \frac{1}{e^\ell}. \end{aligned}$$

Hence, the claim follows if we can show that inequality (2.6) holds.

If a_r denotes the number of blue balls that fell into bin r, then $c_r = a_r/(a_r+1)$; each a_r is strictly positive since $c_r > 0$. Let $a = \sum_{r=1}^{\ell-1} a_r \geq \ell - 1$, $p = 1/(\Delta - \ell + 1)$, and $q = 1 - p$. Then

$$\begin{aligned} E[Y_\ell | \bigwedge_{r=1}^{\ell-1} Y_r = c_r] &= E[Y_\ell | \bigwedge_{r=1}^{\ell-1} Z_r = a_r] \\ &= \sum_{r=1}^{\Delta-1-a} t(r,a), \end{aligned}$$

where

$$t(r,a) \doteq \binom{\Delta - 1 - a}{r} p^r q^{\Delta-1-a-r} \frac{r}{r+1}.$$

It is easy to check that $t(r,a) \geq t(r,a+1)$. As a consequence, the maximum value of $E[Y_\ell | \bigwedge_{r=1}^{\ell-1} Y_r = c_r]$ is attained at $a = \ell - 1$, in which case we have

$$\begin{aligned}
\sum_{r=1}^{\Delta-1-a} t(r,a) &= \sum_{r=1}^{\Delta-\ell} t(r,\ell-1) \\
&= \sum_{r=1}^{\Delta-\ell} \binom{\Delta-\ell}{r} p^r q^{\Delta-\ell-r} \frac{r}{r+1} \\
&= q^{\Delta-\ell+1} \leq 1/e,
\end{aligned}$$

by Proposition 1. □

Putting all the pieces together (recall that $k = |I|$):

$$\begin{aligned}
Pr(\bigwedge_{e_i \in I} X_i = 1) &= \frac{perm(M_I)}{[\Delta]_k} \\
&\leq \frac{\prod_{i \in I} S_i}{[\Delta]_k} \\
&\leq (1+\epsilon_1)^k \frac{\Delta^k}{e^k} \frac{1}{[\Delta]_k} \text{ (w.h.p.)} \\
&= (1+\epsilon_1)^k \frac{\Delta^k}{[\Delta]_k} \frac{1}{e^k},
\end{aligned}$$

for any $\epsilon_1 > 0$. The first line follows from Lemma 2, the second from Lemma 3, and the third from Lemma 4 and holds *w.h.p.* The last line is just a re-writing of the third one. Given $\epsilon_1 > 0$, let $\mathcal{A}_{\epsilon_1}$ be the event that, for all $e_i \in A$, $S_i \leq (1+\epsilon_1)\Delta/e$. Lemma 4 says that $\mathcal{A}_{\epsilon_1}$ happens *w.h.p.*, for any $\epsilon_1 > 0$. The last line of the above chain of inequalities means that, given that $\mathcal{A}_{\epsilon_1}$ occurred, the X_i's are λ-correlated with $\lambda = (1+\epsilon_1)^k \Delta^k / [\Delta]_k$. (Alternatively, we can say that the X_i's are λ–correlated *w.h.p.*) We now want an estimate of λ. The next lemma explains why we subdivided the edges incident at v_B into groups.

Lemma 5 *For positive integers t and k, $t^k/[t]_k \leq e^{k^2/t}$, if $k \leq t/2$.*

PROOF. We first note that $\ln(1-x) \geq -2x$, for $0 \leq x \leq 1/2$. This is true, since if we define $g(x) \doteq \ln(1-x) + 2x$, then $g(0) = 0$ and $g'(x) = \frac{1-2x}{1-x}$, which is non-negative for $0 \leq x \leq 1/2$. Now,

$$\begin{aligned}
\frac{[t]_k}{t^k} &= \prod_{i=1}^{k-1} \frac{(t-i)}{t} \\
&= \exp\left(\sum_{i=1}^{k-1} \ln\left(1 - \frac{i}{t}\right)\right) \\
&\geq \exp\left(-\sum_{i=1}^{k-1} \frac{2i}{t}\right) \qquad \text{(since } k \leq t/2\text{)} \\
&= \exp\left(-\frac{(k-1)k}{t}\right) \\
&\geq e^{-\frac{k^2}{t}}.
\end{aligned}$$

□

For $t = \Delta$ and $k = |I| \leq \sqrt{\Delta}$ this lemma implies that $\Delta^k/[\Delta]_k \leq e$ and hence, given $\mathcal{A}_{\epsilon_1}$, the X_i are λ-correlated with $\lambda = (1+\epsilon_1)^k e \leq e^{1+\epsilon_1\sqrt{\Delta}}$, via the standard approximation $1 + x \leq e^x$. Though λ goes to infinity superpolynomially fast as Δ goes to infinity, we can still upper bound $Pr(X_A > (1+\epsilon)\Delta/e)$ by a term that goes to zero superpolynomially fast, as follows. (Here, $\mu = \sqrt{\Delta}/e$.)

$$\begin{aligned}
Pr(X_A > (1+\epsilon)\mu) &= Pr(X_A > (1+\epsilon)\mu \mid \mathcal{A}_{\epsilon_1})\, Pr(\mathcal{A}_{\epsilon_1}) \; + \\
&\quad\; Pr(X_A > (1+\epsilon)\mu \mid \mathcal{A}^c_{\epsilon_1})\, Pr(\mathcal{A}^c_{\epsilon_1}) \\
&\leq Pr(X_A > (1+\epsilon)\mu \mid \mathcal{A}_{\epsilon_1}) \; + \; Pr(\mathcal{A}^c_{\epsilon_1}) \\
&\leq \lambda\, G(\mu, \epsilon) \; + \; Pr(\exists A\; X_A > (1+\epsilon_1)\mu) \\
&\leq e^{1+\epsilon_1\sqrt{\Delta}}\, G(\mu, \epsilon) \; + \; \sqrt{\Delta}\, G(\mu, \epsilon_1) \\
&\leq e^{1+\epsilon_1\sqrt{\Delta}}\, e^{-\epsilon^2\sqrt{\Delta}/3} \; + \; \sqrt{\Delta}\, G(\mu, \epsilon_1) \\
&\leq e^{1+(\epsilon_1-\epsilon^2/3)\sqrt{\Delta}} \; + \; \sqrt{\Delta}\, G(\mu, \epsilon_1).
\end{aligned}$$

Since ϵ_1 can be chosen arbitrarily small, we can make the above term go to zero exponentially fast by choosing $\epsilon_1 > 0$ such that $\epsilon_1 - \epsilon^2/3 < 0$.

Remark. We can now see why the parameter THRESHOLD must be $\Omega(\log^{2+\delta} n)$: the above failure probability goes to zero superpolynomially fast as long as $\Delta = \Omega(\log^{2+\delta} n)$, for any fixed $\delta > 0$.

Hence, we have proven that $X_A < (1+\epsilon)\sqrt{\Delta}/e$ *w.h.p.*, for any fixed $\epsilon > 0$. In turn, this implies that the new degree of v_B after one iteration of the algorithm is at most $(1+\epsilon)\Delta/e$ *w.h.p.* To see this,

$$\begin{aligned} Pr(X > (1+\epsilon)\Delta/e) &\leq Pr(\exists A\ X_A > (1+\epsilon)\sqrt{\Delta}/e) \\ &\leq \sqrt{\Delta}\ Pr(X_A > (1+\epsilon)\sqrt{\Delta}/e) \\ &\leq \sqrt{\Delta}\ \left(e^{1+(\epsilon_1-\epsilon^2/3)\sqrt{\Delta}} + \sqrt{\Delta}G(\mu,\epsilon_1)\right). \end{aligned}$$

This inequality holds for any ϵ and ϵ_1. Hence the new degree of *all* the vertices is at most $(1+\epsilon)\Delta/e$ *w.h.p.* and we get

Theorem 4 *For any iteration of Part (I) of our algorithm, the maximum degree of the graph after that iteration is at most $(1+\epsilon)\Delta_C/e$ w.h.p. for any fixed $\epsilon > 0$, if its maximum degree before the iteration was at most Δ_C.*

In conjunction with Theorem 3, Theorem 4 proves the key claim, and hence, as shown in Section 2.3.1, proves the stated bounds on the performance of our algorithm. Thus, the probabilistic tools developed in Sections 2.2.1 and 2.2.2 have been crucial in the analysis of our algorithm. As mentioned earlier, these results served partly as the motivation for the work of the next chapter, and the work of Section 3.1 generalizes the results of Sections 2.2.1 and 2.2.2.

Chapter 3

Chernoff–Hoeffding Bounds for Applications with Limited Independence

As pointed out in Chapter 2, the most fundamental tools used in bounding the tail probabilities, *i.e.*, the probabilities of deviation from the mean, of the sums of bounded and independent random variables, are based on techniques of Chernoff [27] and Hoeffding [50]. They are frequently used in the design and analysis of randomized algorithms, derandomization, and in the probabilistic method. Motivated in part by the work of Chapter 2, we present, in this chapter, a simple method which generalizes, somewhat, the classical method for proving the Chernoff-Hoeffding bounds in the case of bounded random variables confined to the interval $[0, 1]$. More importantly, this approach requires only *limited independence* among the random variables, and thereby imports a variety of standard results to the case of limited independence for free. This and related bounds lead to a variety of applications ranging from improved bounds for tail probability distributions to new algorithmic results. This chapter describes joint work with J. P. Schmidt and A. Siegel [108].

The "limited independence" result implies that sources of randomness that are weaker than the standard model of unbiased and independent bits, are sufficient for any algorithm whose analysis uses the Chernoff–Hoeffding bounds. It also provides a better understanding of the proof techniques behind these bounds, and gives improved bounds for various tail probability distributions. Via standard techniques, it leads to algorithms requiring fewer random bits for such classical problems as random sampling. The formulation also leads to approximation algorithms with better approximation guarantees for certain problems.

Given n random variables $X_1, X_2, \ldots, X_n$, suppose we want to upper bound the "upper tail" probability $Pr(X \geq a)$, where $X \doteq \sum_{i=1}^{n} X_i$, $\mu \doteq E[X]$, $a = \mu(1+\delta)$ and $\delta > 0$. The classical idea behind the Chernoff–Hoeffding bounds (see, for instance, Chernoff [27], Hoeffding [50], Raghavan [101] and Alon, Spencer, & Erdős [8]) is as follows; Hoeffding states that this was apparently first used by S. N. Bernstein. For any fixed $t > 0$, $Pr(X \geq a) = Pr(e^{tX} \geq e^{at}) \leq \frac{E[e^{tX}]}{e^{at}}$, by Markov's inequality. Computing an upper bound $u(t)$ on $E[e^{tX}]$ and minimizing $\frac{u(t)}{e^{at}}$ over $t > 0$ gives an upper bound for $Pr(X \geq a)$.

An important situation in computation is the one in which $X_i \in \{0,1\}$, $i = 1, 2, \ldots, n$. For this case, we construct a class of functions of X that is as easy to analyze, and which includes the class $\{e^{tX} : t > 0\}$ and do the above minimization over this class. In the process, we discover that $X_1, X_2, \ldots, X_n$ **need only be** $h(n, \mu, \delta)$**-wise independent** for a suitably defined function $h(\cdot, \cdot, \cdot)$, which is typically much less than n for many algorithms; recall that a set of random variables V exhibit k–wise independence if any subset of k or fewer random variables from V are jointly independent, which is to say that their joint probability distribution function is just the product of the individual distributions. One reason for the use of the e^{tX} function in the classical methods is that $E[e^{tX}]$ generates all higher moments of X; using only a constant number of higher moments, for instance, gives weak bounds. However, in the binary case, the first n moments are sufficient to generate *all* higher moments, which motivates our method. Interestingly, this formulation also can be applied to general X_i that take arbitrary values in the interval $[0, 1]$, even though it is *not* true that the first n moments of $X = \sum_{i=1}^{n} X_i$ determine all higher ones.

The results have many applications to tail probability distributions. They imply similar "limited independence" results when $X_1, X_2, \ldots, X_n$ take values in the interval $[0, 1]$; this can be extended to bounded random variables, by scaling their ranges to $[0, 1]$. In the case of the hypergeometric distribution (sampling without replacement), it provides an elementary mechanism to attain slightly better bounds than those implied in [50] and by Chvátal [30]. The method also yields good upper bounds for the tail probabilities of the sums of random variables with limited independence.

These constructions also provide pointers to further improvement of the independence bounds. For example, we redirect the basic method to obtain improved tools for analyzing the behavior of the sum of k–wise independent random variables. The results simplify and sharpen some of the analyses done in Schmidt & Siegel [109, 110]. In particular, we derive good upper bounds on $E[((\sum_{i=1}^{n} X_i) - E[\sum_{i=1}^{n} X_i])^k]$, where $X_1, X_2, \ldots, X_n$ are k–wise independent random variables, each of which lies in the interval $[0, 1]$; this leads to better independence bounds than our $h(n, \mu, \delta)$ when $\delta < 1$. It can be shown that the independence which we guarantee to be sufficient for the Chernoff–Hoeffding bounds cannot be improved in general by more than a factor of $O(\log n)$. In fact, by combining our constructions with results of [85,6], we get optimal independence (to within a constant factor) and optimal tail probabilities (to within a constant factor in the exponent) simultaneously, for commonly occurring situations such as the case where the X_i's are binary with $Pr(X_i = 1) = 1/2$. We also prove good bounds on the probability of *exactly* r successes in a sequence of k–wise independent Bernoulli trials, which shows that even with modest independence, probabilities and conditional probabilities are close to the fully independent case. in situations like hashing.

The sufficiency of limited independence has several computational applications. First, it means that any random process whose analysis uses the Chernoff–Hoeffding bounds can be simulated with a weaker random source than one which outputs unbiased and independent bits. Next, via known constructions of random variables with limited independence using fewer random bits (Joffe [54], Carter & Wegman [24], Mehlhorn & Vishkin [79], Alon, Babai & Itai [5], Siegel [114]), we can reduce the randomness required for certain algorithms. One example is that of **random sampling**: given a universe U and a subset $X \subseteq U$, the problem is to estimate the fraction of objects of type X in U, such that the absolute error of the output is at most δ with probability at least $1 - \epsilon$, for given error parameters δ and ϵ. The new constructions imply that if R independent samples are required to yield the desired bound, then it in fact suffices for those R samples to be k^*-wise independent, for $k^* = O(\log(\frac{1}{\epsilon}))$. These samples can be generated by $O(\log(\frac{1}{\epsilon}))$ random samples from U, using standard methods. Since the best previously attained bound on R is $O(\frac{1}{\delta^2}\log(\frac{1}{\epsilon}))$, this significantly reduces the amount of randomness needed for this problem. This construction is not optimal with regards to the number of random bits used–see Bellare, Goldreich & Goldwasser [14] for an optimal construction, but it is extremely simple. It is also easily parallelizable, while it is not yet known how to parallelize other schemes for reducing randomness, *e.g.*, random walks on expanders. Via weaker bounds on the kth moment, essentially the same bounds for the random sampling problem have been obtained by Bellare & Rompel [15]. We believe that there should be additional applications yielding reduced randomness "for free". A spectrum of explicit constructions of oblivious routing algorithms on the butterfly with varying time–randomness parameters is among the results of Peleg & Upfal [95]; our "limited independence" result directly matches these bounds on the hypercube and, we believe, should extend to other interconnection networks.

Finally, we combine the method of conditional probabilities [100,117] with the new construction to obtain two results. We get a much faster implementation of the sequential jobshop scheduling algorithm of Shmoys, Stein & Wein [112]. It is comparable in time complexity to the speedups due to Plotkin, Shmoys & Tardos

[97] and Stein [119] but importantly, the approximation bound it presents is better than the ones of [97] and [119]. Here, we show that a problem can be derandomized directly, thereby avoiding the bottleneck step of solving a huge linear program. We also prove an "exact partition" result for set discrepancy, and derive a polynomial-time algorithm for it.

3.1 The Basic Method, and Applications to Tail Probabilities

We now introduce the method, discuss its implications to the tail probabilities of various distributions, and analyze some related approaches. We also prove probability bounds for exactly r successes in a sequence of Bernoulli trials under limited independence. As discussed above, the basic idea used in the Chernoff–Hoeffding (henceforth CH) bounds is as follows. Given n random variables (henceforth "r.v."s) $X_1, X_2, \ldots, X_n$, we want to upper bound the upper tail probability $Pr(X \geq a)$, where $X \doteq \sum_{i=1}^{n} X_i$, $\mu \doteq E[X]$, $a = \mu(1+\delta)$ and $\delta > 0$. For any fixed $t > 0$,

$$Pr(X \geq a) = Pr(e^{tX} \geq e^{at}) \leq \frac{E[e^{tX}]}{e^{at}};$$

by computing an upper bound $u(t)$ on $E[e^{tX}]$ and minimizing $\frac{u(t)}{e^{at}}$ over $t > 0$, we can upper bound $Pr(X \geq a)$. When $X_1, X_2, \ldots, X_n$ are binary, we construct a class of functions of X that includes the class $\{e^{tX} : t > 0\}$ and do the minimization over this class; in the process, we discover that $X_1, X_2, \ldots, X_n$ `need only be` $h(n, \mu, \delta)$`-wise independent` for a function $h(\cdot,\cdot,\cdot)$ that will be defined in equation 3.3 of the next section.

Notation. If x is real and r is a positive integer, then $\binom{x}{r}$ will denote, as usual, $\frac{x(x-1)\cdots(x-r+1)}{r!}$ with $\binom{x}{0} \doteq 1$.

3.1.1 Estimating tail probabilities of binary random variables

The CH bounds are frequently used when the r.v.'s $X_1, X_2, \ldots X_n$ are binary and independent. In this section, we first assume that $X_1, X_2, \ldots X_n$ are 0–1 independent r.v.'s with $Pr(X_i = 1) = p_i$, $1 \leq i \leq n$; the independence assumption will be relaxed later, and the results will be extended to r.v.'s X_i with $0 \leq X_i \leq 1$, in Section 3.1.2. Let $X \doteq \sum_{i=1}^{n} X_i$, and $\mu \doteq E[X] = \sum_{i=1}^{n} p_i$. The reader is referred to Section 2.2 for a quick survey of the CH-bounds. In short, the estimate derived there is

$$Pr(X \geq \mu(1+\delta)) \leq F(n,\mu,\delta) \doteq \frac{(1 + \frac{\mu\delta}{(n-\mu(1+\delta))})^{n-\mu(1+\delta)}}{(1+\delta)^{\mu(1+\delta)}}, \text{ with}$$

$$F(n,\mu,\delta) \leq G(\mu,\delta) \doteq (\frac{e^{\delta}}{(1+\delta)^{1+\delta}})^{\mu}.$$

The basic idea behind these estimates is the simple fact that for any $a > \mu$ and any $t > 0$,

$$Pr(X \geq a) = Pr(e^{tX} \geq e^{at}) \leq \frac{E[e^{tX}]}{e^{at}}$$

and hence,

$$Pr(X \geq a) \leq \quad min_t \frac{E[e^{tX}]}{e^{at}}.$$

Hence, at the heart of such estimates are the simple calculations associated with the multiplicative nature of $E[e^{\Sigma X_i}]$. Recall that $e^{tX} = \sum_{i=0}^{\infty} \frac{t^i}{i!} X^i$. Consider X^2, for instance. $X^2 = (X_1 + X_2 + \cdots + X_n)^2 = \sum_{i=1}^{n} X_i^2 + 2\sum_{1 \leq i_1 < i_2 \leq n} X_i X_j = \sum_{i=1}^{n} X_i + 2\sum_{1 \leq i_1 < i_2 \leq n} X_i X_j$, since $X_i^2 = X_i$ for $X_i \in \{0,1\}$. Similarly, other higher powers of the X_i's are unnecessary, implying that a form simpler and more useful than functions of the form $\{e^{tX} : t > 0\}$ might exist. There are many ways to formalize this. We define, for $z = (z_1, z_2, \ldots, z_n) \in \Re^n$, a family of symmetric multilinear polynomials $S_j(z), j = 0, 1, \ldots, n$, where $S_0(z) \equiv 1$, and for $1 \leq j \leq n$, $S_j(z) \doteq \sum_{1 \leq i_1 < i_2 \cdots < i_j \leq n} z_{i_1} z_{i_2} \cdots z_{i_j}$. We start with the simple

Lemma 6 *Suppose $z_1, z_2, \ldots, z_n$ take on binary values. Then for any positive integer j, $(z_1 + z_2 \cdots + z_n)^j \equiv \sum_{i=1}^{min(j,n)} a_i S_i(z_1, z_2, \ldots, z_n)$, for some* **non-negative** *integers $a_1, a_2, \ldots, a_{min(j,n)}$.*

The proof of Lemma 6 is trivial and is omitted. The converse of Lemma 6 also is true; if $z = (z_1, z_2, \ldots, z_n) \in \{0,1\}^n$, then for any j, $j = 0, 1, \ldots, n$,

$$\forall (u_0, \ldots, u_j) \in \Re^{j+1} \; \exists (v_0, \ldots, v_j) \in \Re^{j+1} : \sum_{i=0}^{j} u_i S_i(z) \equiv \sum_{i=0}^{j} v_i (z_1 + z_2 + \cdots + z_n)^i.$$

So, the two forms: polynomial of $z_1 + z_2 + \cdots + z_n$ and linear combination of $S_0(z), S_1(z), \ldots, S_n(z)$ are equivalent. Note that if the binary random variables $X_1, X_2, \ldots, X_n$ are independent, then $E[S_i(X_1, X_2, \ldots, X_n)]$ is explicitly available: $E[S_i(X_1, X_2, \ldots, X_n)] = S_i(p_1, p_2, \ldots, p_n)$, where $p_j = Pr(X_j = 1)$. This explains our preference for the S_i's.

Since the expansion $e^{tZ} = \sum_{i=0}^{\infty} \frac{t^i}{i!} Z^i$ converges for all t and Z and since all the coefficients $\frac{t^i}{i!}$ are positive if $t > 0$, we get

Corollary 2 *Let $z_1, z_2, \ldots, z_n$ take on binary values. Then for any $t > 0$, there exist* **non-negative** *reals $a_0, a_1, \ldots, a_n$ such that $e^{t(z_1+z_2+\cdots z_n)} \equiv \sum_{i=0}^{n} a_i S_i(z_1, z_2, \ldots, z_n)$.*

One reason for the use of the mysterious e^{tX} function in the CH bounds is the need for higher moments of X. In particular, the **moment generating function** of X is defined to be $E[e^{tX}] = \sum_{i=0}^{\infty} \frac{t^i}{i!} E[X^i]$; its derivatives generate all higher moments of X. Moreover, the use of moment generating functions embed the problem of attaining probability estimates in a space rich with algebraic structure and convex inequalities. (More about the computational aspects of such an alternative approach can be found in [113].) The need for higher moments is due to the fact that a direct application of Markov's or Chebyshev's inequality to upper bound $Pr(X \geq E[X] \cdot (1 + \delta))$ leads to weak bounds. Higher moments and exponentials give dramatically better estimates. But when X is the sum of random bits $X_1, X_2, \ldots, X_n$, Lemma 6 and Corollary 2 imply that *all* the higher moments of X can be linearly generated by

$\{E[S_i(X_1, X_2, \ldots, X_n)] \; : \; i = 0, 1, \ldots, n\}$. Equivalently, they are also generated linearly by *any* n higher moments of X.

So, we now consider functions of the form $\sum_{i=0}^{n} y_i S_i(X_1, \ldots, X_n)$ where $y_0, \ldots, y_n \geq 0$, instead of restricting ourselves to those of the form e^{tX}, for some $t > 0$. Indeed, by Corollary 2, we will be considering a class of functions which includes the class $\{e^{tX} : t > 0\}$. For any $y = (y_0, y_1, \ldots, y_n) \in \Re_+^{n+1}$ and $z = (z_1, z_2, \ldots z_n) \in \Re^n$, define $f_y(z) \doteq \sum_{i=0}^{n} y_i S_i(z)$. With this notation, we can restate Corollary 2 as $\forall t > 0 \; \exists y \in \Re_+^{n+1} : f_y(X_1, X_2, \ldots, X_n) = e^{tX}$. Let $a \doteq \mu(1+\delta)$ be assumed to be integral. Note that for any non-negative integer m, $X = m$ iff $f_y(X_1, X_2, \ldots, X_n) = \sum_{i=0}^{m} y_i \binom{m}{i}$ and hence,

$$\begin{aligned} Pr(X \geq a) &= Pr\left(f_y(X_1, \ldots, X_n) \geq \sum_{i=0}^{a} y_i \binom{a}{i}\right) \\ &\leq \frac{E[f_y(X_1, \ldots, X_n)]}{\sum_{i=0}^{a} y_i \binom{a}{i}} \\ &= \frac{\sum_{i=0}^{n} y_i S_i(p_1, p_2, \ldots, p_n)}{\sum_{i=0}^{a} y_i \binom{a}{i}}. \end{aligned} \tag{3.1}$$

So, our goal now is to minimize this upper bound over $(y_0, y_1, \ldots y_n) \in \Re_+^{n+1}$. To do this, note that $y_{a+1}, y_{a+2}, \ldots, y_n$ must all be set to 0 since they contribute non-negative terms to the numerator and nothing to the denominator. Next, note that (3.1) is minimized by setting $y_i = 1$ if $i = j^*$ and 0 otherwise, where j^* is the integer at which $S_i(p_1, p_2, \ldots, p_n)/\binom{a}{i}$ is minimized, over the range $i = 0, 1, \ldots, a$. To get a better handle on this minimum, we need

Lemma 7 *For any $i > 0$ and $s \geq 0$, $S_i(z_1, z_2, \ldots, z_n)$ is maximized by setting $z_1 = z_2 \cdots = z_n = \frac{s}{n}$, when subject to the constraints that $(z_1, z_2, \ldots, z_n) \in \Re_+^n$ and $\sum_{j=1}^{n} z_j = s$.*

PROOF. Suppose $z_p < z_q$ for some vector z satisfying the constraints $(z_1, \ldots, z_n) \in \Re_+^n$ and $\sum_{j=1}^{n} z_j = s$. Then, set $z'_p = z_p + \epsilon$ and $z'_q = z_q - \epsilon$ for any $\epsilon < z_q - z_p$, and set $z'_j = z_j$ for all indices j, $j \notin \{p, q\}$. It is easy to verify that z' satisfies the above constraints, and that $S_i(z') > S_i(z)$. Hence $S_i(z)$ is maximized at $z = z^*$, where $z^*_j = \frac{s}{n}$ for $j = 1, 2, \ldots, n$. $\square$

Inequality(3.1) and Lemma 7 imply that if $p \doteq \frac{\sum_{i=1}^n p_i}{n} = \frac{\mu}{n}$, then for any $y \in \Re_+^{n+1}$,

$$Pr(X \geq a) \leq \frac{\sum_{i=0}^n y_i \binom{n}{i} p^i}{\sum_{i=0}^a y_i \binom{a}{i}}. \tag{3.2}$$

Since $\frac{\binom{n}{i+1}p^{i+1}/\binom{a}{i+1}}{\binom{n}{i}p^i/\binom{a}{i}} = \frac{(n-i)p}{a-i}$ which is less than, equal to, or greater than 1 according as i is less than, equal to, or greater than $\frac{a-np}{1-\mu/n}$, $\binom{n}{i}p^i/\binom{a}{i}$ is minimized at

$$i^* = h(n, \mu, \delta) \doteq \lceil \frac{a-\mu}{1-\mu/n} \rceil = \lceil \frac{\mu\delta}{1-\mu/n} \rceil. \tag{3.3}$$

So, the r.h.s. of inequality(3.2) is minimized at $y = y^*$, where $y^*_i = 1$ if $i = i^*$, and 0 otherwise. Hence, we get

$$Pr(X \geq \mu(1+\delta)) \leq U_1(n, p_1, \ldots, p_n, \delta) \doteq \frac{S_{j^*}(p_1, p_2, \ldots, p_n)}{\binom{\mu(1+\delta)}{j^*}} \leq U_2(n, \mu, \delta) \doteq \frac{\binom{n}{i^*}(\frac{\mu}{n})^{i^*}}{\binom{\mu(1+\delta)}{i^*}}.$$

$U_1(n, p_1, \ldots, p_n, \delta)$ is guaranteed to be better than any estimate based on the CH method, since we have considered a larger class of functions. Also, the upper bound $U_2(n, \mu, \delta)$ on $U_1(n, p_1, \ldots, p_n, \delta)$ is better than any such estimate which depends only on μ and which is oblivious to the actual values of $p_1, p_2, \ldots, p_n$; this includes $F(n, \mu, \delta)$ and $G(\mu, \delta)$.

But most importantly, note that *these new bounds will hold even if* $X_1, X_2, \ldots X_n$ *are only* $h(n, \mu, \delta)$*-wise independent.* This is because each term in $S_k(X_1, X_2, \ldots, X_n)$ is of the form $X_{i_1}X_{i_2}\ldots X_{i_k}$ for any integer k, and hence, $E[S_k(X_1, X_2, \ldots X_n)]$ will be the same for k-wise independent $X_1, X_2, \ldots X_n$ as for completely independent $X_1, X_2, \ldots X_n$. Since $\mu(1+\delta) \leq n$, $h(n, \mu, \delta) \leq n$; in typical algorithmic situations, $h(n, \mu, \delta) \ll n$. This will be seen to be of great use later on.

Theorem 5 *Let bits* $X_1, X_2, \ldots, X_n$ *be random with* $Pr(X_i = 1) = p_i$, $X = \sum_{i=1}^n X_i$ *and* $\mu = E[X] = \sum_{i=1}^n p_i$. *Suppose further, that the* X_i *are* k*-wise independent for* $k \geq h(n, \mu, \delta)$. *Then for any* $\delta > 0$,

$$Pr(X \geq \mu(1+\delta)) \leq U_1(n, p_1, \ldots, p_n, \delta) \leq U_2(n, \mu, \delta).$$

Furthermore, $U_2(n,\mu,\delta) \leq F(n,\mu,\delta) \leq G(\mu,\delta)$, *i.e.,* the CH upper bounds hold even if the X_is are only $h(n,\mu,\delta)$-wise independent.

Our results also imply upper tail bounds for r.v.'s with smaller independence than $h(n,\mu,\delta)$.

Lemma 8 *Let $X_1, X_2, \ldots, X_n$ be binary r.v.'s with $X \doteq \sum_{i=1}^n X_i$ and $\mu \doteq E[X]$. Then for any $\delta > 0$, $Pr(X \geq \mu(1+\delta)) \leq \binom{n}{k}(\mu/n)^k/\binom{\mu(1+\delta)}{k}$, if the X_i's are k-wise independent for any $k < h(n,\mu,\delta)$.*

PROOF. Set $y_i = 0$ for $i \neq k$ and $y_k = 1$ in inequality (3.2). □

It turns out that $U_2(n,\mu,\delta)$ is almost the same as $F(n,\mu,\delta)$.

Theorem 6 *Given n random bits $X_1, X_2, \ldots, X_n$, let $X = \sum_{i=1}^n X_i$, $\mu = E[X]$, and $p \doteq \mu/n$. Then for any $\delta > 0$,*

1. *If the X_i's are are $\lceil \mu\delta \rceil$-wise independent, then $Pr(X \geq \mu(1+\delta)) \leq G(\mu,\delta)$, where:*

$$G(\mu,\delta) = \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu \leq \begin{cases} e^{-\mu\delta^2/3}, & \text{if } \delta < 1; \\ e^{-\mu\delta\ln(1+\delta)/2} \leq e^{-\mu\delta/3}, & \text{if } \delta \geq 1. \end{cases}$$

2. *If the X_i's are $\lceil \frac{\mu\delta}{1-p} \rceil$-wise independent, then $Pr(X \geq \mu(1+\delta)) \leq F(n,\mu,\delta)$.*

3. *If the X_i's are $\lceil \frac{\mu\delta}{p} \rceil = \lceil n\delta \rceil$-wise independent, then $Pr(X \leq \mu(1-\delta)) \leq F(n,\mu,-\delta)$, where:*

$$F(n,\mu,-\delta) \leq \begin{cases} e^{-\mu\delta^2/(2(1-p))}, & \text{if } p \leq 1/2; \\ e^{-2p\mu\delta^2}, & \text{if } p > 1/2. \end{cases}$$

PROOF. The first claim follows by setting $k = \lceil \mu\delta \rceil \leq h(n,\mu,\delta)$ in Lemma 8. The only interesting case is that $k < h(n,\mu,\delta)$. We apply the inequality

$$\binom{n}{k}(\mu/n)^k/\binom{a}{k} \leq Q(n,k,a) \doteq \left(\frac{n}{a}\right)^k(\mu/n)^k\left(\frac{n}{n-k}\right)^{n-k}\left(\frac{a-k}{a}\right)^{a-k},$$

valid for any $a < n$; this inequality follows by induction on k, combined with the fact that the function $(1 - \frac{1}{x})^{x-1}$ is nonincreasing for $x > 1$. Let $a = (1+\delta)\mu$ and $k' = \mu\delta$. Then,

$$\begin{aligned} Q(n, k', a) &= \frac{(n/(n-\mu\delta))^{n-\mu\delta}}{(1+\delta)^{\mu(1+\delta)}} \\ &= \left(\frac{1}{1+\delta}\right)^{\mu(1+\delta)} \left(1 + \frac{\mu\delta}{n-\mu\delta}\right)^{n-\mu\delta} \\ &\leq \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu} = G(\mu, \delta). \end{aligned}$$

It is easy to verify that $Q(n, x, a)$ is nonincreasing for $x \leq h(n, \mu, \delta)$, and hence the bound $G(\mu, \delta)$ established for k' also holds for $k = \lceil \mu\delta \rceil$. The upper bounds for $G(\mu, \delta)$ are either straightforward or have been established in [9,101,8].

The second claim follows immediately from Theorem 5, while the third claim follows by obtaining lower tail bounds from the upper tail of $\sum_{i=1}^{n}(1 - X_i)$, and importing the upper tail bounds established in [50]. By Theorem 5, these bounds hold with independence $h(n, \mu, \delta)$. □

As we will show in Section 3.1.3, bounds almost as good as $G(\mu, \delta)$ and $F(n, \mu, -\delta)$ hold with the much smaller independence $k = \lfloor \mu\delta^2 \rfloor$, when $\delta < 1$.

3.1.2 Tail probabilities of bounded random variables

We now show that almost the same results hold for arbitrary r.v.'s which take values in $[0, 1]$. Analogous bounds for bounded r.v.'s that are constrained to lie in other intervals can be obtained by a linear transformation of their ranges to $[0, 1]$. Given arbitrary r.v.'s X_i such that $0 \leq X_i \leq 1$, $i = 1, 2, \ldots, n$, we wish to upper bound $Pr(X \geq \mu(1+\delta))$, where $X = \sum_{i=1}^{n} X_i$, $\mu = E[X]$ and $\delta > 0$. Hoeffding has proved upper tail bounds for bounded random variables, assuming full independence among the X_i's; the main point of interest here again is that partial independence suffices, giving almost as good bounds as Hoeffding's. Almost all of the work is done by

Lemma 9 *Let z_i be real numbers, with $0 \leq z_i \leq 1$, $i = 1, 2, \ldots, n$, and suppose that $a \geq 0$, $j \leq \lfloor a \rfloor$ and $\sum_{i=1}^n z_i \geq a$. Then, $S_j(z_1, z_2, \ldots, z_n) \geq \binom{a}{j}$.*

PROOF. We will just consider the case $\sum_{i=1}^n z_i = a$; then the upper bound will directly follow if $\sum_{i=1}^n z_i > a$. If $0 < z_p \leq z_q < 1$ for $p \neq q$, then $S_j(z)$ decreases if we set $z_p := z_p - \epsilon$ and $z_q := z_q + \epsilon$, for any $\epsilon < \min(z_p, 1 - z_q)$. So, if $S_j(z)$ is minimized at z^* in the domain $[0,1]^n$ under the constraint that $\sum_{i=1}^n z_i = a$, then $0 < z_i^* < 1$ for at most one i, $1 \leq i \leq n$.

If a is integral, then $z_i^* \in \{0, 1\}$, $i = 1, 2, \ldots, n$, and hence $S_j(z^*) = \binom{a}{j}$. Otherwise, suppose a is non-integral; let $a_1 = \lfloor a \rfloor$ and $a_2 = a - a_1$. Hence, $z_p^* = a_2$ for some index p, and $z_i^* \in \{0, 1\}$ for $i \in \{1, 2, \ldots, n\} - \{p\}$; so,

$$S_j(z^*) = \binom{a_1}{j} + a_2 \cdot \binom{a_1}{j-1},$$

and we need to show that this is at least $\binom{a}{j}$; *i.e.*, that

$$[a_1]_j + a_2 j \cdot [a_1]_{j-1} \geq [a_1 + a_2]_j = [a]_j,$$

where $[x]_r$ denotes $x(x-1)\cdots(x-r+1)$ and $[x]_0 \doteq 1$. This is easily seen by induction on j, as follows. Equality clearly holds for $j = 1$. For $j > 1$, $[a_1]_j + a_2 j \cdot [a_1]_{j-1} = [a_1]_j + (j-1)a_2[a_1]_{j-1} + a_2[a_1]_{j-1}$. Since $a_2 < 1$, $a_2[a_1]_{j-1} \geq a_2[a_1 + a_2 - 1]_{j-1}$ and hence,

$$\begin{aligned}
[a_1]_j + a_2 j \cdot [a_1]_{j-1} \quad &\geq \quad a_1([a_1 - 1]_{j-1} + (j-1)a_2[a_1 - 1]_{j-2}) + \\
&\qquad a_2[a_1 + a_2 - 1]_{j-1} \\
&\overset{ind\,hyp.}{\geq} \quad a_1[a_1 + a_2 - 1]_{j-1} + a_2[a_1 + a_2 - 1]_{j-1} \\
&= \quad [a_1 + a_2]_j.
\end{aligned}$$

□

By essentially the same analysis as before, we get

Theorem 7 *Given n arbitrary r.v.'s $X_1, X_2, \ldots, X_n$ with $0 \leq X_i \leq 1$ and $E[X_i] = p_i$, let $X \doteq \sum_{i=1}^{n} X_i$ and $\mu \doteq E[X]$. Then if $X_1, X_2, \ldots, X_n$ are k–wise independent for $k \geq h(n, \mu, \delta)$, then $Pr(X \geq \mu(1+\delta)) \leq U_1(n, p_1, \ldots, p_n, \delta) \leq U_2(n, \mu, \delta)$, for any $\delta > 0$.*

PROOF. From Lemma 9, we have that for any $a > 0$ and for *non–negative* $y_0, y_1, \ldots, y_n$,

$$Pr(X \geq a) \leq Pr(f_y(X_1, \ldots, X_n) \geq \sum_{i=0}^{\lfloor a \rfloor} y_i \binom{a}{i}) \leq \frac{E[f_y(X_1, \ldots, X_n)]}{\sum_{i=0}^{\lfloor a \rfloor} y_i \binom{a}{i}},$$

and the rest of the proof follows as before. □

Remark. The methods of Section 3.1.1 were motivated by the fact that if X is the sum of n 0–1 random variables, then any n higher moments of X linearly generate all the higher moments of X. However, note that if random variables $X_1, X_2, \ldots, X_n$ take arbitrary values in the interval $[0, 1]$ and if $X = \sum_{i=1}^{n} X_i$, then such a result is *not* true: in fact, no bound can be put on the number of higher moments needed to generate all the moments of X. However, the intuition gained from Section 3.1.1 has helped us obtain a large deviation bound for X, which is as good as the known bound [50]. This is despite the fact that we have not considered all the higher moments of X; one of the original motivations for Chernoff to consider $E[e^{tX}]$ was that it generates all the higher moments of X. A possible interpretation of our result of this subsection is that it pinpoints the "crucial" higher moments.

3.1.3 Redirecting the method

Recall that in Section 3.1.1 we introduced the class of functions $S_0(z), S_1(z), \ldots, S_n(z)$ and generalized Chernoff's idea by working with `non-negative` linear combinations of these functions. A natural generalization of this is to allow arbitrary linear combinations, but the corresponding optimization problem, described below, seems hard to analyze.

Suppose we have n *binary* random variables $X_1, X_2, \ldots, X_n$ with $Pr(X_i = 1) = p_i$ and with $X = \sum_{i=1}^{n} X_i$, and we want good upper bounds on $Pr(X \geq a)$ where $a > E[X]$, when the X_i's are k–wise independent. As before, let

$$f_y(X_1, X_2, \ldots, X_n) = \sum_{i=0}^{n} y_i S_i(X_1, X_2, \ldots, X_n),$$

with the further restriction that $y_i = 0$ for $i \geq k+1$ to capture the idea of k–wise independence; note that $f_y(X_1, X_2, \ldots, X_n)$ is a function of X,

$$f_y(X_1, X_2, \ldots, X_n) = g_y(X) \doteq \sum_{i=0}^{min(k,X)} y_i \binom{X}{i}.$$

Now, if $g(t) \geq 0$ for $t = 0, 1, \ldots, n$ (so that Markov's inequality can be applied) and if $g_y(b) \geq g_y(a)$ for $b \geq a$, then

$$Pr(X \geq a) \leq Pr(g_y(X) \geq g_y(a)) \leq \frac{E[g_y(X)]}{g_y(a)} = \frac{\sum_{i=0}^{k} y_i S_i(p_1, p_2, \ldots, p_n)}{g_y(a)}.$$

We can scale the y_i's so that $g_y(a) = 1$ and thus, we get the following linear program with $y_0, y_1, \ldots, y_k$ being arbitrary real variables.

$LP(a, k, p_1, p_2, \ldots, p_n)$:

Minimize $\sum_{i=0}^{k} y_i S_i(p_1, p_2, \ldots, p_n)$ subject to

- $g_y(j) \geq 0$, $j = 0, 1, \ldots, n$.
- $g_y(a) = 1$, and
- $g_y(b) \geq 1$, $b = a+1, a+2, \ldots, n$.

We unfortunately have been unable to analytically compute the optimum of this linear program. However, we now consider an important case where some of the multipliers are negative, and which is a feasible solution to the above LP; our results generalize a result of [68,17,81]. We use the kth moment inequality

$$Pr(|X - E[X]| \geq \delta E[X]) \leq \frac{E[|X - E[X]|^k]}{(\delta E[X])^k},$$

which is attributable, in various formulations and generalizations, to Chebyshev, Markov, and Loéve [51], and has been used to attain probability deviation estimates for over a century. Note that if $X = X_1+X_2+\cdots+X_n$ where the X_i's are random bits, then $(X - E[X])^k$ is a linear combination of $S_0(X_1, X_2, \ldots, X_n)$, $S_1(X_1, X_2, \ldots, X_n)$, $\ldots$, $S_k(X_1, X_2, \ldots, X_n)$, with some of the multipliers being negative. We derive good upper bounds on $E[|X - E[X]|^k]$, where $X = \sum_{i=1}^n X_i$, with the X_i's being k–wise independent random variables which satisfy $|X_i - E[X_i]| \leq 1$, yielding bounds that are better than those given by Theorem 5 and Lemma 8, when $k \ll h(n, \mu, \delta)$ and $\delta < 1$. Moreover, the large deviation bounds derived in Theorem 9 for k-wise independent random variables agree with the simple exponential forms of the large deviation bounds most often cited for sequences of fully independent Bernoulli trials.

Theorem 8 is similar in spirit and proof to Lemma 4.19 of [68] for identically distributed X_i and constant k, but the present result is somewhat tighter even in the case of identically distributed X_i, especially if $X = \sum_{i=1}^n X_i$ has small variance. A slightly weaker form of a special case of one of the inequalities proven in Theorem 8 was also obtained in [17] and some related formulae were given in [81]. The proof of Theorem 8, as well as related proofs presented elsewhere, is based upon estimates for the k-th moment of X. Estimates related to ours, but for a more general class of random variables, were established in [84]. That formulation however, is considerably more complicated than ours, and is not as tight for the cases specifically considered here. In particular, Theorem 9 cannot be derived from the bounds in [84] for the k-th moment. Other related work was done by Gladkov ([43], with later improvements in [44]). He shows that if $Y_1, Y_2, \ldots, Y_n$ are *independent* r.v.'s with Y_i having the same distribution as X_i and with $Y \doteq Y_1 + Y_2 + \cdots + Y_n$, then as $n \to \infty$, the convergence of Y to the normal distribution implies a comparable convergence for X, provided k is sufficiently large.

Theorem 8 *Let $X_1, \ldots, X_n$ be a sequence of k-wise independent random variables, that satisfy $|X_i - \mathrm{E}[X_i]| \leq 1$. Let $X = \sum_{i=1}^n X_i$ with $E[X] = \mu$, and let $\sigma^2[X]$ denote* *the variance of X, so that $\sigma^2[X] = \sum_{i=1}^n \sigma^2[X_i]$ (provided $k \geq 2$, which we require.) Then the following hold for any even k.*

(I) *For*[1] $C \geq \sigma^2[X]$, $\quad Pr(|X-\mu| \geq T) \leq \sqrt{2}\cosh\left(\sqrt{\frac{k^3}{36C}}\right)\left(\frac{kC}{eT^2}\right)^{k/2}$.

(II) *For* $2 \leq k \leq 3(\sigma^2[X])^{1/3}$, $\quad Pr(|X-\mu| \geq T) \leq 2\left(\frac{k\sigma^2[X]}{eT^2}\right)^{k/2}$.

(III) *For* $C \geq max\{k, \sigma^2[X]\}$, $\quad Pr(|X-\mu| \geq T) \leq \left(\frac{kC}{e^{2/3}T^2}\right)^{k/2}$.

PROOF. We use the kth moment inequality

$$Pr(|X-\mu| \geq T) \leq \frac{\mathrm{E}\left[|X-\mu|^k\right]}{T^k}. \tag{3.4}$$

For even k, $\mathrm{E}\left[|X-\mu|^k\right] = \mathrm{E}\left[(X-\mu)^k\right]$. Most of our effort will therefore be invested in estimating the k-th moment of $X-\mu$. Let $p_i = \mathrm{E}[X_i]$, for $1 \leq i \leq n$. Then,

$$\begin{aligned} \mathrm{E}\left[(X-\mu)^k\right] &= \mathrm{E}\left[\left(\sum_{i=1}^n (X_i - p_i)\right)^k\right] \\ &= \sum_{i_1+\cdots+i_n=k} \binom{k}{i_1,\ldots,i_n} \prod_{j=1}^n \mathrm{E}[(X_j-p_j)^{i_j}]. \end{aligned} \tag{3.5}$$

Clearly, $\mathrm{E}[X_i - p_i] = 0$ and any term in (3.5) that has some $i_j = 1$ must be zero. More generally, $|\mathrm{E}[X_i - p_i]^\ell| \leq \sigma^2[X_i]$ for any $\ell \geq 2$, since $|X_i - p_i| \leq 1$ and therefore $|X_i - p_i|^\ell \leq |X_i - p_i|^2$, hence $|\mathrm{E}[X_i - p_i]^\ell| \leq |\mathrm{E}[X_i - p_i]^2| = \sigma^2[X_i]$. Thus,

$$\begin{aligned} \mathrm{E}\left[(X-\mu)^k\right] &= \sum_{\ell=0}^{k/2-1} \sum_{\substack{j_1+\cdots+j_{k/2-\ell}=k \\ j_i \geq 2}} \binom{k}{j_1,\ldots,j_{k/2-\ell}} \sum_{i_1<\cdots<i_{k/2-\ell}} \prod_{r=1}^{k/2-\ell} \mathrm{E}[X_{i_r} - p_{i_r}]^{j_r} \\ &\leq \sum_{\ell=0}^{k/2-1} \sum_{\substack{j_1+\cdots+j_{k/2-\ell}=k \\ j_i \geq 2}} \binom{k}{j_1,\ldots,j_{k/2-\ell}} \sum_{i_1<\cdots<i_{k/2-\ell}} \prod_{r=1}^{k/2-\ell} \sigma^2[X_{i_r}] \\ &\leq \sum_{\ell=0}^{k/2-1} \sum_{\substack{j_1+\cdots+j_{k/2-\ell}=k \\ j_i \geq 2}} \binom{k}{j_1,\ldots,j_{k/2-\ell}} \frac{\left(\sigma^2[X]\right)^{k/2-\ell}}{(k/2-\ell)!}. \end{aligned} \tag{3.6}$$

[1] Recall that $\cosh(x) = \frac{e^x + e^{-x}}{2}$, where e denotes, as usual, the base of natural logarithms.

Estimate (3.6) comes about because the summation

$$\sum_{i_1<i_2<\cdots<i_{k/2-\ell}} \prod_{r=1}^{k/2-\ell} \sigma^2[X_{i_r}]$$

is maximized when all the $\sigma^2[X_{i_c}]$ are equal, by Lemma 7, and hence is at most

$$\binom{n}{k/2-\ell}\left(\frac{\sigma^2[X]}{n}\right)^{k/2-\ell} \le \frac{\left(\sigma^2[X]\right)^{k/2-\ell}}{(k/2-\ell)!}.$$

Let $T_0, T_1, \ldots T_{k/2-1}$ denote the $k/2$ terms in summation (3.6), hence:

$$T_\ell = \sum_{\substack{j_1+j_2+\cdots+j_{k/2-\ell}=k \\ j_i\ge 2}} \binom{k}{j_1, j_2, \ldots, j_{k/2-\ell}} \frac{\left(\sigma^2[X]\right)^{k/2-\ell}}{(k/2-\ell)!}, \tag{3.7}$$

and

$$T_0 = \binom{k}{2,\ 2, \ldots, 2} \frac{\left(\sigma^2[X]\right)^{k/2}}{(k/2)!}. \tag{3.8}$$

There are exactly $\binom{k/2+\ell-1}{2\ell}$ terms (*i.e.*, sets of assignments for $j_1, \ldots, j_{k/2-\ell}$) in T_ℓ, since $\sum j_i = k$ and $j_i \ge 2$. For each such assignment of values, $\sum_{k=1}^{k/2-\ell}(j_k - 2) = 2\ell$, hence $\prod_{k=1}^{k/2-\ell} j_k! \ge 2^{k/2-\ell}3^{2\ell}$ (and equality holds only when $\ell \le k/6$ and exactly 2ℓ of the $k/2-\ell$ values equal 3, while the remaining values equal 2). Hence $\binom{k}{j_1,j_2,\ldots,j_{k/2-\ell}} \le (2/9)^\ell \binom{k}{2,\ 2,\ldots,2}$, and

$$T_\ell \le \binom{k/2+\ell-1}{2\ell}\left(\frac{2}{9\sigma^2[X]}\right)^\ell \frac{(k/2)!}{(k/2-\ell)!} T_0.$$

Since $\binom{k/2+\ell-1}{2\ell}\frac{(k/2)!}{(k/2-\ell)!} < \frac{(k/2)^{3\ell}}{(2\ell)!}$, we have $\mathrm{E}\left[(X-\mu)^k\right] \le T_0 \sum_{\ell=0}^{k/2-1} \frac{k^{3\ell}}{(36\sigma^2[X])^\ell (2\ell)!}$. Using Taylor's Series for $\cosh(x) = \frac{e^x+e^{-x}}{2} = \sum_j \frac{x^{2j}}{(2j)!}$, we see that

$$\sum_{\ell=0}^{k/2-1} \frac{\left(\frac{k^3}{36\sigma^2[X]}\right)^{\frac{2\ell}{2}}}{(2\ell)!} \le \cosh\left(\sqrt{\frac{k^3}{36\sigma^2[X]}}\right).$$

Consequently

$$\mathrm{E}\left[(X-\mu)^k\right] \le \cosh\left(\sqrt{\frac{k^3}{36\sigma^2[X]}}\right) T_0. \tag{3.9}$$

T_0 is readily bounded by expanding (3.8) to get $T_0 = \frac{k!}{2^{k/2}(k/2)!}(\sigma^2[X])^{k/2}$. We may apply a strong version of Stirling's Formula [105]:

$$(r/e)^r\sqrt{2\pi r}e^{1/(12r+1)} \le r! \le (r/e)^r\sqrt{2\pi r}e^{1/(12r)}.$$

which is valid for all $r \ge 1$, to bound both $k!$ and $(k/2)!$. This yields $T_0 \le \sqrt{2}\left(\frac{k\sigma^2[X]}{e}\right)^{k/2}$. Substituting for T_0 in (3.9) gives

$$\mathrm{E}\left[(X-\mu)^k\right] \le \sqrt{2}\cosh\left(\sqrt{\frac{k^3}{36\sigma^2[X]}}\right)\left(\frac{k\sigma^2[X]}{e}\right)^{k/2}, \tag{3.10}$$

which establishes the desired bound for $\mathrm{E}\left[(X-\mu)^k\right]$.

Now, estimate (3.6) is an increasing function of $\sigma^2[X]$, and the estimate in (3.10) exceeds (3.6). Therefore $\sigma^2[X]$ can be replaced by any $C \ge \sigma^2[X]$ in (3.10). The proof of (I) is completed by applying this estimate to (3.4).

All other bounds are special cases of (I). When $k \le 3(\sigma^2[x])^{1/3}$, we use $C = \sigma^2[X]$ in Theorem 8.I and overestimate $\cosh\left(\sqrt{\frac{k^3}{36\sigma^2[X]}}\right)$ by $\cosh\left(\sqrt{\frac{3}{4}}\right) < \sqrt{2}$.

(III) is easily verified for $k = 2$, by applying Chebyshev's inequality:

$$Pr(|X-\mu| \ge T) \le \frac{\sigma^2[X]}{T^2} \le \frac{2\sigma^2[X]}{e^{2/3}T^2}$$

For $k \ge 4$ we may replace C by $max\{\sigma^2[X], k\}$ in (I) and overestimate $\cosh\left(\sqrt{\frac{k^3}{36C}}\right)$ by $\cosh(k/6)$. Since $\cosh(x) < e^x/\sqrt{2}$ for $x \ge 1/2$, we get $\cosh(k/6) \le e^{k/6}/\sqrt{2}$, and hence

$$Pr(|X-\mu| \ge T) \le \left(\frac{kC}{e^{2/3}T^2}\right)^{k/2}.$$

This concludes the proof of estimate (III). □

We now combine the results of Theorem 8 and Theorem 6 to establish Chernoff-like bounds [27,50], where the independence k might even be much smaller than the deviation we wish to bound.

Theorem 9 *If X is the sum of k–wise independent r.v.'s, each of which is confined to the interval $[0,1]$ with $\mu = E[X]$, then:*

(I) *For* $\delta \le 1$ *and* $k = \left\lfloor \delta^2 \mu e^{-1/3} \right\rfloor$, $\quad Pr(|X - \mu| \ge \delta\mu) \le e^{-\lfloor k/2 \rfloor} \le e^{-\lfloor \delta^2 \mu/3 \rfloor}$.

(II) *For* $\delta \ge 1$ *and* $k = \left\lfloor \delta \mu e^{-1/3} \right\rfloor$, $\quad Pr(|X - \mu| \ge \delta\mu) \le e^{-\lfloor k/2 \rfloor} \le e^{-\lfloor \delta\mu/3 \rfloor}$.

(III) *For* $\delta \ge 1$ *and* $k = \lceil \delta\mu \rceil$, $\quad Pr(|X - \mu| \ge \delta\mu) \le G(\mu, \delta) \le e^{\frac{-\delta \ln(1+\delta)\mu}{2}}$.

Proof. (I). We apply Theorem 8.III with $C = \mu$, and $k = \left\lfloor \delta^2\mu/e^{1/3} \right\rfloor$, which is permissible since $\mu \ge k$ and $\mu \ge \sigma^2[X]$ for variables in the range $[0,1]$. If $\left\lfloor \delta^2\mu/e^{1/3} \right\rfloor$ is even, this gives a bound of

$$Pr(|X - \mu| \ge \delta\mu) \le \left(\frac{k}{e^{2/3}\delta^2\mu}\right)^{k/2} \le e^{-k/2} \le e^{-\lfloor \delta^2\mu/3 \rfloor}, \text{ since } 2e^{1/3} < 3.$$

If $\left\lfloor \delta^2\mu/e^{1/3} \right\rfloor$ is odd, we follow a calculation similar to that above, but use independence $k = \left\lfloor \delta^2\mu/e^{1/3} \right\rfloor - 1$. This gives

$$Pr(|X - \mu| \ge \delta\mu) \le \left(\frac{k}{e^{2/3}\delta^2\mu}\right)^{k/2} \le \left(\frac{k}{e(k+1)}\right)^{k/2} \le e^{-k/2}e^{-k/2(k+1)}.$$

Since $k \ge 2$, $e^{-k/2(k+1)} \le e^{-1/3}$ and hence,

$$Pr(|X - \mu| \ge \delta\mu) \le e^{-(k/2+1/3)} \le e^{-\lfloor \delta^2\mu/3 \rfloor}.$$

In part (II) we follow the same iteration as in part (I), but set $C = \delta\mu$ and $k = \lfloor \delta\mu/e^{1/3} \rfloor$ or $\lfloor \delta\mu/e^{1/3} \rfloor - 1$, depending on the parity of $\lfloor \delta\mu/e^{1/3} \rfloor$. In part (III) we reiterate the result of Theorem 6, combined with Theorem 7. □

Remark. The proofs of parts (I) and (III) of Theorem 9 also point out the relative merits of the basic method (Sections 3.1.1 and 3.1.2) versus its redirection of this subsection. The basic method of using non–negative linear combinations of the symmetric polynomials S_i gives better probability bounds when δ, the relative deviation from the mean, is greater than 1: it yields the probability bound of $exp(-\Theta(\delta \ln(1+\delta)\mu))$ in this case. On the other hand, the formulation involving the kth moment inequality gives a much smaller bound on the amount of independence needed, when $\delta < 1$.

3.1.4 Probability Bounds for Exactly r Successes under Limited Independence

Some applications require estimates for the probability that **exactly** r successes occur in cases where the occurrence of at least r successes is not too improbable. The following theorem shows how and when this can be done. It also provides relative errors, which can be useful for estimating conditional probabilities.

Theorem 10 *Let $X_1, X_2, \ldots, X_n$ and $Y_1, Y_2, \ldots, Y_n$ be Bernoulli trials with probabilities of success* $\mathrm{E}[X_i] = \mathrm{E}[Y_i] = p_i$. *We let the Y_i's be independent, but only require the X_i's to be k-wise independent. Let $p(r) = Pr(\sum_i Y_i = r)$, and $p_k(r) = Pr_k(X = r)$, where the subscript k denotes the k-wise independent trials. Let $P(r) = \sum_{\ell \geq r} p(\ell)$, and $P_k(r) = \sum_{\ell \geq r} p_k(\ell)$.*

(I) If $r \leq k$, then $$|p_k(r) - p(r)| \leq \binom{n}{k}\binom{k}{r}\left(\frac{\mu}{n}\right)^k.$$

(II) If $k \geq e\mu + \ln(1/p(0)) + r + D$, then $$|p_k(r) - p(r)| \leq e^{-D} p(r).$$

(III) If $k \geq e\mu + \ln(1/p(0)) + r + D$, then $$|P_k(r) - P(r)| \leq (1 - P(r))e^{-D}.$$

(IV) If $r \geq (1+\delta)\mu + k$, then $$P_k(r), P(r) \leq (1+\delta)^{-k},$$
and hence $$|P_k(r) - P(r)| \leq (1+\delta)^{-k}.$$

Although (IV) holds for all values of k it is meant to be used for $k \ll \lceil \delta\mu \rceil$, indeed:

(V) If $r \geq (1+\delta)\mu + k$ and $k \geq \lceil \delta\mu \rceil$, then $$P_k(r), P(r) \leq G(\mu, \delta),$$
and hence $$|P_k(r) - P(r)| \leq G(\mu, \delta).$$

Proof. For an arbitrary event A, we may use standard inclusion-exclusion to estimate the probability of the event $\left[A \wedge [\bigwedge_{\ell \notin \{i_1, \ldots, i_r\}} (X_\ell = 0)]\right]$. The probability $p(r)$ can be expressed in terms of events $A = [\bigwedge_{j \in \{i_1, \ldots, i_r\}} (X_j = 1)]$, which admits a simple estimation as follows.

$$\begin{aligned}
p_k(r) &= \sum_{i_1<i_2<\cdots<i_r} Pr_k\left(\left(\bigwedge_{j\in\{i_1,\ldots,i_r\}}(X_j=1)\right)\wedge\left(\bigwedge_{\ell\notin\{i_1,\ldots,i_r\}}(X_\ell=0)\right)\right)\\
&= \sum_{i_1<i_2<\cdots<i_r}\sum_{l=0}^{n-r}\sum_{\substack{i_{r+1}<\cdots<i_{r+l}\\ i_{r+1},\ldots,i_{r+l}\notin\{i_1,\ldots,i_r\}}}(-1)^l Pr_k\left(\bigwedge_{j\in\{i_1,\ldots,i_{r+l}\}}(X_j=1)\right)\\
&= \sum_{l=0}^{n-r}\sum_{i_1<\cdots<i_{r+l}}(-1)^l\binom{r+l}{r}Pr_k\left(\bigwedge_{j\in\{i_1,\ldots,i_{r+l}\}}(X_j=1)\right).
\end{aligned}$$

Truncating the outer summation at $l = k-r$ introduces an error that is bounded by the last term of the truncated sum. Let $p_k^T(r)$ and $p^T(r)$ denote these truncated sums, in the respective cases of k-wise and full independence. Since the first $k-r$ terms in the outer summation are the same for both fully and k-wise independent random variables, $p_k^T(r) = p^T(r)$. Furthermore, $Pr_k\left(\bigwedge_{j\in\{i_1,\ldots,i_k\}}(X_j=1)\right) = \prod_{j=1}^k p_{i_j}$. Hence,

$$p_k(r) = p^T(r) - (-1)^{k-r}\delta_k\binom{k}{r}\sum_{i_1<\ldots<i_k}\prod_{j=1}^k p_{i_j}$$

for some $\delta_k \in [0,1]$, and an identical inequality holds without the k subscripts. Hence,

$$|p_k(r) - p(r)| \le \binom{k}{r}\sum_{i_1<\ldots<i_k}\prod_{j=1}^k p_{i_j}. \tag{3.11}$$

$\sum_{i_1<\ldots<i_k}\prod_{j=1}^k p_{i_j}$ is maximized when all p_{i_j} are equal (Lemma 7), and hence,

$$|p_k(r) - p(r)| \le \binom{k}{r}\binom{n}{k}\frac{(p_1+p_2+\ldots+p_n)^k}{n^k} = \binom{n}{k}\binom{k}{r}(\mu/n)^k,$$

and (I) now follows.

To get multiplicative error bounds, let $Q_r = \sum_{i_1<i_2<\cdots<i_r}\prod_{\ell=1}^r\left(\frac{p_{i_\ell}}{1-p_{i_\ell}}\right)$, and define the summation in the error estimate of equation (3.11) by $R_k = \sum_{i_1<\ldots<i_k}\prod_{j=1}^k p_{i_j}$. Observe that R_k is the expected number of size k sets of successes among n trials,

so that z successes total accounts for $\binom{z}{k}$ such sets. In the fully independent case, $p(r) = p(0) \sum_{i_1<i_2<\cdots<i_r} \prod_{\ell=1}^{r} \left(\frac{p_{i_\ell}}{1-p_{i_\ell}}\right) = p(0)Q_r$. Furthermore, $R_r \leq Q_r$, and $\binom{k}{r} R_k \leq R_r \times R_{k-r}$. It follows that

$$|p_k(r) - p(r)| \leq \binom{k}{r} R_k \leq R_r \times R_{k-r} \leq Q_r R_{k-r} \leq \frac{p(r)\mu^{k-r}}{p(0)(k-r)!},$$

since $R_{k-r} \leq \binom{n}{k-r}\left(\frac{\mu}{n}\right)^{k-r} \leq \frac{\mu^{k-r}}{(k-r)!}$.

For $k \geq e\mu - \log(p(0)) + r + D$, the factor $\frac{(\mu)^{k-r}}{p(0)(k-r)!}$ is bounded by

$$\left(\frac{k + \log(p(0)) - D - r}{k-r}\right)^{k-r} e^{-\log(p(0))} \leq e^{-D},$$

which establishes (II).

Part (III) is immediate, since $P_k(r) = 1 - \sum_{\ell<r} p_k(\ell)$ and hence,

$$|P_k(r) - P(r)| \leq \sum_{\ell<r} |p_k(\ell) - p(\ell)| \leq \sum_{\ell<r} p(\ell)e^{-D} = (1 - P(r))e^{-D}.$$

Finally, suppose that $r = (1+\delta)\mu + k$. Then by Lemma 8,

$$\begin{aligned} P_k(r) &\leq \frac{\mu^k}{k!\binom{r}{k}} \leq \frac{\mu^k}{r(r-1)(r-2)\ldots(r-k+1)} \\ &\leq (1+\delta)^{-k}. \end{aligned} \tag{3.12}$$

Part (IV) is completed by observing that (3.12) also holds with $P(r)$ substituted for $P_k(r)$. Part (V) is an immediate consequence of Theorem 6. This concludes the proof of Theorem 10. □

It is worth pointing out that parts (I) through (III) of Theorem 10 are not strong when $\mu > \sqrt{n}$, since it follows from the work of Linial & Nisan [71] that $Pr(X = \ell) = Pr(Y = \ell)(1 + O(e^{-2(k-\ell)/\sqrt{n}}))$ independently of μ, which gives a much sharper bound in this case. Also, independent of our work, a result similar to part(I) of Theorem 10 has been proven by Even, Goldreich, Luby, Nisan & Veličković [41]: they show that $|p_k(0) - p(0)| \leq 2^{-\Omega(k)}$.

Theorem 10, in fact, achieves its greatest strength when μ is small, say $\mu = o(n)$, or even $\mu = O(1)$. Such instances are not unusual when pseudorandom integers are being generated uniformly in the range $[0, n]$ and a successful trial corresponds to just a few different values. This is precisely the usual circumstance in, for instance, hashing [110,136]. As an example, consider the (uniformly distributed) random placement of n balls among n slots. The expected number of items in slot 1 is just 1. The probability $p(0)$ that no item lands in a given slot is about $\frac{1}{e}$. Theorem 10 shows that if the independence k is $e + 1 + r + D$, the probability that exactly r items land in that slot is the same in the k-wise independent case as in the fully independent case, up to a multiplicative factor of $(1 + e^{-D})$ or less. Suppose that, during the placements of balls l through m, exactly r balls fall into slot j for $1 \leq l \leq m < n$, $r \leq m - l + 1$. Let $\chi^r_{[l,m]}$ denote this event (with the dependence upon j understood). The conditional probability that, under k-wise independence, ball $m + 1$ also falls into slot j is $Pr_k(\chi^1_{[m+1,m+1]} | \chi^r_{[l,m]})$. If we use one degree of freedom for the $m + 1$-st ball, it will be uniformly distributed while the previous m balls will enjoy $(k - 1)$-wise independence. We may estimate the conditional probability as $Pr_k(\chi^r_{[l,m]} | \chi^1_{[m+1,m+1]}) \times \frac{Pr(\chi^1_{[m+1,m+1]})}{Pr_{k-1}(\chi^r_{[l,m]})}$. Since both $Pr_{k-1}(\chi^r_{[l,m]})$ and $Pr_k(\chi^r_{[l,m]} | \chi^1_{[m+1,m+1]})$ can be estimated by $Pr(\chi^r_{[l,m]})(1 \pm err_{k-1})$ where err_{k-1} is the relative error that results from the limited independence, we see that $Pr_k(\chi^1_{[m+1,m+1]} | \chi^r_{[l,m]})$ is very close to $1/n$, with a relative error that is approximately $2err_{k-1}$. For $k \geq 1 + e + 1 + r + D$, the resulting accuracy is about $1 \pm 2e^{-D}$. Thus even with modest independence, this process behaves "as expected" much of the time; that is, the corresponding conditional probabilities for k-wise independence are very close to the ones for full independence, in many cases.

3.1.5 How close to optimal are our results?

It is known that the standard Chernoff–Hoeffding bounds are optimal in general to within a constant factor in the exponent, since we know by the Central Limit

Theorem that as $n \to \infty$, the tail of the scaled sum of i.i.d. r.v.'s tends to the tail of a normal distribution, and hence we cannot significantly improve the tail probabilities presented by Theorem 9. However, what about the independence we get? Can it be reduced further to get the same tail probabilities?

To answer this, we note that the tail probabilities presented by Theorem 9 for k–wise independent r.v.'s are of the form $e^{-c \cdot k}$. However, n k–wise independent r.v.'s require a sample space of size at least

$$\sum_{i=0}^{\lfloor k/2 \rfloor} \binom{n}{i} \approx (O(n/k))^{\lfloor k/2 \rfloor},$$

as shown for binary unbiased r.v.'s by Chor, Goldreich, Håstad, Friedman, Rudich & Smolensky [29] and for general r.v.'s by Alon, Babai & Itai [5]. Noting next that any nonzero probability in a sample space of size t is at least $1/t$, we see that to get a tail probability of the form $e^{-c \cdot k}$, we need at least $\Omega(\frac{k}{\log(n/k)})$–wise independence. Thus, the independence we get cannot in general be reduced by more than a factor of $O(\log n)$.

However, by using results from the newly developing theory of *approximating probability distributions* (Naor & Naor [85], Azar, Motwani & Naor [11], Alon, Goldreich, Håstad & Peralta [6], Even, Goldreich, Luby, Nisan & Veličković [41] and Chari, Rohatgi & Srinivasan [25]), we get optimal results in the case where the X_i's are binary with $Pr(X_i = 1) = 1/2$. A sample space X for n–bit vectors was defined to be k–wise ϵ–biased by Naor & Naor [85] (see also Vazirani [132]) if

$$\forall S \subseteq \{1, 2, \ldots, n\}, 1 \leq |S| \leq k, |Pr(\bigoplus_{i \in S} x_i = 1) - Pr(\bigoplus_{i \in S} x_i = 0)| \leq \epsilon,$$

where $\bigoplus$ denotes the XOR function and x_i denotes the ith bit of an n–bit string x picked uniformly at random from X. One property of such a sample space is that $\forall \ell, \ell = 1, 2, \ldots, k, \forall \{i_1, i_2, \ldots i_\ell\} \subseteq \{1, 2, \ldots, n\}, \forall b_1 b_2 \ldots b_\ell \in \{0, 1\}^\ell$,

$$|Pr(x_{i_1} = b_1, x_{i_2} = b_2, \ldots, x_{i_\ell} = b_\ell) - \frac{1}{2^\ell}| \leq \epsilon. \tag{3.13}$$

X is ϵ–biased if it is n–wise ϵ–biased. Constructions of k–wise ϵ–biased sources of size $poly(k, \log n, \frac{1}{\epsilon})$ were presented in [85,6]. Such sample spaces have been shown to have several applications to explicit constructions and to derandomization, mainly since probabilistic analyses may be expected to be robust under small perturbations of the probabilities. Now, it is easy to see how our methods can be used to derive large deviation bounds for $x_1 + x_2 + \cdots + x_n$; from inequality (3.13), it follows that for a k–wise ϵ–biased source X,

$$\forall \ell \leq k \;\; E[S_\ell(x_1, x_2, \ldots, x_n)] \leq \binom{n}{\ell} \cdot \left(\frac{1}{2^\ell} + \epsilon\right)$$

and hence by picking $\epsilon \leq \frac{1}{2^\ell}$, this quantity can at most be double its unbiased value of $\binom{n}{\ell} \cdot \frac{1}{2^\ell}$. Thus, for a k–wise ϵ–biased random source with $k = h(n, n/2, \delta) = n\delta$ and with $\epsilon = 2^{-k}$,

$$Pr(\sum_{i=1}^{n} x_i \geq \frac{n}{2}(1+\delta)) \leq 2 \cdot U_2(n, n/2, \delta). \tag{3.14}$$

Since such a source can be generated using $poly(k, \log n)$ random bits, we see that this result is optimal as long as $k = \Omega(\log n)$; if $k = O(\log n)$, then the probability space is polylogarithmic in size and should in most cases be dispensable, via brute-force search of the sample space. Similar results hold when the X_i's are binary with their probabilities of being one being the same negative power of two (not necessarily $1/2$), using identical methods.

3.1.6 Upper Tail Bounds for some other Distributions

Suppose we have random bits $X_1, X_2, \ldots, X_n$ with some arbitrary distribution. Let $X = \sum_{i=1}^{n} X_i$, and let $\mu = E[X]$. Then, for any $a > \mu$, the methods of Section 3.1.1 yield

$$Pr(X \geq a) \leq \frac{\sum_{i=0}^{a} y_i S_i(X_1, X_2, \ldots, X_n)}{\sum_{i=0}^{a} y_i \binom{a}{i}}, \;\; \forall (y_0, y_1, \ldots y_a) \in \Re_+^{a+1}.$$

The following theorem is immediate.

Theorem 11 *Given n random bits $X_1, X_2, \ldots, X_n$ with $X = \sum_{i=1}^{n} X_i$ and $\mu = E[X]$, suppose z_j is an upper bound on $E[S_j(X_1, X_2, \ldots, X_n)]$, $j = 1, 2, \ldots, n$. Then, if $a = \mu(1+\delta)$ for $\delta > 0$,*

- $Pr(X \geq a) \leq \dfrac{\sum_{i=0}^{a} y_i z_i}{\sum_{i=0}^{a} y_i \binom{a}{i}}, \ \forall (y_0, y_1, \ldots y_a) \in \Re_{+}^{a+1}.$
- *If $X_1, X_2, \ldots, X_n$ are k–wise independent, then $Pr(X \geq a) \leq min_{i=1,2,\ldots,k} \dfrac{z_k}{\binom{a}{k}}$.*

As an example of a distribution which benefits from the above, recall the λ-*correlated random variables* defined and used in Chapter 2: random bits $X_1, \ldots, X_n$ are defined to be λ–correlated if for all j and for all distinct indices $X_{i_1}, X_{i_2}, \ldots, X_{i_j}$, $E[\Pi_{\ell=1}^{j} X_{i_\ell}] \leq \lambda \Pi_{\ell=1}^{j} E[X_{i_\ell}]$; note that $z_j \leq \lambda \binom{n}{j} (\frac{\mu}{n})^j$ in this case. Hence, Theorem 11 directly implies one of the main theorems of Chapter 2, Theorem 1 which states that if $X_1, X_2, \ldots, X_n$ are λ–correlated random bits with $X = \sum_{i=1}^{n} X_i$ and $\mu = E[X]$, then for any $\delta > 0$, $Pr(X \geq \mu(1+\delta))$ is at most λ times any Chernoff–Hoeffding type upper bound on the corresponding probability had the X_i been independent, with the same individual distributions. Indeed, it was the work described in Chapter 2 which mainly motivated the methods of Section 3.1.1. Further, the applications sketched in Section 3.2.2 use Theorem 11.

Theorem 11 helps improve the known upper tail probability bounds for the `hypergeometric distribution`, an important source of λ–correlated random variables. Suppose n balls are picked at random `without replacement` from an urn containing M red balls and $N - M$ balls of other colors. Let X be the number of red balls picked in the random sample, and let $p \doteq M/N$. Then for $\delta > 0$, a special case of a result of Hoeffding [50] (see Chvátal [30] for another proof) implies that

$$Pr(X \geq np(1+\delta)) \leq F(n, np, \delta). \tag{3.15}$$

We prove the following strengthened version of inequality (3.15).

Lemma 10 *Suppose a random set of n balls is picked from an urn containing M red balls and $N - M$ balls of other colors. If X denotes the number of red balls picked, $p = M/N$, $\delta > 0$ and if $k \doteq h(n, np, \delta) = o(N)$, then*

$$Pr(X \geq np(1+\delta)) \leq U_2(n, np, \delta)e^{-\Theta(k^2/M)} \leq F(n, np, \delta)e^{-\Theta(k^2/M)}.$$

PROOF. (SKETCH) Number the balls picked as $1, 2, \ldots, n$, and let X_i be the indicator r.v. for the event that ball i was red. Then $X = \sum_{i=1}^{n} X_i$ and

$$Pr(X \geq a) \leq U_3(n, np, \delta) \doteq \frac{E[S_k(X_1, X_2, \ldots, X_n)]}{\binom{a}{k}},$$

where $a \doteq \lceil np(1+\delta) \rceil$. For distinct indices $i_1, i_2, \ldots, i_k$,

$$E[\Pi_{j=1}^{k} X_{i_j}] = Pr(\bigwedge_{j=1}^{k} X_{i_j} = 1) = \prod_{i=0}^{k-1} \frac{M-i}{N-i};$$

hence,

$$\begin{aligned} \frac{U_3(n, np, \delta)}{U_2(n, np, \delta)}7 &= (\frac{N}{M})^k \cdot \prod_{i=0}^{k-1} (\frac{M-i}{N-i}) \\ &= \prod_{i=1}^{k-1} \left(1 - (\frac{N}{M} - 1)\frac{i}{N-i}\right) \leq e^{-(\frac{N}{M}-1)\sum_{i=1}^{k-1} \frac{i}{N-i}} \end{aligned}$$

which is $e^{-\Theta(k^2/M)}$, if $k = o(N)$. □

Remark. Note that sampling without replacement produces r.v.'s which are 1–correlated. Lemma 10 gives good improvements over inequality (3.15) in many interesting cases, *e.g.*, consider the case $p =$ constant, $\delta =$ constant, and $\Omega(M^{0.5+\epsilon}) \leq n = o(N)$, for any fixed $\epsilon > 0$.

Also, the CH bounds [27,50,101,8] depend only on μ and not on the actual values of p_i, and give the upper bound $F(n, \mu, \delta) \geq U_2(n, \mu, \delta)$. We know for any $\delta > 0$ that $Pr(X \geq \mu(1+\delta)) \leq U_1(n, p_1, \ldots, p_n, \delta) = S_k(p_1, \ldots p_n)/\binom{\mu(1+\delta)}{k}$, where $k = h(n, \mu, \delta)$. By Lemma 7, this is maximized by setting $p_i = \mu/n \; \forall i$, subject to the constraint that $E[X] = \mu$. But if the values p_i are rather different, we get bounds formally superior to $U_2(n, \mu, \delta)$ and $F(n, \mu, \delta)$; suppose, for example, that $\mu = n/2$,

$p_i = \epsilon$, $i = 1, 2, \ldots, n/2$, and $p_i = 1 - \epsilon$, $i = n/2 + 1, \ldots, n$, where $0 < \epsilon \leq 1/2$. Then,

$$\frac{U_1(n, p_1, p_2, \ldots, p_n, \delta)}{U_2(n, \mu, \delta)} \leq f(\epsilon) \doteq 2^k \left(\sum_{i=0}^{k} \binom{n/2}{i} \binom{n/2}{k-i} \epsilon^i (1-\epsilon)^{k-i}\right) / \binom{n}{k}.$$

where $k = h(n, \mu, \delta)$. Note that $f(0) \leq e^{-\Theta(k^2/n)}$ by Lemma 10 and that $f(\epsilon)$ can get arbitrarily close to $f(0)$ since $f(\cdot)$ is continuous.

Remark. This particular result can only increase the constant factor in the exponent of $U_2(n, \mu, \delta)$. But, it is a small step towards better understanding of the dependence of $Pr(X \geq \mu(1+\delta))$ on $n, p_1, \ldots, p_n$, and δ. Similar improvements can also be made in the case of non–binary r.v.'s. An alternative approach might be to derive Chernoff-Hoeffding bounds for a sum of Bernoulli trials as a function of the variance as well as μ, a, and n, as in Siegel [113].

A final application is to the *semi-random* source introduced by Santha & Vazirani [107]. A random source which outputs bits $X_1, X_2, \ldots, X_n$ is defined to be ϵ-semirandom in [107] if

$$\forall i \;\; 1/2 - \epsilon \leq Pr(X_i = 1 | X_1, X_2, \ldots, X_{i-1}) \leq 1/2 + \epsilon,$$

i.e., the random bits can be correlated, but only to a limited extent, independent of the past history. Despite its seemingly weak nature, such a model has been shown to be able to simulate complexity classes such as *RP* (Vazirani & Vazirani [133]), and the study of a generalization of this model due to Zuckerman [140] has led to rich results recently (Nisan & Zuckerman [90], Wigderson & Zuckerman [138]). Noting that for such a source,

$$E[\prod_{j=1}^{k} X_{i_j}] \leq (1/2 + \epsilon)^k$$

for all $k \geq 1$ and for all distinct indices $i_1, i_2, \ldots, i_k$, we see that

$$Pr(\sum_{i=1}^{n} X_i \geq n(1/2 + \epsilon)(1 + \delta)) \leq U_2(n, n(1/2 + \epsilon), \delta), \;\; \forall \delta > 0,$$

for an ϵ-semirandom source.

Inequality (3.14) shows another application of our techniques.

3.2 Applications to Computation

The most striking point of Theorems 5 and 9 in our opinion is that bounds as good as the CH bounds can be obtained with small independence. This implies, for any analysis that relies on the CH bounds, much weaker requirements on the random sources used. We now present some further computational applications of the new results.

3.2.1 Reduced randomness for randomized algorithms

There are known constructions of r.v.'s with limited independence using a small number of random bits; for example, the construction of [54] and the use of universal hash functions [24] to generate $|F|$ many k–wise independent random elements from a finite field F using $O(k \log |F|)$ random bits, and the result of [5] using coding techniques [78], which gives a polynomial (in n) time algorithm to construct n k–wise independent and unbiased random bits, given $O(k \log n)$ independent unbiased bits for any k, $k \leq n$. Combining these with our result on reduced independence for the CH bounds, we get a major reduction in the amount of randomness needed for several randomized algorithms.

3.2.1.1 Reduced randomness for random sampling

In random sampling, we have a huge finite universe U and a subset $W \subseteq U$, and we want to estimate the fraction $f^* = |W|/|U|$. Given error parameters $\delta, \epsilon > 0$, the method used is to pick a random sample S of size $N(\delta, \epsilon)$ from U and output the fraction f of type W elements in S; $N(\delta, \epsilon)$ must be such that $Pr(|f^* - f| \geq \delta) \leq \epsilon$. This is required, for instance, in PAC learning [129] and in running BPP algorithms. What falls out of the CH bounds is that $N(\delta, \epsilon) = O(\frac{1}{\delta^2} \log(\frac{1}{\epsilon}))$ with all the samples being independent. We can improve this to

Theorem 12 *Given a universe U, a subset $W \subseteq U$, and error parameters $\delta, \epsilon > 0$, suppose a set S of $O(\frac{1}{\delta^2}\log(\frac{1}{\epsilon}))$ random samples with $O(\log(\frac{1}{\epsilon}))$-wise independence are picked from U. Then, if f^* and f are the respective fractions of type W elements in U and S, $Pr(|f^* - f| \geq \delta) \leq \epsilon$ will hold.*

PROOF. Consider a randomized algorithm which looks at a random set of samples S from U, and outputs the ratio f of type W elements in S. We now look at the random properties of S which are required for the claim $Pr(|f^* - f| \geq \delta) \leq \epsilon$ to hold.

Let $|S| = n$. In the notation of Theorem 9, we want to claim that $Pr(|X - E[X]| \geq \delta' E[X]) \leq \epsilon$, where $X = nf$ and $\delta' = \delta/f^*$. Parts (I) and (II) of Theorem 9 show that for a suitable of choice of independence k among the elements of S, $Pr(|X - E[X]| \geq \delta' E[X])$ can be bounded by $e^{-\lfloor k/2 \rfloor}$. Hence, it suffices to choose $k \geq 2\lceil \ln(\frac{1}{\epsilon}) \rceil$, for the above probability to be bounded by ϵ. To compute the required sample size n, we distinguish between the two cases $\delta' \leq 1$ and $\delta' > 1$. Then, from part(I) of Theorem 9, the probability that $|X - E[X]|$ is greater than $\delta' E[X]$ is bounded by ϵ, if the samples are $\lfloor \delta'^2 E[X]/e^{1/3} \rfloor$-wise independent and if $e^{-\lfloor \delta'^2 E[X]/3 \rfloor} \leq \epsilon$, *i.e.*, if $\delta'^2 E[X]/3 \geq \lceil \ln(\frac{1}{\epsilon}) \rceil$. This will hold if $n\delta^2/(3f^*) \geq \lceil \ln(\frac{1}{\epsilon}) \rceil$, *i.e.*, if $n \geq N_1(\delta, \epsilon) = \frac{3f^*}{\delta^2} \lceil \ln(\frac{1}{\epsilon}) \rceil$. If $\delta' > 1$, then a similar analysis using Theorem 9.II implies that $N_2(\delta, \epsilon) = \frac{3}{\delta} \lceil \ln(\frac{1}{\epsilon}) \rceil$ many samples with $2\lceil \ln(\frac{1}{\epsilon}) \rceil$-wise independence suffice to satisfy the error bounds.

Note that since both f^* and δ are clearly bounded by 1, the number of samples and independence needed in both the above cases can be upper bounded by $N_3(\delta, \epsilon) = \frac{3}{\delta^2} \lceil \ln(\frac{1}{\epsilon}) \rceil$ and $k^* = 2\lceil \ln(\frac{1}{\epsilon}) \rceil$. Note further that though by Theorem 9 the choice for the independence that minimizes the error bound is an increasing function of the sample size, increasing the sample space size when given a fixed independence will reduce the error probability; a proof of this claim follows. In the proof of Theorem 9, parts(I) and (II) were derived from part(III) of Theorem 8 with $C = E[X]$. Note that in the current problem, $C = nf^*$ and $T = n\delta$ where n is the number of random samples picked, in the notation of Theorem 8. When the independence k is fixed, the bound given by part(III) of Theorem 8 decreases with n for these values of C and T, and the claim follows. □

Theorem 8 also allows an estimate for the required size of a sample space with k–wise independent variables, in case $k < 3\lceil \ln(\frac{1}{\epsilon}) \rceil$.

Theorem 13 *Given a universe U, a subset $W \subseteq U$, and error parameters $\delta, \epsilon > 0$, suppose that S is a sample space of U whose elements are k-wise independent, for some even k. Then, if f^* and f are the respective fractions of of type W elements in U and S, then for $Pr(|f^* - f| \geq \delta) \leq \epsilon$ to hold, it is sufficient to choose $|S| \geq \frac{ck}{\delta^2 \epsilon^{(2/k)}}$, for some constant c.*

PROOF. Use part(III) of Theorem 8 by setting $C = n$, where $n = |S|$. □

Theorems 12 and 13 imply "reduced randomness" results for random sampling, if the universe U has some properties. For instance, if U is a finite field and if the field operations can be done in polynomial (in $\frac{1}{\delta}$ and $\log(\frac{1}{\epsilon})$) time, then any number of k–wise independent samples from U can be generated from k independent random samples [54,24].

We note, however, that the results of Theorem 12 are not optimal; see [14] for an optimal construction. The main advantages of our construction are its simplicity and direct parallelizability: it is not yet known how to parallelize other schemes for reducing randomness, *e.g.*, random walks on expanders (which is the method used in [14]). Via weaker bounds on the kth moment, essentially the same bounds for the random sampling problem have been obtained by Bellare & Rompel [15].

3.2.1.2 Reduced randomness for oblivious permutation routing

We now show how our results directly imply bounds that match the explicit constructions of algorithms with reduced randomness due to Peleg & Upfal [95], for *oblivious permutation routing on fixed interconnection networks* (see also [56,101, 104,128,130]).

Given some interconnection network with N nodes and a permutation $\sigma : \{1, \ldots, N\} \rightarrow \{1, \ldots, N\}$, the problem is to route a packet ν_i residing at each node i, to its destination $\sigma(i)$ so that the total time taken is small. Further, the routing must be *oblivious* in that the path $P_\sigma(x)$ chosen for a packet x must be "independent" of the path $P_\sigma(y)$ chosen for any other packet y (see [95] for a precise definition when randomized routing protocols are allowed). Explicit constructions of algorithms with a spectrum of time–randomness parameters are among the results proved in [95] for the degree–4 butterfly network; these are also extendible to other networks (see Karloff & Raghavan [56] for a protocol for the hypercube with slightly weaker bounds). Here, we show how our results of Section 3.1.1 directly imply the bounds of [95] for the hypercube; we believe that similar results should hold for other interconnection networks.

Consider the implementation of Valiant's two–phase scheme [128] (see also Valiant and Brebner [130]) on a hypercube with $N = 2^n$ nodes: (I) Each vertex i picks a random $\rho(i) \in \{1, 2, \ldots, N\}$ as an *intermediate destination* for ν_i, and routes ν_i there; (II) Each packet ν_i is routed to its final destination $\sigma(i)$. We now follow the discussion of the standard aspects of this from [101]. Assume FIFO queues at each edge, and that phase(I) routes ν_i from i to $\rho(i)$ by "correcting" its bits from left to right assuming that the nodes of the hypercube are indexed by n bits, and that phase(II) "corrects" bits right to left. So, phase(II) is like "running phase(I) backwards", and so we consider phase(I) alone here. It is shown in [101] that the time taken for packet ν_i in phase(I) is at most

$$n + \sum_{j=1}^{N} H_{ij}, \tag{3.16}$$

where $H_{ij} = 1$ if the paths $< i, \rho(i) >$ and $< j, \rho(j) >$ share an edge in phase(I), and 0 otherwise (recall that $n = \log_2 N$). It is also shown in [101] that if each $\rho(i)$ is uniformly distributed in $\{1, 2, \ldots, N\}$, then $\forall i$, $E[\sum_{j=1}^{N} H_{ij}] \leq n$. Here is the theorem that matches the explicit construction of [95].

Theorem 14 *There are explicit constructions of oblivious routing algorithms on the hypercube which, for any T, $c\log N \le T \le \sqrt{N}$ ($c > 4$ is a constant):*

1. *use $O(\frac{\log(N/Q)}{\log(T/\log N)}\log N)$ random bits and terminate in T steps with probability at least $1 - Q$ for any $0 < Q \le 1$;*
2. *use $O(\frac{\log^2 N}{\log(T/\log N)})$ random bits and terminate in expected time at most T.*

PROOF. (SKETCH.) Consider any packet ν_i; the probability that it takes more than $T/2$ steps in phase(I) is at most $Pr(\sum_{j=1}^{N} H_{ij} \ge T/2 - \log N)$. If the $\rho(i)$s are picked uniformly and in k-wise independent fashion, then the H_{ij} are $k-1$-way independent, while $E[\sum_{j=1}^{N} H_{ij}] \le \log N$, as before. It follows from our discussion of Section 3.1.1 and from Lemma 8 that, if $T > E[\sum_{j=1}^{N} H_{ij}]$, (*i.e.*, if $T > 4\log N$), then

$$Pr(\sum_{j=1}^{N} H_{ij} \ge T/2 - \log N) \le \frac{\binom{N}{k-1}(\frac{\log N}{N})^{k-1}}{\binom{T/2-\log N}{k-1}} \le \frac{(\log N)^{k-1}}{\Pi_{j=0}^{k-2}(T/2 - \log N - j)}.$$

By picking $k = \Theta(\frac{\log(N/Q)}{\log(T/\log N)})$, we can ensure that $Pr(\sum_{j=1}^{N} H_{ij} \ge T/2 - \log N) \le Q/(2N)$ holds. Arguing similarly for phase(II) and summing up over all i, we get (1) above.

For (2) above, we set $Q = 1/(2N)$ and replace T by $T-1$ in (1). Let T_{max} be a random variable denoting the time taken by the protocol, *i.e.*, the maximum, over all packets i, of the number of steps taken by packet ν_i to reach $\sigma(i)$. Note that $T_{max} \le \log N + N$, from the upper bound (3.16). Also,

$$Pr(T_{max} > (T-1)) \le Q,$$

from (1) above. Hence,

$$\begin{aligned} E[T_{max}] &\le (T-1)\cdot Pr(T_{max} \le (T-1)) + (\log N + N)\cdot Pr(T_{max} > (T-1)) \\ &\le (T-1)(1 - \frac{1}{2N}) + (\log N + N)\frac{1}{2N} \\ &\le T. \end{aligned}$$

Note that for any k, k–wise independent $\rho(i)$s can be generated from $k \log N$ random bits using hash functions [24], since the $\rho(i)$s can be thought of as belonging to the field $GF(2^n)$. Hence, we get bounds that match those of [95]. □

The above example typifies the type of application we expect our methods to find, *i.e.*, as direct "plug–in"s in analyses where the CH bounds are normally used.

3.2.2 The New Formulation and the Method of Conditional Probabilities

The `method of conditional probabilities` [100,117] is an important technique for the derandomization of algorithms; the reader is referred to [101] for details. We now show how this method can be combined with the formulation of Section 3.1.6. This will enable us to derive simple and efficient deterministic polynomial–time algorithms from randomized algorithms which can be analyzed using our formulation, *in a unified way.*

Given n random bits $X_1, X_2, \ldots, X_n$, can the conditional expectation

$$E[S_k(X_1, X_2, \ldots, X_n) | X_1 = b_1, X_2 = b_2, \ldots, X_i = b_i]$$

be evaluated, or at least be given a "reasonable" upper bound, for any k, any i, $i = 0, 1, 2, \ldots, n-1$, and any $b_1 b_2 \ldots b_i \in \{0,1\}^i$? If the X_i's are identically distributed, then it is reasonable to assume that an upper bound U_ℓ on $E[\Pi_{r=1}^{\ell} X_{j_r} | X_1 = b_1, X_2 = b_2, \ldots, X_i = b_i]$ is known for all ℓ, $\ell = 1, 2, \ldots, n-i$, and for all distinct indices $X_{j_1}, X_{j_2}, \ldots, X_{j_\ell} \in \{i+1, i+2, \ldots, n\}$; this is sufficient for the two applications shown below. Then, if

$$|\{j \mid (1 \le j \le i) \ \wedge \ (b_j = 1)\}| = i_1,$$

we can see that

$$E[S_k(X_1, X_2, \ldots, X_n) | X_1 = b_1, X_2 = b_2, \ldots X_i = b_i] \le \sum_{r=0}^{min(i_1,k)} \binom{i_1}{r} \binom{n-i}{k-r} U_{k-r}. \tag{3.17}$$

We now present two applications where the combination of our formulation and the method of conditional probabilities leads to fast polynomial–time algorithms, via upper bound(3.17). The first application, to jobshop scheduling, is a "natural" derandomization of the randomized algorithm of [112], faster than the derandomization techniques of [112] and [97]; this is shown in Section 3.2.2.1. The second application is to discrepancy theory, and is discussed in Section 3.2.2.2. The "usual" method of conditional probabilities for these problems frequently calls for independence among the random variables corresponding to the bits $X_1, X_2, \ldots, X_n$ seen above; this is not the case for these two problems and in general for many other problems.

3.2.2.1 Improved algorithms for packet routing and jobshop scheduling

We now present simpler approximation algorithms for `packet routing` (Leighton, Maggs & Rao [69]) and `jobshop scheduling` (Shmoys, Stein & Wein [112]) which provide improved approximation guarantees, by using ideas from above. The *non-preemptive jobshop scheduling* problem is as follows: given n jobs, m machines and a sequence of `operations` for each job where each operation is assigned to a specific machine, construct a schedule to run the jobs so that the time taken to process all the jobs is minimized, subject to: (i) the operations of each job must be done in sequence; (ii) no operation of any job running on any machine can be preempted till it is completed, and (iii) a machine can process at most one operation at a time. One of the results of [69] tackles a special case of this problem; the general case is handled in [112]. Both these papers give polynomial–time algorithms to produce good approximations to an optimal schedule.

Let P_i be the total time needed for job J_i, and let $P_{max} = max_{i \in [1,n]} P_i$. Let Π_j be the total time for which machine M_j is needed, and let $\Pi_{max} = max_{i \in [1,m]} \Pi_i$. Before an actual schedule is constructed in [112], a pseudo–schedule $\mathcal{S}$ is constructed which temporarily assumes that each machine can work on upto D operations simultaneously, where $D > 1$ depends on the input instance. The pseudo–schedule is later used to construct an actual schedule. The only step where randomization is used in [112] is during the construction of the pseudo–schedule and is the following.

An initial random delay $d_i \in \{1, 2, \ldots, \Pi_{max}\}$ is assigned for each job J_i. Suppose that the sequence of operations of job J_i are $O_{i,1}, \ldots O_{i,r_i}$, and that operation $O_{i,r}$ takes time $t_{i,r}$; then, in the pseudo–schedule $\mathcal{S}$, job J_i is scheduled to start at time d_i and runs to completion without interruption, *i.e.*, operation $O_{i,r}$ starts at time $d_i + \sum_{\ell=1}^{r-1} t_{i,\ell}$. We denote the offset $\sum_{\ell=1}^{r-1} t_{i,\ell}$ by $\tau(O_{i,r})$. As shown in [69,112], if the d_i's are generated uniformly and independently, then with high probability, every machine at every unit of time will have (a `congestion` of) at most $D(n, m_{max}) \doteq c \cdot \frac{\log(n \cdot m_{max})}{\log\log(n \cdot m_{max})}$ jobs scheduled on it for some constant c, where m_{max} is the maximum number of operations in any job. This step is then derandomized to deterministically compute initial delays leading to a congestion bound of $O(\log(n \cdot m_{max}))$. Linear programming is used for the derandomization, making this step the bottleneck. This step is sped up in [97,119]. Here, we get a better congestion bound of $D(n, m_{max})$ as opposed to the previously known $O(\log(n \cdot m_{max}))$ bound, with an algorithm which is more direct than the ones of [97,119], while having time complexities comparable to theirs.

We assign random initial delays $\{d_i \in \{1, 2, \ldots, \Pi_{max}\}\}$ uniformly and independently to the jobs. Suppose that the operations scheduled on machine M_i are $O_1, \ldots O_{m_i}$, which respectively belong to jobs $J_{i_1}, J_{i_2}, \ldots, J_{i_{m_i}}$ and take $t_1, \ldots, t_{m_i}$ units of time. For any machine M_i and time instance $t \in \{1, 2, \ldots, \Pi_{max} + P_{max}\}$, we define $\sum_{\ell \in [1, m_i]} t_\ell = \Pi_i$ many indicator r.v.'s $X_j^i(t)$, $j = 1, 2, \ldots, \Pi_i$, to analyze the congestion on machine M_i at time t in $\mathcal{S}$; each of these r.v.'s is an indicator for the event that a particular unit of time of some operation gets scheduled on M_i at time t in $\mathcal{S}$, as follows. The index j encodes the time unit and operation: let $j_r = \sum_{\ell=1}^{r-1} t_\ell$ for $r = 1, 2, \ldots, m_i$; then if $j_r < j \leq j_{r+1}$, j represents the $j - j_r$th time unit of O_r

as follows.

$$X^i_j(t) = \begin{cases} 1 & \text{if } j = j_r + p,\ 1 \le p \le t_r, \text{ and the } p\text{th time unit of } O_r \text{ is scheduled for} \\ & \text{time } t, \textit{ i.e.}, \text{ if } d_{i_r} + \tau(O_r) + p - 1 = t; \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to see that $E[X^i_j(t)] \le \frac{1}{\Pi_{max}}$, and that for any positive integer k, the probability that machine M_i has congestion at least k at time unit t is

$$Pr(\sum_{r=1}^{\Pi_i} X^i_r(t) \ge k) \le E[S_k(X^i_1(t), \ldots, X^i_{\Pi_i}(t)) \le \binom{\Pi_i}{k} \left(\frac{1}{\Pi_{max}}\right)^k.$$

In addition, if $\sum_{r=1}^{\Pi_i} X^i_r(t) \ge k$ holds for some time t, then $\sum_{r=1}^{\Pi_i} X^i_r(t') \ge k$ also holds for some time t', where t' is one of the starting times of the operations scheduled on M_i. Further, the starting time of each operation O_r is uniformly distributed in $[\tau(O_r), \Pi_{max} - 1 + \tau(O_r)]$. Hence, for any k, Pr(some machine has congestion at least k at some time instance) is at most

$$\sum_{i=1}^{m} \sum_{r=1}^{m_i} \sum_{t=\tau(O_r)}^{\Pi_{max}-1+\tau(O_r)} \frac{1}{\Pi_{max}} E[S_{k-1}(X^i_1(t), \ldots, X^i_{\Pi_i}(t))], \tag{3.18}$$

which is at most

$$\sum_{i=1}^{m} m_i \frac{\Pi_{max}}{\Pi_{max}} \binom{\Pi_i}{k-1} \left(\frac{1}{\Pi_{max}}\right)^{k-1} \le \sum_{i=1}^{m} \frac{m_i}{(k-1)!}.$$

Clearly $\sum_{i=1}^{m} m_i \le n \cdot m_{max}$, hence for $k - 1 > k^* = c_1 \frac{\log(n \cdot m_{max})}{\log\log(n \cdot m_{max})}$ for some suitable constant c_1, the above probability estimate is less than one. We may now use the above form as a `pessimistic estimator` [100] to deterministically set the delays d_i for the jobs one–by–one by the method of conditional probabilities [100,117], to achieve the congestion bound of $D(n, m_{max})$.

Assume inductively that initial delays $d_1 = d^*_1, d_2 = d^*_2, \ldots, d_s = d^*_s$ have been set deterministically for the jobs $J_1, J_2, \ldots, J_s$; the aim is to compute d^*_{s+1} now. Consider any machine M_i on which J_{s+1} has at least one operation; let these operations be $A_{s+1,1}, A_{s+1,2}, \ldots, A_{s+1,a_i}$. Let $B_{s,1}, B_{s,2}, \ldots, B_{s,b_i}$ be the operations which belong to some job in $\{J_1, J_2, \ldots, J_s\}$ and which are scheduled on M_i, and let $t_{s,1}, t_{s,2}, \ldots, t_{s,b_i}$ be the times at which they are scheduled to start on M_i; these times are known, since we know the values of $d_1, d_2, \ldots, d_s$. Let $O_1, O_2, \ldots, O_{c_i}$ be the operations on machine M_i that belong to jobs from the set $\{J_{s+2}, J_{s+3}, \ldots, J_n\}$. We define, for any $t \in \{1, 2, \ldots, P_{max} + \Pi_{max}\}$ and $r \in \{1, 2, \ldots, \Pi_{max}\}$,

$$g(s+1, i, t, r) \doteq E[S_{k^*}(X_1^i(t), X_2^i(t), \ldots, X_{\Pi_i}^i(t)) | d_1 = d_1^*, \ldots, d_s = d_s^*, d_{s+1} = r],$$

and $num(i, t, < x_1, x_2, \ldots, x_j >)$ to be the number of operations from jobs $J_1, \ldots, J_j$ that are scheduled on machine M_i at time t, given that $d_1 = x_1, \ldots, d_j = x_j$.

When conditioned on the event $d_1 = d_1^*, d_2 = d_2^*, \ldots, d_s = d_s^*, d_{s+1} = \Delta$ for any $\Delta \in \{1, 2, \ldots, \Pi_{max}\}$, upper bound(3.18) becomes

$$\begin{aligned} f(s+1, \Delta) \doteq \sum_{i=1}^{m} \quad & (\sum_{j=1}^{a_i} g(s+1, i, \tau(A_{s+1,j}) + \Delta, \Delta) + \\ & \sum_{j=1}^{b_i} g(s+1, i, t_{s,j}, \Delta) + \\ & \sum_{j=1}^{c_i} \sum_{t=\tau(O_j)}^{\tau(O_j)+\Pi_{max}-1} \frac{1}{\Pi_{max}} g(s+1, i, t, \Delta)). \end{aligned}$$

Recall that the method of conditional probabilities will set $d_{s+1} = d_{s+1}^*$, where d_{s+1}^* is the index at which $f(s+1, \cdot)$ is minimized. Note that $f(s+1, \Delta)$ can be readily computed for any Δ and hence, so can d_{s+1}^*. To make the computation more efficient, we use the following observations.

1. Suppose we need to compute $f(s+1, \Delta)$ for some Δ, using upper bound (3.17). Then, for any machine M_j which has some operation from job J_{s+1}, the term "$n - i$" in upper bound (3.17) corresponds to the number of operations of jobs $J_1, J_2, \ldots, J_{s+1}$ on machine M_j, *i.e.*, $a_j + b_j$. Hence, for any $t \in \{1, 2, \ldots, P_{max} + \Pi_{max}\}$, $g(s+1, j, t, \Delta)$ can be computed in $O(k^*)$ time if $num(j, t, < d_1^*, d_2^*, \ldots, d_s^*, \Delta >)$ is known, since upper bound(3.17) involves a sum over at most k^* terms (recall that k^* is an upper bound on the number of operations scheduled at the same time).

2. Suppose inductively that $num(i, t, < d_1^*, d_2^*, \ldots, d_s^* >)$ is known, for all machines M_i and for all times t. Let ω_{s+1} be the number of operations of job J_{s+1}. We will consider only those machines that have some operation of J_{s+1}; the number of such machines is clearly at most ω_{s+1}. Hence, given the $num(i, t, < d_1^*, d_2^*, \ldots, d_s^* >)$ values, the $num(\cdot, \cdot, < d_1^*, d_2^*, \ldots, d_s^*, 1 >)$ values need to be updated for at most ω_{s+1} machines and for $P_{max} + \Pi_{max}$ time units, and can be done in $O(\omega_{s+1}(P_{max} + \Pi_{max}))$ time. Given the $num(\cdot, \cdot, < d_1^*, d_2^*, \ldots, d_s^*, 1 >)$ values, computation of $f(s+1, 1)$ takes $O(\omega_{s+1} k^*(P_{max} + \Pi_{max}))$ time, since $g(\cdot, \cdot, 1)$ can be computed in $O(k^*)$ time, given these values.

3. Suppose we have computed the $num(\cdot, \cdot, < d_1^*, d_2^*, \ldots, d_s^*, \Delta >)$ values and $f(s+1, \Delta)$ for some value Δ, and that we need to compute $f(s+1, \Delta+1)$. We can proceed to first compute the $num(\cdot, \cdot, < d_1^*, d_2^*, \ldots, d_s^*, \Delta+1 >)$ values and then $f(s+1, \Delta+1)$, as follows. Suppose some operation α of J_{s+1} is scheduled to run on some machine M_i from time t_1 to time t_2, when we set $d_{s+1} = \Delta$. Then, note that

   $$num(i, t', < d_1^*, \ldots, d_s^*, \Delta+1 >) = num(i, t', < d_1^*, \ldots, d_s^*, \Delta >), \forall t' \in [t_1+1, t_2].$$

   Hence, α leaves at most two of the $num(\cdot, \cdot, < d_1^*, d_2^*, \ldots, d_s^*, \Delta+1 >)$ values different from the corresponding $num(\cdot, \cdot, < d_1^*, \ldots, d_s^*, \Delta >)$ values. Hence, $num(\cdot, \cdot, < d_1^*, d_2^*, \ldots, d_s^*, \Delta+1 >)$ can be updated in $O(\omega_{s+1})$ time. It now follows from arguments similar to those used above that $f(s+1, \Delta+1)$ can be computed in $O(k^* \omega_{s+1})$ time.

The above observations imply an efficient algorithm to compute d_{s+1}: inductively maintain the $num(\cdot, \cdot, < d_1^*, d_2^*, \ldots, d_s^*, r >)$ values as r goes from $1, 2, \ldots,$ to Π_{max}, and compute $f(s+1, 1), f(s+1, 2), \ldots, f(s+1, \Pi_{max})$ in that order by sequentially updating the corresponding num values. Since computing d_{s+1}^* takes $O(\omega_{s+1} k^*(P_{max} + \Pi_{max}))$ time, the total time complexity is $O(\omega k^*(P_{max} + \Pi_{max}))$, where ω is the total number of operations. Hence, we have

Theorem 15 *Initial delays* $\{d_i : 1 \leq i \leq n\}$ *in the range* $\{1, 2, \ldots, \Pi_{max}\}$ *for each job* J_i *can be set in* $O((P_{max}+\Pi_{max})\omega\frac{\log(n \cdot m_{max})}{\log\log(n \cdot m_{max})})$ *time where* ω *is the total number of operations, such that in the (infeasible) schedule in which every job* J_i *starts at time* d_i *and runs without interruption, every machine has at most* $O(\frac{\log(n \cdot m_{max})}{\log\log(n \cdot m_{max})})$ *jobs scheduled on it at any time.*

We feel that the above is a natural derandomization of the randomized algorithm since it sets the delays one–by–one, as opposed to the more complex ways used before.

3.2.2.2 Exact Partitions in Set Discrepancy

Set discrepancy problems [8] are combinatorially important, special cases of which can model divide–and–conquer situations; see, *e.g.*, the RNC edge coloring algorithm of Karloff & Shmoys [57]. Given a finite set X and a family of subsets $\mathcal{F} = \{S_1, S_2, \ldots, S_n\}$ of X, the goal is to come up with a "2–coloring" $\chi : X \rightarrow \{0,1\}$ such that the **discrepancy** $disc(\chi) \doteq max_i\ disc_i(\chi)$, where $disc_i(\chi) \doteq \{|(\sum_{j \in S_i} \chi(j)) - |S_i|/2|\}$, is minimized. It is known that a 2–coloring χ with $disc(\chi) = O(\sqrt{\Delta \log n})$ exists and can be computed in polynomial (in $|X|$ and n) time [8], and that a 2–coloring χ with $disc(\chi) = O(\Delta^{0.5+\epsilon}\sqrt{\log n})$ for any fixed $\epsilon > 0$ can be computed in NC [17,81,85], where $\Delta \doteq max_i|S_i|$. Using the ideas of Section 3.1.1, we can prove

Theorem 16 *Given a finite set* X *with* $|X|$ *even and a family of subsets* $\mathcal{F} = \{S_1, S_2, \ldots, S_n\}$ *of* X *such that* $\Delta = max_i|S_i|$, *there exists a 2–coloring* $\chi^* : X \rightarrow \{0,1\}$ *computable in polynomial (in* $|X|$ *and* n*) time, such that: (i)* $disc(\chi^*) = O(\sqrt{\Delta \log n})$, *and (ii)* $|\{y \in X : \chi^*(y) = 0\}| = |\{y \in X : \chi^*(y) = 1\}|$.

PROOF. For the existence proof, we can show that if we pick a random subset $Z \subseteq X$ with $|Z| = |X|/2$ uniformly from the set of all size $|X|/2$ subsets of X and set $\chi_Z(y) = 1$ iff $y \in Z$, then $Pr_Z(disc(\chi_Z) = O(\sqrt{\Delta \log n})) > 0$, as follows. It is well–known [8] and easily checked via the CH bounds that there is a constant $c > 0$ such that if $\chi(y)$ is picked uniformly and independently from $\{0,1\}$, then for any S_i,

$Pr(disc_i(\chi) > c\sqrt{\Delta \log n}) < \frac{1}{n}$; hence,

$$Pr(disc(\chi) > c\sqrt{\Delta \log n}) = Pr(\exists i:\ disc_i(\chi) > c\sqrt{\Delta \log n}) < n \cdot \frac{1}{n} = 1.$$

Note that $\{\chi_Z(y) :\ y \in X\}$ is a set of self–weakening random bits with parameter 1, *i.e.*, if Z is picked uniformly at random from the set of $|X|/2$ sized subsets of X, then for any distinct $y_1, y_2, \ldots, y_i \in X$,

$$\begin{aligned} E[\Pi_{j=1}^{i} \chi_Z(y_j)] &= Pr(\chi_Z(y_1) = \chi_Z(y_2) = \cdots = \chi_Z(y_i) = 1) \\ &\leq \Pi_{j=1}^{i} Pr(\chi_Z(y_j) = 1) \\ &= \Pi_{j=1}^{i} E[\chi_Z(y_j)]. \end{aligned}$$

Hence, it follows from Section 3.1.6 that for any S_i, $Pr(disc_i(\chi_Z) > c\sqrt{\Delta \log n}) < \frac{1}{n}$ still holds, concluding the existence proof.

Assume that $|X| = m$ and that $X = \{1, 2, \ldots, m\}$. To use the method of conditional probabilities to derandomize the above construction, note that for $\{i_1, i_2, \ldots, i_j\} \subseteq \{i+1, i+2, \ldots, m\}$,

$$\begin{aligned} E[\Pi_{\ell=1}^{j} \chi_Z(i_\ell) | \chi_Z(1) = b_1, \ldots, \chi_Z(r) = b_r] &= \text{undefined, if } r_1 > m/2 \text{ or } r_2 > m/2 \\ &= 0, \text{ if } r_1 + j > m/2 \\ &= \frac{\binom{m-r-j}{m/2-r_1-j}}{\binom{m-r}{m/2-r_1}}, \text{ otherwise} \end{aligned}$$

where

$$|\{\ell \mid (1 \leq \ell \leq r) \ \wedge \ (b_\ell = 1)\}| = r_1$$

and $r_2 = r - r_1$. This can now be derandomized, by our initial discussion in Section 3.2.2. $\square$

3.3 Conclusions and Open Problems

In this chapter, we have presented a simple method which shows that the CH upper bounds hold even under (a suitable amount of) limited independence, via a new technique. We then extend this method somewhat, and our combined results yield good tail probabilities for some other distributions too. The main computational application of this work, in our opinion, is the reduction in randomness we get for any analysis which requires the CH bounds, due to the sufficiency of limited independence. We expect to see more such applications in future.

An interesting open problem is to analytically solve or approximate well, the linear program introduced in Section 3.1.3. Techniques and tools from approximation theory have recently been put to good use to solve (or at least to approximate quite well) similar problems (Linial & Nisan [71], Paturi [94]). Can similar methods help here? Another open problem is to resolve the optimality of our results, though we cannot expect to get significant returns here as shown in Section 3.1.5.

Chapter 4

Randomness-Optimal Unique Element Isolation, with Applications to Perfect Matching and Related Problems

Given a graph G, a *matching* in G is a subset of the edges such that no two edges in the subset have a common endpoint. A *maximum matching* in G is a matching in G of maximum cardinality, and a *perfect matching* is a special type of maximum matching, in which each vertex is covered by an edge, *i.e.*, each vertex is *matched*. Finding a perfect matching, if any, in a graph is a fundamental problem in combinatorial optimization. While its sequential complexity has been well-studied (Lovász & Plummer [74]), it is a major open question whether a perfect matching can be found or even detected in NC. Considerable effort has been devoted towards developing parallel algorithms for the perfect matching problem. For instance, sublinear time parallel algorithms for general graphs (Goldberg, Plotkin & Vaidya [46], Vaidya [127], Grover [48]) and for bipartite graphs (Goldberg, Plotkin, Shmoys & Tardos [45]) are known. NC algorithms for special classes of graphs: cocomparability graphs

(Kozen, Vazirani & Vazirani [67]), strongly chordal graphs (Dahlhaus & Karpinski [33]), graphs with polynomially many perfect matchings (Grigoriev & Karpinski [47]), and planar bipartite graphs (Miller & Naor [80]) have also been developed. A successful approach for the general problem has been to use randomness: both Monte Carlo RNC algorithms (Karp, Upfal & Wigderson [61], Mulmuley, Vazirani & Vazirani [83]) and Las Vegas RNC algorithms (Karloff [55]) have been developed. This chapter describes joint work with S. Chari and P. Rohatgi [26].

In this chapter, we investigate the *parallel randomness complexity* of perfect matching and related problems, *i.e.*, the amount of randomness required to solve them in parallel. For perfect matching we present an RNC^2 algorithm which uses $O(\log Z)$ random bits where Z is any given upper bound on $|\mathrm{Mat}(G)|$, the number of perfect matchings in the graph G. This improves and generalizes the result of Grigoriev & Karpinski [47] who give an NC^3 algorithm when Z is polynomially bounded. Since $|\mathrm{Mat}(G)|$ is at most $(2m/n)^n$, the worst case randomness complexity of our algorithm is $O(n \log(m/n))$ which improves on the previous bound of $O(m \log n)$ due to Mulmuley, Vazirani & Vazirani [83]. Thus by linking the randomness complexity of this problem to $|\mathrm{Mat}(G)|$, we unify and generalize previous results, while also improving on them. In special cases, *e.g.*, $K_{3,3}$-free graphs, $|\mathrm{Mat}(G)|$ can be computed in NC^2 (see Vazirani [134], whose work was motivated by that of Kastelyn [63] and Little [73]); in general, if no good upper bound on $|\mathrm{Mat}(G)|$ is known, our results yield an RNC^3 algorithm to find a perfect matching which uses at most $O(\log |\mathrm{Mat}(G)|)$ random bits with high probability, if $|\mathrm{Mat}(G)| > 0$. This is a significant reduction if $0 < |\mathrm{Mat}(G)| \ll (2m/n)^n$. Our results are obtained by a *randomness-optimal* generalization of the Isolating Lemma of [83]. Since the Isolating Lemma has several applications such as to basic problems on linearly representable matroids, variants of matching and structural complexity theory, our generalization reduces the number of random bits needed in these settings as well.

Given a set $S = \{x_1, x_2, \ldots, x_N\}$ and an *unknown* family $\mathcal{F} \subseteq 2^S$, the Isolating

Lemma of [83] shows that if the x_i's are *independently* assigned weights uniformly at random from the range $\{1, 2, \ldots, 2N\}$, then with probability at least 1/2, the minimum weight set in $\mathcal{F}$ is *unique*. This clearly requires $\Theta(N \log N)$ random bits and it was left open whether this independence is necessary. We generalize this as follows: given an upper bound Z on $|\mathcal{F}|$, we assign polynomially bounded weights to the x_i's using only $O(\log Z + \log N)$ random bits and achieve the same result. In the worst case ($Z = 2^N$), our scheme needs $O(N)$ random bits as compared to $\Theta(N \log N)$; for smaller Z, we get better randomness complexity. This settles the open question of [83] by showing that independence is not necessary. Our scheme provides an explicit construction of a collection of $(NZ)^{O(1)}$ weight assignments such that, for every family $\mathcal{F}$ of size at most Z, at least one (in fact, at least 50%) of these assignments makes the minimum weight subset in $\mathcal{F}$ unique. Such a collection of size $O(NZ)$ is guaranteed to exist by applying Adleman's technique [1] to the original Isolating Lemma, and thus this is one of the few instances where Adleman's result can be made constructive with only a polynomial blowup in size. Furthermore, we prove a matching lower bound for our generalization, *i.e.*, any scheme which assigns polynomial weights to the x_i's must use $\Omega(\log Z + \log N)$ random bits even to achieve isolation with only *nonzero* probability.

As mentioned earlier, our Isolating Lemma directly yields randomness–efficient algorithms for perfect matching. We get an RNC^2 algorithm to find a perfect matching in a graph G which uses $O(\log Z)$ random bits, where Z is a *given* upper bound on $|\mathrm{Mat}(G)|$. We also obtain randomness–efficient algorithms for the exact matching problem. Matroid intersection and matroid matching are generalizations of bipartite and general graph matching, and the Isolating Lemma has been used to obtain RNC^2 algorithms for these basic problems on linearly representable matroids (Narayanan, Saran & Vazirani [86]). Here, by obtaining good bounds on the size of the family on which the Isolating Lemma operates, we obtain significant savings in the number of random bits required. The randomness complexities of our algorithms for matroids, when specialized to the case of matching, yield precisely our results on matching. In particular, we give NC^2 algorithms for these problems on matroids under certain restrictions which in the case of graph matching correspond exactly to a polynomial bound on $|\mathrm{Mat}(G)|$. This generalizes the results of [47] to matroids.

Using the generalized Isolating Lemma we obtain an alternative to the Valiant & Vazirani random reduction [131] from SAT to $USAT$. The randomness complexity of our reduction is logarithmic in the number of satisfying assignments and in the worst case matches the best–known bound of $O(n)$ (Tarui [121]). This directly gives new proofs of the results that $FewP$, where the number of satisfying assignments is polynomially bounded, is contained in $\oplus P$ (Cai & Hemachandra [22]) and in $C_{=}P$ (Köbler, Schöning, Toda & Toran [64]).

4.1 The Generalized Isolating Lemma

We now present a randomness–efficient generalization of the Isolating Lemma [83] and prove a matching lower bound for this problem.

Definition A *set system* $(S,\ \mathcal{F})$ consists of a finite set $S = \{x_1, \ldots, x_N\}$ and a family $\mathcal{F}$ of subsets of S, *i.e.*, $\mathcal{F} = \{S_1, \ldots, S_k\}$ where $S_j \subseteq S$. A weight assignment $\vec{w} = \langle w_1, \ldots, w_N \rangle$ to the elements $x_1, \ldots, x_N$, extends naturally to sets in $\mathcal{F}$ with $\vec{w}(S_j) = \sum_{x_i \in S_j} w_i$.

Isolating Lemma [83] *Let $(S,\ \mathcal{F})$ be any set system. If the elements x_i of S are assigned random weights* uniformly and independently *from $\{1, 2, \ldots, 2N\}$ then, with probability at least $1/2$, there will be a* unique *minimum weight set in $\mathcal{F}$.*

Two features make the isolating lemma widely applicable. First, the isolation process works for arbitrary and unknown families $\mathcal{F}$ and second, isolation is achieved by assigning only polynomial weights. For example, in the bipartite perfect matching problem where S is the set of edges and $\mathcal{F}$ is the collection of perfect matchings, the first feature allows isolation to be done in RNC without any knowledge of the matchings and the second ensures that a matching can be found by inverting a matrix with *polynomial* sized entries [83].

4.1.1 The New Isolation Scheme

We prove a randomness–efficient generalization of the Isolating Lemma which also assigns polynomial weights and which depends only on any given upper bound Z on the size of the *unknown* family $\mathcal{F}$. The motivation for this generalization is that in several applications, *e.g.*, perfect matching and matroid intersection, $Z \ll 2^N$. The original lemma uses $\Theta(N \log N)$ random bits since weights are assigned *independently* and it was left open in [83] whether independence was necessary. Since our generalized lemma uses $O(\log Z + \log N)$ random bits, this shows that independence is unnecessary, even when $Z = 2^N$.

Lemma 11 [Generalized Isolating Lemma] *Let $(S,\ \mathcal{F})$ be any set system and let Z be a* given *upper bound on the size of the unknown family $\mathcal{F}$. There is a simple scheme which uses $O(\log Z + \log N)$ random bits to assign integer weights to the x_i's in the range $[0, N^7)$ such that with probability at least $1/4$, there is a* unique *minimum weight set in $\mathcal{F}$. (This probabillity of $1/4$ can be boosted to any constant lesser than 1 using results of de Bruijn [34], with a constant factor blowup in the number of random bits used.)*

PROOF. We outline a four-step process for assigning weights to the x_i's and prove that this scheme has the desired properties. At step j, we assign an intermediate weight $\vec{w}^{(j)}$.

Since $\mathcal{F}$ is unknown we deterministically assign very large weights in Step 1 so that every set in $\mathcal{F}$ gets a *distinct* weight.

Step 1: For each i, set $w_i^{(1)} = 2^i$.

Under $\vec{w}^{(1)}$, sets in $\mathcal{F}$ have *distinct* weights in $\{1, 2, \ldots, 2^{N+1}\}$. Clearly, if $Z \ll 2^{N+1}$, the same property should be obtainable with much lower weights. Since $\mathcal{F}$ is unknown, a deterministic strategy for reducing weights may fail; instead, we use a randomized strategy.

Step 2: Choose m uniformly at random from $\{1, 2, \ldots, (2NZ^2)^2\}$. For each i, define $w_i^{(2)} = w_i^{(1)} \bmod m$.

Step 2 requires $O(\log Z + \log N)$ random bits. Under $\vec{w}^{(2)}$, the Z or fewer sets in $\mathcal{F}$ have weights in the interval $[0, \min(N(2NZ^2)^2, 2^{N+1})]$, which is a big improvement for small values of Z. We now claim that with good probability, these weights are also *distinct*.

Claim 1 *With probability at least* 1/2*,* all *sets in* $\mathcal{F}$ *have* distinct *weights under* $\vec{w}^{(2)}$.

PROOF. Suppose $\mathcal{F} = \{S_1, \ldots, S_k\}$ where $k \leq Z$. Consider the (unknown) integer

$$I = \prod_{1 \leq i < j \leq k} (\ \vec{w}^{(1)}(S_i) - \vec{w}^{(1)}(S_j)\ ).$$

Clearly the properties of $\vec{w}^{(1)}$ ensure that $I \neq 0$ and that $|I| < 2^{2NZ^2}$. The following number-theoretic proposition, together with the Chinese Remainder Theorem establishes that when m is chosen uniformly from $\{1, 2, \ldots, (2NZ^2)^2\}$, the probability that $m \nmid I$ is at least 1/2. Several versions of this proposition appear in the literature; this version is from [125] and the constant 89 has been improved to 3 by Thrash [122].

Proposition 2 *Let* $L \geq 89$ *and let* S *be any subset of* $\{1, \ldots, L^2\}$ *such that* $|S| \geq \frac{1}{2}L^2$*. Then, the least common multiple of the elements of* S *exceeds* 2^L.

We claim that if $m \nmid I$, then all sets in $\mathcal{F}$ have *distinct* weights under $\vec{w}^{(2)}$. If not, then $\vec{w}^{(2)}(S_i) = \vec{w}^{(2)}(S_j)$ for some $i < j$. Then we have $m \mid (\vec{w}^{(1)}(S_i) - \vec{w}^{(1)}(S_j))$, which contradicts the fact that $m \not| I$. □

Remark. We say that Step 2 *succeeds* if sets in $\mathcal{F}$ get distinct weights under $\vec{w}^{(2)}$. The success probability of this step can be boosted to any constant lesser than 1, since it follows from the results of de Bruijn [34] that for any fixed $p < 1$, there exist constants c_1 and c_2 such that if m is picked uniformly at random from $\{1, 2, \ldots, N^{c_1} Z^{c_2}\}$ in Step 2, then Step 2 will succeed with probability at least p.

Note that if $Z \leq N^c$, then $w_i^{(2)} \leq N^{4c+3}$ and with probability at least 1/2, *all* sets in $\mathcal{F}$ have *distinct* weights. Later, we use this strong condition to design an NC^2 algorithm to find *all* perfect matchings in graphs with at most $N^{O(1)}$ perfect matchings. For larger Z, the weights are still too big and in Steps 3 and 4 we reduce them further. Step 2 assigns q-bit weights to the x_i's where $q = min(N, \log m) \leq min(N, 4\log Z + 2\log(2N))$. Let $t = \lceil q/\log N \rceil$.

Step 3: For each i, write $w_i^{(2)}$ as q-bit number. Split these bits into t blocks of size $\log N$ bits each as shown. Let $b_{i,j}$ be the number in $[0, N-1]$ formed by the bits in block j.

$$w_i^{(2)} : \quad \overbrace{\boxed{b_{i,t-1}}\boxed{\cdots}\boxed{b_{i,1}}\underbrace{\boxed{b_{i,0}}}_{\log N}}^{q}$$

Let $w_i^{(3)}$ be the *linear form* $\sum_{j=0}^{t-1} b_{i,j} \cdot y_j$ over the variables $y_0, \ldots, y_{t-1}$.

Claim 2 *If Step 2 succeeds then the linear forms $\vec{w}^{(3)}(S_j)$, where $S_j \in \mathcal{F}$, are all* distinct.

Proof. Assume that Step 2 succeeds, *i.e.*, that all the weights $\vec{w}^{(2)}(S_j)$ are *distinct*, where $S_j \in \mathcal{F}$. Note that each $w_i^{(3)}$ *evaluated* at $y_k = 2^{k \log N}$, $0 \leq k \leq t-1$, is *exactly* $w_i^{(2)}$. This implies that each $\vec{w}^{(3)}(S_j)$ evaluates to the *distinct* value $\vec{w}^{(2)}(S_j)$ at $y_k = 2^{k \log N}$, $0 \leq k \leq t-1$, which implies that the forms $\vec{w}^{(3)}(S_j)$'s must be distinct. ∎

Note that each $\vec{w}^{(3)}(S_j)$ is a linear forms with coefficients in the range $[0, N(N-1)]$. We will use this property in a crucial way in the analysis of the next and final step.

Step 4: Choose $r_0, \ldots, r_{t-1}$ uniformly and independently at random from $\{1, 2, \ldots, N^5\}$. For each i, set $w_i^{(4)}$ to be the evaluation of $w_i^{(3)}$ at $y_k = r_k$, $0 \le k \le t-1$.

We claim that $\vec{w}^{(4)}$ achieves the requirements of the Generalized Isolating Lemma. Clearly, Step 4 requires $5 \log N \times t = O(\log Z + \log N)$ random bits and since Step 2 has a similar randomness complexity, the overall procedure requires $O(\log Z + \log N)$ random bits. It is easy to check that each $w_i^{(4)}$ is in $[0, N^7)$. Since Step 2 succeeds with probability at least $1/2$, by using $\mathcal{C} = \{\vec{w}^{(3)}(S_j) \mid S_j \in \mathcal{F}\}$ and $k = 2$ in the following proposition, we obtain that the weight assignment $\vec{w}^{(4)}$ achieves isolation with probability at least $1/4$.

Proposition 3 *Let $\mathcal{C}$ be any collection of* distinct *linear forms over at most N variables $\vec{y} = y_0, y_1, \ldots, y_{N-1}$ with coefficients in $\{0, 1, \ldots, N^k - 1\}$. Choose a random $\vec{r} = r_0, \ldots, r_{N-1}$ by assigning each r_i uniformly and independently from $\{1, 2, \ldots, N^{2k+1}\}$. Then in the assignment $\vec{y} = \vec{r}$ there will be a* unique *linear form with minimum value, with probability at least $1/2$.*

Proof of Proposition: Our proof parallels that of [83]; we define a variable y_i to be *singular* under an assignment $\vec{r}$ to the variables $\vec{y}$ iff under this assignment there exist two minimum valued linear forms in $\mathcal{C}$ having different coefficients of y_i.

For each y_i, we upper bound the probability that it is *singular.* Fix i and partition $\mathcal{C}$ into N^k classes $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_{N^k-1}$ based on the coefficient of y_i, *i.e.*, forms in $\mathcal{C}_j$ have j as the coefficient of y_i. By definition, y_i is *singular* under an assignment $\vec{r}$ iff all the minimum valued forms do not come from a *unique* class $\mathcal{C}_j$. Assume that $\vec{r}(\bar{\imath}) = r_0, \ldots, r_{i-1}, r_{i+1}, \ldots, r_{N-1}$ have been assigned values $\vec{a}(\bar{\imath}) = a_0, \ldots, a_{i-1}, a_{i+1}, \ldots, a_{N-1}$ and that r_i is not yet chosen. Consider the probability that the choice of r_i makes y_i singular, conditioned on $\vec{r}(\bar{\imath}) = \vec{a}(\bar{\imath})$. Under this partial assignment, each form $f \in \mathcal{C}_j$ has weight $b + jr_i$ where b is completely determined. Call b the *partial weight* of f. For each j, let p_j be the minimum partial weight among all forms in $\mathcal{C}_j$. Since any form with minimum weight after the choice of r_i must have had the minimum partial weight in its class, it follows that the probability that r_i makes y_i *singular* is exactly the probability that the minimum valued element in the list $\mathcal{D} = [p_0, p_1 + r_i, p_2 + 2r_i, \ldots, p_{N^k-1} + (N^k - 1)r_i]$ is not unique. This is at most the probability that under a random r_i the values of elements in $\mathcal{D}$ are not all distinct. For each pair of elements in $\mathcal{D}$, *at most one* choice of r_i makes their values equal. Thus the probability that y_i becomes *singular* by choice of r_i conditioned on $\vec{r}(\bar{\imath}) = \vec{a}(\bar{\imath})$ is at most $\binom{N^k}{2}/N^{2k+1} \leq 1/2N$, since r_i is independent of $\vec{r}(\bar{\imath})$.

This upper bound does not depend on the particular choice of $\vec{a}(\bar{\imath})$ and hence we get Prob[y_i is *singular*] $\leq 1/2N$. Since this holds for each i, Prob[*some* y_i is *singular*] $\leq N \times 1/2N = 1/2$. Thus with probability at least $1/2$, all y_i's are *non-singular* and hence there is a *unique* minimum valued form in $\mathcal{C}$. $\square$

Note that the Generalized Isolating Lemma can be implemented in NC^2. Finally, we note that the Generalized Isolating Lemma provides an *explicit construction* of $(NZ)^{O(1)}$ many weight assignment functions for the ground set $S = \{x_1, x_2, \ldots, x_N\}$, such that for any set system $(S, \mathcal{F})$ with $|\mathcal{F}| \leq Z$, at least half of these functions will produce a unique minimum weight set in $\mathcal{F}$. For a given ground set S, the number of set systems $(S, \mathcal{F})$ with $|\mathcal{F}| \leq Z$ is $\sum_{i=1}^{Z} \binom{2^N}{i}$; hence, the technique of Adleman [1] combined with the original isolating lemma shows *nonconstructively* that there exists a set of $O(\log(\sum_{i=1}^{Z} \binom{2^N}{i})) = O(NZ)$ many weight assignment functions such that for any set system $(S, \mathcal{F})$ with $|\mathcal{F}| \leq Z$, at least *one* of these functions will produce a unique minimum weight set in $\mathcal{F}$.

4.1.2 Lower bound for the Isolation problem

We now establish a matching lower bound of $\Omega(\log Z + \log N)$ random bits on the randomness complexity of the generalized Isolating Lemma, and in fact for the following weaker problem, thus showing that the randomness complexity of our Generalized Isolating Lemma is optimal to within a constant factor.

Generalized Isolation Let $S = \{x_1, \ldots, x_N\}$ and let k be a constant. Assign weights to the elements of S in the range $[1, N^k]$ such that for any family $\mathcal{F} \subseteq 2^S$ of size at most Z, the minimum weight set in $\mathcal{F}$ is unique with *nonzero* probability.

We first prove a lower bound of $\Omega(\log Z)$ random bits for any randomized scheme for the above problem. The following theorem shows that at least $t+1$ different weight assignments are needed in order to achieve isolation in all possible families of size at most Z, where $t = \min(\lfloor \frac{Z}{2} \rfloor, \lfloor \frac{2^N - 3}{N^{k+1}} \rfloor)$. Hence any randomized scheme must use $\Omega(\log t) = \Omega(\log Z)$ random bits.

Theorem 17 *Let $f_1, f_2, \ldots, f_t$ be any collection of weight assignments which assign weights in the range $[1, B]$ to the elements of S. If $t \leq \min(\frac{Z}{2}, \frac{2^N - 3}{NB})$, then there exists a family $\mathcal{F}$ of subsets of S with $|\mathcal{F}| \leq 2t \leq Z$, such that the minimum weight subset of $\mathcal{F}$ is not unique under any of these assignments.*

PROOF. Given any t weight assignments we explicitly construct the family $\mathcal{F}$ as follows. First, for each f_i, we compute the "histogram" of the weights of the subsets of S, *i.e.*, for each nonempty subset X we place a mark above the weight $f_i(X)$ in the histogram for f_i; one distinct mark is made for each such subset X.

The weight of any subset is in the range $[1, NB]$. Initialize, for each i, a pointer p_i to 1 in the histogram for f_i. The pointers p_i advance according to the following rules:

- If there are no marks on weight p_i in the histogram of f_i, then advance p_i to $p_i + 1$.
- If there is exactly one mark on weight p_i which corresponds to some subset X, remove the marks corresponding to X from *all* the histograms and advance p_i to $p_i + 1$.

If more than one pointer can advance, we choose one among them arbitrarily. We continue this process until the pointers cannot move any further. First we argue that not all marks can be removed since every time a mark is removed, one of the pointers advances and hence the potential function $\Psi = \sum_i (p_i - 1)$ always increases. Since each p_i is bounded by $NB+1$, Ψ is bounded by tNB. By assumption $tNB \leq 2^N - 3$, and hence when the pointers come to a halt there are at least two marks corresponding to nonempty subsets X_{i1} and X_{i2} on weight p_i in the histogram of f_i, for each i. The family $\mathcal{F} = \{X_{i1}, X_{i2} |\ 1 \leq i \leq t\}$ has size at most $2t \leq Z$ and for each i, there are two nonempty minimum weight subsets, X_{i1} and X_{i2}, of minimum weight p_i, under the assignment f_i. □

Also when $B = N^{O(1)}$ and $t = o(N/\log N)$, we can construct two sets which have the same weight under all the assignments $f_1, f_2, \ldots, f_t$, as follows. Since f_1 maps the nonempty subsets of S to the range $[1, NB]$, there exists a family A_1 of nonempty subsets of S, all of which get the same weight under f_1 and such that $|A_1| \geq \frac{2^N-1}{NB}$. Similarly, there exists a family $A_2 \subseteq A_1$ with $|A_2| \geq \frac{2^N-1}{(NB)^2}$, such that $f_i(X_1) = f_i(X_2)\ \forall X_1, X_2 \in A_2$, for $i = 1, 2$. Repeating this, we end up with a set A_t of nonempty subsets of S with $|A_t| \geq \frac{2^N-1}{(NB)^t}$, such that the minimum weight element of A_t is not unique under none of $f_1, f_2, \ldots, f_t$; since $\frac{2^N-1}{(NB)^t} \geq 2$ for $t = o(N/\log N)$, our claim holds. Thus, any scheme for Generalized Isolation requires $\Omega(\log Z + \log N)$ random bits.

Theorem 17 also gives lower bounds when the elements are assigned superpolynomial weights. In fact, we can prove the same lower bound even when the assignments map subsets of S to an arbitrary linearly ordered set of size B, since the above proofs use no property of the integers apart from their total ordering. Thus we have

Theorem 18 *Let $f_1, \ldots, f_t$ be a collection of assignments which are functions from 2^S to a linearly ordered range D. If $t \leq \min(\lfloor \frac{Z}{2} \rfloor, \frac{2^N-3}{|D|})$ or if $t = o(N/\log(ND))$, then there exists a family $\mathcal{F}$ of subsets of X with $|\mathcal{F}| \leq Z$, such that the minimum weight subset of $\mathcal{F}$ is not unique under any of the assignments.*

4.2 Applications to Matching Problems

The Generalized Isolating Lemma immediately yields randomness-efficient algorithms for several problems related to matching. Here we only consider matchings in bipartite graphs; with more work, identical results can be obtained for matchings in general graphs [74,83]. The failure probabilities of our algorithms can be made inverse polynomial with only a constant blowup in the time and randomness complexities and a polynomial blowup in the number of processors using results of Chor & Goldreich [28]. Our isolating lemma is a general tool to bound the randomness complexity of a problem in terms of the number of its solutions. However, due to the large polynomial weights used by our lemma, algorithms based directly on the lemma usually suffer a processor penalty.

Notation We denote a bipartite graph G by $G = (U, V, E)$, where $U = \{u_1, \ldots, u_{n/2}\}$, $V = \{v_1, \ldots, v_{n/2}\}$, and $|E| = m$. The phrase " perfect matching" is abbreviated "p.m.". The determinant of a matrix A is denoted by $det(A)$. The number of p.m.'s in G is denoted by $|\text{Mat}(G)|$.

4.2.1 Algorithms for matching

To apply the Isolating Lemma to perfect matching, we let the ground set be the set of edges and let the family of subsets be the set of perfect matchings of G, as in [83].

By the Generalized Isolating Lemma, given an upper bound Z on $|\mathrm{Mat}(G)|$, we can assign polynomially bounded weights to the edges of G using $O(\log Z + \log n)$ random bits such that there is a unique p.m. of minimum weight, with good probability. We construct a matrix $M[1..n/2, 1..n/2]$ with $M[i,j] = 2^{w_{ij}}$ if $(u_i, v_j) \in E$ and $M[i,j] = 0$ otherwise, where w_{ij} is the weight assigned to edge (u_i, v_j); a p.m. in G can now be found by inverting M via the NC^2 algorithm of Csanky [32], as shown in [83]. Note further that the additive $O(\log n)$ factor in the randomness complexity can be absorbed by a processor penalty. Also for any graph $G = (V, E)$, $|\mathrm{Mat}(G)| \leq \prod_{v \in V} degree(v)$, which is at most $(2m/n)^n$ since $\sum_{v \in V} degree(v) = 2m$. In the worst case, by letting $Z = (2m/n)^n$ we obtain a randomness complexity of $O(n \log(m/n))$ which improves the previous bound of $O(m \log n)$ [83]. Thus, we get

Theorem 19 *There is an RNC^2 algorithm to find a perfect matching in a graph G (if one exists) which uses $O(\log Z)$ random bits, given an upper bound Z on $|\mathrm{Mat}(G)|$. In the worst case, this algorithm uses $O(n \log(m/n))$ random bits.*

A careful analysis of the proof of our isolating scheme when applied to the worst case where $Z = (2m/n)^n$ and $m = \Theta(n^2)$, shows that the following algorithm also achieves the same randomness complexity: the edges of each vertex $u \in U$ are assigned weights *pairwise independently* in the range $[1, m^7]$. This idea was, in fact, the starting point for this research.

Perfect matching is a special case of the *Exact Matching Problem*: given a graph G with edges colored red and blue arbitrarily and an integer k, an exact matching is a p.m. of G with *exactly* k red edges (Papadimitriou & Yannakakis [93]). Even though the problem of finding an exact matching is not known to be in P, the Isolating Lemma has been used to solve it in RNC^2 [83]! By observing that isolation is needed only among the exact matchings, we can link the randomness complexity of this problem to a bound X on the number of exact matchings and *not* on $|\mathrm{Mat}(G)|$; here again, we follow the lead of [83]. Suppose we assign a nonnegative integral weight w_{ij} to every edge $(u_i, v_j) \in E$. Let y be an indeterminate; construct a symbolic matrix $M_y[1..n/2, 1..n/2]$ such that

$$\begin{aligned} M_y[i,j] &= y \cdot w_{i,j} \text{ if } (u_i, v_j) \in E \text{ and is colored red,} \\ &= w_{ij} \text{ if } (u_i, v_j) \in E \text{ and is colored blue,} \\ &= 0, \text{ otherwise.} \end{aligned}$$

Then, it is easy to see that the contribution to the coefficient of y^k in $det(M_y)$ comes precisely from the exact matchings in G and that if there is a unique minimum weight exact matching M^*, then its weight equals the highest power of 2, say p^*, which divides this coefficient of y^k. Furthermore, any edge can be tested for membership in M^* by increasing its weight by 1 and testing if the new value of p^* is higher than its old value or not. Hence, this problem can be solved by using our Generalized Isolating Lemma over the ground set E and the (unknown) family of exact matchings of G. Since the determinant of a matrix in one variable can be computed in NC^2 (Borodin, Cook & Pippenger [21]), we get

Theorem 20 *Given an upper bound X on the number of exact matchings in a graph G, an exact matching (if any) in G can be found in RNC^2 using $O(\log X)$ random bits.*

To find a maximum matching in G, we proceed as follows. Note that G has a maximum matching of size at least k iff the graph $G_k = (V_k, E_k)$, got by adding to G an independent set of $n - 2k$ vertices, each of which is adjacent to every vertex in V, has a perfect matching; it is also straightforward to extract a matching in G of size k, from a perfect matching in G_k. We first make the failure probability of our perfect matching algorithm at most $\frac{1}{n+2}$ and make parallel calls to find a perfect matching in G_k for $k = 0, 1, \ldots, n/2$, using the *same sequence of random bits* for each call. Then, with probability at least 1/2, all the calls will return the correct value and we can then find a maximum matching in G. We require $O(n \log n)$ random bits for this, since $|V_k| \log(|E_k|/|V_k|) = O(n \log n)$ for each k.

Theorem 21 *A maximum matching in G can be found in RNC^2 using $O(n \log n)$ random bits.*

In contrast, the best-known randomness complexity for maximum matching so far is $O(n^2 \log n)$ [83].

4.2.2 Graphs with a polynomial number of perfect matchings

Grigoriev & Karpinski [47] consider the problem of finding all the perfect matchings in graphs with a polynomial bound on $|\mathrm{Mat}(G)|$. Using techniques from algebra, they give an NC^2 algorithm for this problem if $|\mathrm{Mat}(G)| \leq \log^{0.5-\epsilon} n$, and an NC^3 algorithm when $|\mathrm{Mat}(G)| \leq n^c$ for constant c. However, their matching extraction algorithm is limited by their matching detection algorithm and hence they suggest that $\log^{0.5-\epsilon} n$ may be a limiting upper bound on $|\mathrm{Mat}(G)|$ to find all (or even one) of the perfect matchings in NC^2 via parallel algebra. We present a simple NC^2 algorithm for the detection problem which provides enough information to find all matchings in NC^2 even when $|\mathrm{Mat}(G)| \leq Z = n^c$.

As before, we let S and $\mathcal{F}$ be E and $\mathrm{Mat}(G)$ respectively. Running steps 1 and 2 of the generalized Isolating scheme, we use $O(\log n)$ random bits to assign each edge a polynomially bounded weight such that *all* the perfect matchings of G have *distinct* weights with probability at least $1/2$. Let $w_{i,j}$ be the weight assigned to edge (u_i, v_j). Construct a matrix A with $A[i,j] = 2^{2w_{i,j}}$ if $(u_i, v_j) \in E$, and 0 otherwise. Since every matching has a distinct weight, $det(A)$ gives information about *every* p.m., as follows.

Assume that all the perfect matchings indeed get distinct weights. Recall that if we denote the matrix obtained from M by removing its ith row and jth column by M_{ij}, then

$$(M^{-1})[j,i] = \frac{(-1)^{i+j} det(M_{ij})}{det(M)}.$$

Now, if there is a perfect matching of weight k containing the edge (u_i, v_j), then $det(M_{ij})$ is of the form

$$\pm 2^{2k-2w_{ij}} + \sum_{\ell=0}^{k-1} a_\ell 2^{2\ell-2w_{ij}} + \sum_{\ell \geq k+1} a_\ell 2^{2\ell-2w_{ij}},$$

where each a_ℓ is in $\{-1, 0, 1\}$. Since

$$|\sum_{\ell=0}^{k-1} a_\ell 2^{2\ell-2w_{ij}}| < 2^{2k-1-2w_{ij}},$$

this is of the form

$$\pm 2^{2k-2w_{ij}} \pm 2^{2k+2-2w_{ij}} n_0 \pm 2^{2k-1-2w_{ij}} x,$$

where n_0 is a nonnegative integer and $0 \leq x < 1$. Similarly, if there is no perfect matching of weight k containing the edge (u_i, v_j), then $det(M_{ij})$ is of the form

$$\pm 2^{2k+2-2w_{ij}} n_0 \pm 2^{2k-1-2w_{ij}} x.$$

Hence, there is a perfect matching of weight k containing the edge (u_i, v_j) iff

$$\frac{|(M^{-1})[j,i] \cdot det(M) \cdot 2^{2w_{ij}}|}{2^{2k-1}}$$

is of the form $8n_1 \pm y$, where n_1 is an integer $1 < y < 3$. Thus, we have a way to generate all the perfect matchings of G, since k is polynomially bounded. Since we use $O(\log Z + \log n) = O(\log n)$ random bits, we can derandomize this by searching over all points in the sample space in parallel to get

Theorem 22 *There is an NC^2 algorithm to detect if a perfect matching exists and to find all perfect matchings in a graph G, when given that $|\mathrm{Mat}(G)| \leq n^c$ for some constant c.*

Grigoriev & Karpinski also give a Las Vegas RNC^3 algorithm for the problem of testing if a graph has at most n^c perfect matchings, which uses expected $O(n^{2c+8.5} \log n)$ random bits. We can, using the algorithm of Theorem 22 and by extending ideas of [47,55], improve on the running time and substantially on the number of random bits used, as follows. Karloff presents an NC^2 algorithm which calls a perfect matching algorithm (oracle) on $n+1$ graphs (each with $n-1$ or n vertices and at most m edges) in parallel, and which, assuming that these calls worked correctly, will announce *correctly* if G has a p.m. or not; in any case, if it ever says that $|\mathrm{Mat}(G)| = 0$, then $|\mathrm{Mat}(G)|$ is indeed zero. (Actually, Karloff's result is more powerful– it works for maximum matchings.) Note that in conjunction with any Monte Carlo RNC^2 algorithm to find a perfect matching (in particular, ours) this yields a Las Vegas RNC^2 algorithm to decide if G has a perfect matching or not, and if so, to produce one. We first make the error probability of our perfect matching algorithm at most $\frac{1}{2(n+1)}$ (as mentioned in Section 4.1, this can be done in RNC^2), and we now invoke Karloff's algorithm, which will make calls in parallel to our algorithm, using the *same sequence of random bits* for each call; the probability that at least one of these calls produced a wrong result is at most $\frac{n+1}{2(n+1)} = 1/2$. Hence, we can decide if $|\mathrm{Mat}(G)| > 0$ or not *correctly*, using an expected $O(n \log(m/n))$ many random bits. If we conclude that $|\mathrm{Mat}(G)| > 0$, then we assume that $|\mathrm{Mat}(G)| \leq n^c + 1$, and run the algorithm of Section 4.2.2 to generate all the perfect matchings of G; if G has at most n^c perfect matchings, then at most n^c perfect matchings will be generated. Thus, we get

Theorem 23 *There exists a Las Vegas RNC^2 algorithm to verify if a given graph G has $|\mathrm{Mat}(G)| \leq n^c$ whose expected randomness complexity is $O(n \log(m/n))$.*

4.2.3 Algorithms with no information on $|\mathrm{Mat}(G)|$

The algorithms of Theorem 19 can be used to obtain randomness–efficient algorithms for perfect matching even when no upper bound on $|\mathrm{Mat}(G)|$ is given, by using the worst case upper bound of $Z = (2m/n)^n$. Consider the following improved algorithm:

for $i := 0$ **to** $\lceil \log_2 \log_2((2m/n)^n) \rceil$ **do**

Let $Z = 2^{2^i}$ and run the algorithm of Section 4.2.1 to find a perfect matching; **exit** if a perfect matching is found.

If $|\mathrm{Mat}(G)| > 0$, then this will find a p.m. with good probability when $2^{2^i} \geq |\mathrm{Mat}(G)|$. If $M = |\mathrm{Mat}(G)|$, the number of random bits used and the time taken will be $O(\log M)$ and $O(\log^2 n \log\log M)$ respectively with good probability, and so we get

Theorem 24 *Let* $M = |\mathrm{Mat}(G)|$. *There is an RNC algorithm to find a perfect matching in* G *with no given upper bound on* M*, which uses* $O(\log M)$ *random bits and runs in time* $O(\log^2 n \log\log M) = O(\log^3 n)$ *with high probability, if* $M > 0$*; in the* worst case, *it uses* $O(n \log(m/n))$ *random bits and* $O(\log^3 n)$ *time.*

4.3 Other applications

Due to the wide applicability of the Isolating Lemma, our generalized Isolating Lemma leads to randomness–efficient solutions to various problems, which we show below.

4.3.1 Randomized reductions from SAT to $USAT$

A seminal result in complexity theory is Valiant & Vazirani's randomized reduction from SAT to $USAT$, the language of uniquely satisfiable Boolean formulae [131]. This reduction is at the core of several fundamental results in complexity theory, most notably in the results of Toda on the unexpected power of counting classes [124]. Using the Isolating Lemma, an alternative randomized reduction for this problem was derived in [83]. We apply the Isolating Lemma to get a slightly different reduction and here, our generalization yields better randomness complexity.

Given a formula $F(x_1, x_2, \ldots, x_n)$ we let the ground set S be $\{x_1, x_2, \ldots, x_n\}$ and let the family $\mathcal{F}$ be the satisfying assignments of F, where a satisfying assignment is represented by the subset of variables that are true in it. Using our generalized

Isolating Lemma, we construct a formula F' such that if $F \in SAT$, then $F' \in USAT$ with probability at least $\frac{1}{2n^8}$ and if $F \notin SAT$, then F' is not satisfiable, as follows.

We construct an NP machine $M(f(x_1, x_2, \ldots, x_n), < w_1, w_2, \ldots, w_n >, y)$, whose inputs are a boolean formula f over the variables $x_1, x_2, \ldots, x_n$, weights w_i for the x_i's, and an integer y. Given these inputs, M will nondeterministically guess a satisfying assignment S_f for f and accept the input iff the weight of the subset corresponding to S_f is y, under $\vec{w}$. Now given F, we first generate random weights $w_1, w_2, \ldots, w_n$ as specified by the Generalized Isolating Lemma for $(S, \mathcal{F})$, and also pick an integer r uniformly at random from $\{0, 1, \ldots, n^8 - 1\}$. F' is defined to be the boolean formula corresponding to $M(F(x_1, x_2, \ldots, x_n), < w_1, w_2, \ldots, w_n >, r)$, as given by the Cook-Levin reduction. If $F \in SAT$, then with probability at least $1/2$, the Generalized Isolating Lemma will succeed and given this, r is the weight of the unique minimum weight satisfying assignment with probability at least $\frac{1}{n^8}$ and hence, $F' \in USAT$ with probability at least $\frac{1}{2n^8}$. On the other hand, if $F \notin SAT$, then F' is clearly not satisfiable.

This requires $O(\log Z + \log n)$ random bits, where Z an upper bound on the number of satisfying assignments of F. In the worst case this randomness complexity is $O(n)$, which improves on the $O(n \log n)$ bound of [83], the $O(n^2)$ bound of [131] and matches the result of Tarui [121]. If the number of satisfying assignments of F is polynomially bounded, as in the class $FewP$, we can derandomize this to get a polynomial number of formulas such that F is satisfiable iff one of these formulas is uniquely satisfiable. This directly yields simple new proofs of the results $FewP \subseteq \oplus P$ [22] and $FewP \subseteq C_{=}P$ [64]. $\oplus P$ and $C_{=}P$ are counting versions of NP where acceptance is based on the number of accepting paths being odd and exactly half of the total respectively.

Theorem 25 *There is a randomized reduction from SAT to $USAT$ which uses $O(\log Z + \log n)$ random bits, where n and Z are the number of variables and an upper bound on the number of satisfying assignments respectively. Hence, $FewP \subseteq \oplus P$*

and $FewP \subseteq C_=P$.

4.3.2 Improved parallel randomness complexity for problems on matroids

The Isolating Lemma has been used to obtain RNC algorithms for basic problems on linearly representable matroids such as matroid intersection and matching (Narayanan, Saran & Vazirani [86]). These problems are generalizations of bipartite and general graph matching. A set system $M = (S, \mathcal{F})$ is a *matroid* if:

- $\phi \in \mathcal{F}$,
- if $A \subseteq B \subseteq S$ and $B \in \mathcal{F}$, then $A \in \mathcal{F}$, and
- if $A_1 \in \mathcal{F}$ and $A_2 \in \mathcal{F}$, then $\exists x \in A_2 - A_1 : A_1 \cup \{x\} \in \mathcal{F}$.

Every element of $\mathcal{F}$ is called an *independent set.* A basic consequence of the matroid axioms is that every maximal independent set is also a maximum independent set, and the cardinality of any maximum independent set of M is also called the *rank* of M. A matroid M of rank r over a ground set S of cardinality n is *linearly representable* over a field $\mathbf{F}$ if there exists a matrix $C \in \mathbf{F}^{r \times n}$ whose columns are indexed by the elements of S, such that a subset of S is independent in M iff the corresponding set of columns of C are linearly independent over $\mathbf{F}$. M is *linearly represented* if it is presented as the matrix B. Henceforth, the field $\mathbf{F}$ is assumed to be the field of rationals, $\mathbf{Q}$. See, for example, Welsh [137] for an introduction to matroid theory. All the results of [86] and of this section hold *only* for linearly represented matroids. See Camerini, Galbiati & Maffioli [23] for related work on randomized algorithms for matroid problems.

The matroid intersection problem is to find a maximum cardinality independent set in both of two given matroids M_1 and M_2, each of rank r and over the same ground set S of cardinality n. In this case, we observe that the size of the family on which the Isolating Lemma operates, as presented in [86], is at most $I(\binom{r}{r-h})^2$

where h is the size of the largest set in $M_1 \cap M_2$ and I is the number of sets of size h in $M_1 \cap M_2$. Since this can be bounded by $(nr^2)^h \leq (nr^2)^r \leq n^{3r}$, we can use generalized isolation to reduce the randomness complexity from the previous bound of $O((n + r^2) \log n)$ [86] to $O(r \log n)$, improving by a factor of at least $\Omega(\sqrt{n})$ in all cases, and by more in general. If $h \geq r - O(\log_r n)$ and there is a polynomial (in n) bound on I then since $I(\binom{r}{r-h})^2 = n^{O(1)}$, we get the first NC algorithms. In the case of bipartite matching which is a special case of matroid intersection, $h = r$, and a polynomial bound on I corresponds exactly to a polynomial bound on $|\mathrm{Mat}(G)|$. Thus, this generalizes of the results of [47] to matroids.

Theorem 26 *Matroid intersection for linearly represented matroids of rank r can be solved in RNC^2 using $O(r \log n)$ random bits. If the cardinality of the maximum intersection is given to be at least $r - O(\log_r n)$ and if the number of maximum cardinality intersections is given to be bounded by a given polynomial of n, then it can be solved in NC^2.*

For linearly represented vectors, the matroid matching problem is the following: given m pairs of vectors over $\mathbf{Q}^{2n}$, pick the largest number of pairs so that the vectors picked are linearly independent. For this problem, the RNC^2 algorithm of [86] uses the Isolating Lemma on a set family over a ground set of $m + \binom{2n}{2}$ elements and at most

$$\binom{m+\binom{2n}{2}}{n} \leq (m + 2n^2)^n$$

subsets and thus by invoking the Generalized Isolating Lemma, we improve the randomness complexity from $O((m + n^2) \log(m + n^2))$ [86] to $O(n \log(m + n^2))$. Tiwari, using sparse interpolation techniques, gives NC algorithms for this problem when the number of full dimensional solutions (*i.e.*, those in which n pairs get picked) is given to be polynomially bounded in n and m [123].

Theorem 27 *Matroid matching can be solved in RNC^2 using $O(n \log(m + n^2))$ random bits.*

4.3.3 Randomness–efficient hash functions

Given finite sets A and B, Carter & Wegman [24] define a family of functions H from A to B to be k–universal if when a random function f is picked from H, then the random variables $\{f(a) : a \in A\}$ are k–wise independent. Their original construction requires $\Theta(k(\log|A| + \log|B|))$ random bits to sample a function from H, and this is also optimal in general. Universal hash functions have wide–ranging applications. However, the known constructions have the drawback that even in cases where we only have some small but arbitrary subset of A to be hashed, they still require $\Omega(k(\log|A| + \log|B|))$ random bits. This is rectified by using ideas from step 2 of the Generalized Isolating Lemma. The following theorem has already been proved earlier (Dietzfelbinger, Gil, Matias & Pippenger [35]), but we present here a proof for conciseness.

Theorem 28 *Given finite sets A and B, an unknown subset $C \subseteq A$ and an upper bound M on $|C|$, we can construct, using $O(\log\log|A| + \log M)$ random bits, a family of hash functions H from C to B which is k–universal with probability at least 1/2 and which is samplable with $O(k(\log\log|A| + \log|B| + \log M))$ random bits.*

PROOF. Since each element of A can be indexed by a subset of $\{1, 2, \ldots, \lceil\log_2|A|\rceil\}$, any $C \subseteq A$ with $|C| \leq M$ can be thought of as a family of at most M subsets of the ground set $\{1, 2, \ldots, \lceil\log_2|A|\rceil\}$. By applying steps 1 and 2 of our Generalized Isolating Lemma to this set system, we effectively get weights in the range $\{0, 1, \ldots, (M\log_2|A|)^a\}$ to each element of C (where a is a constant) such that no two of these weights are the same, with probability at least 1/2. We can thus index each element of C by a distinct element of $\{0, 1, \ldots, (M\log_2|A|)^a\}$ and then pick a k–universal family from $\{0, 1, \ldots, (M\log_2|A|)^a\}$ to B. $\square$

As an application, consider the situation where any n jobs from a large ($2^{n^{O(1)}}$ sized) universe may compete for a resource at a given time and one of them must be selected at random. The jobs are isolated from each other but can read from a shared random source. Since the IDs of the n currently competing jobs are unknown, the random source cannot be used to sample a global process ID because it is unlikely that any competing job will have that ID. The usage of existing schemes (*e.g.*, Rabin's lemma [99]) requires n independent random sources and $\Omega(n)$ random bits. Using our hash functions with $k = 2$, we get an optimal solution which uses $\Theta(\log n)$ random bits and succeeds with probability at least $1/4 - \epsilon$ for any fixed $\epsilon > 0$, as follows.

Theorem 29 *Given any n objects from a universe with upto $2^{n^{O(1)}}$ objects, a unique winner among the objects can be picked with probability at least $1/4 - \epsilon$ for any fixed $\epsilon > 0$, using only $\Theta(\log n)$ random bits from a common random source and with no other means of communication.*

PROOF. In the notation of Theorem 28, let $A = U$, $B = \{1, 2, \ldots, 2n\}$, C be the (unknown) set of competing processes $p_1, p_2, \ldots, p_n$, and $k = 2$. Pick a hash function $h : C \to B$ as outlined in Theorem 28, and define random bit X_i to be 1 iff $h(p_i) = 1$, for $i = 1, 2, \ldots, n$. We now claim that assuming that the constructed family $H : \{C \to B\}$ is indeed 2–universal, then exactly one X_i, $i = 1, 2, \ldots, n$, will be 1 with probability at least $1/4$; once this event occurs, the corresponding process can be allotted the resource. Assume that H is indeed 2–universal. Then given this,

$$\begin{aligned}
Pr(\exists!\, i : X_i = 1) &= \sum_{i=1}^{n} Pr(X_i = 1) Pr\left(\bigwedge_{j \neq i} X_j = 0 | X_i = 1\right) \\
&\geq \sum_{i=1}^{n} Pr(X_i = 1)\left(1 - \sum_{j \neq i} Pr(X_j = 1 | X_i = 1)\right) \\
&= \sum_{i=1}^{n} Pr(X_i = 1)\left(1 - \sum_{j \neq i} Pr(X_j = 1)\right) \quad \text{(due to 2–universality)} \\
&= \sum_{i=1}^{n} \frac{1}{2n}\left(1 - \frac{n-1}{2n}\right) \\
&\geq 1/4.
\end{aligned}$$

Now since step 2 of the Generalized Isolating Lemma can be made to succeed with probability arbitrarily close to 1, this whole scheme can be made to succeed with probability at least $1/4 - \epsilon$, for any fixed $\epsilon > 0$. The randomness complexity of this algorithm is $\Theta(\log n)$, by Theorem 28; this is clearly optimal to within a constant factor since $\lceil \log_2 n \rceil$ random bits are necessary to index all of the n processes. $\square$

The motivation for this algorithm came from Rabin's probabilistic lemma [99].

4.3.4 Conclusions and open problems

The main contribution of this chapter is our Generalized Isolating Lemma. For various problems solvable via randomness in general [83,86,131] and deterministically when the number of solutions is small [67,47,22], this lemma bounds their randomness complexity in terms of the number of solutions. Our results span the spectrum of the number of solutions, imply, and sometimes improve the previously known results at the extremes. On the other hand, our lower bound for isolation implies that attempts to solve these problems deterministically based on isolation must also exploit their structure, rather than view them merely as abstract unknown collections of sets.

A direct open problem falling out of the Generalized Isolating Lemma is to reduce the magnitude of the weights assigned by it to the elements of the ground set; currently, they can be as high as N^7. In the context of matching, this will lead to more processor–efficient RNC algorithms. Another interesting problem is to see if there is a result analogous to what we get for perfect matching, for maximum matching–given an upper bound M on the number of maximum matchings in a graph, can we find a maximum matching in it in RNC, using $O(\log M)$ random bits?

Chapter 5

Conclusions and Open Problems

In this work, we have added a few tiny drops to the ocean of Large Deviation Theory, in Chapters 2 and 3. It is our hope that the methods of Chapter 3, in particular, will be understood better and/or put to more use in future. In Chapter 4, we have studied the amount of randomness needed for a versatile probabilistic lemma. the Isolating Lemma.

Several interesting open problems remain, in the areas of randomness-efficient computation and derandomization. Of the many on which this author's attempts have been unsuccessful, here are a few.

(i) There are some classical problems for which "almost polynomial-time" solutions are known: randomized polynomial-time algorithms for them, which require just $poly(\log n)$–wise independence (n is the problem size) are known, *e.g.*, explicit constructions of Ramsey graphs (Erdős [38]) and set-discrepancy in parallel [17,81]. Can these algorithms be derandomized?

(ii) Our understanding of ϵ–biased random variables is far from satisfactory; see Section 3.1.5 for the relevant definitions. For instance, if $X_1, X_2, \ldots, X_n$ are binary ϵ–biased random variables, then the only known upper bound on $E[((\sum_{i=1}^{n} X_i) - n/2)^k]$

is

$$E[((\sum_{i=1}^{n} X_i) - n/2)^k] \leq E^* + n^k \epsilon, \tag{5.1}$$

where E^* is the value of $E[((\sum_{i=1}^{n} X_i) - n/2)^k]$ when the X_i's are unbiased and independent. This follows from the simple fact that $E[((\sum_{i=1}^{n} X_i) - n/2)^k]$ is the sum of n^k terms, each of the form

$$c \cdot Pr(X_{i_1} = X_{i_2} = \cdots = X_{i_\ell} = 1).$$

Can we explicitly construct small ϵ–biased sample spaces for which we can get much better guarantees than (5.1)?

(iii) Recent breakthrough results on pseudorandom generators for space bounded computation have shown, in particular, that testing if a given *undirected* graph on n vertices is connected, is achievable *deterministically* in $O(\log^{1.5} n)$ space [87,88,89]. Note that this problem is in randomized Logspace (Aleliunas, Karp, Lipton, Lovász & Rackoff [4]), and in fact, in zero-error probabilistic Logspace and polynomial time, ZPP (Borodin, Cook, Dymond, Ruzzo & Tompa [20]). Can the amount of randomness needed for this problem be further reduced, making it solvable deterministically in $o(\log^{1.5} n)$ space? Indeed, is this problem in Logspace?

Bibliography

[1] L. Adleman. Two theorems on random polynomial time. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 75–83, 1978.

[2] L. Adleman and M.-D. A. Huang. Recognizing primes in random polynomial time. In *Proc. ACM Symposium on Theory of Computing*, pages 462–469, 1987.

[3] M. Ajtai, J. Komlós, and E. Szemerédi. Deterministic simulation in $LOGSPACE$. In *Proc. ACM Symposium on Theory of Computing*, pages 132–140, 1987.

[4] R. Aleliunas, R. M. Karp, R. J. Lipton, L. Lovász, and C. Rackoff. Random walks, universal traversal sequences, and the complexity of maze problems. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 218–223, 1979.

[5] 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.

[6] N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Simple constructions of almost k–wise independent random variables. *Random Structures and Algorithms*, 3(3):289–303, 1992.

[7] N. Alon and J. Spencer. Personal communication. February 1993.

[8] N. Alon, J. Spencer, and P. Erdős. *The Probabilistic Method.* Wiley–Interscience Series, John Wiley & Sons, Inc., New York, 1992.

[9] D. Angluin and L.G. Valiant. Fast probabilistic algorithms for Hamiltonian circuits and matchings. *Journal of Computer and System Sciences*, 18:155–193, 1979.

[10] B. Awerbuch, A. V. Goldberg, M. Luby, and S. A. Plotkin. Network decomposition and locality in distributed computation. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 364–369, 1989.

[11] Y. Azar, R. Motwani, and J. Naor. Approximating arbitrary probability distributions using small sample spaces. Manuscript, 1990.

[12] E. Bach. Realistic analysis of some randomized algorithms. *Journal of Computer and System Sciences*, 42:30–53, 1991.

[13] I. Bárány and Z. Füredi. Computing the volume is difficult. *Discrete and Computational Geometry*, 2:319–326, 1987.

[14] M. Bellare, O. Goldreich, and S. Goldwasser. Randomness in interactive proofs. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 563–573, 1990.

[15] M. Bellare and J. Rompel. Randomness efficient sampling of arbitrary functions. Technical Report MIT/LCS/TM-433.b, Laboratory for Computer Science, Massachusetts Institute of Technology, July 1990.

[16] B. Berger and L. Cowen. Fast deterministic constructions of low-diameter network decompositions. MIT-LCS Technical Memo #460, April 1991.

[17] B. Berger and J. Rompel. Simulating ($\log^c n$)-wise independence in NC. *J. Assoc. Comput. Mach.*, 38(4):1026–1046, 1991.

[18] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. Comput.*, 13:850–864, 1984.

[19] B. Bollobás. *Graph Theory*. Springer Verlag, New York, 1979.

[20] A. Borodin, S. A. Cook, P. W. Dymond, W. L. Ruzzo, and M. Tompa. Two applications of inductive counting for complementation problems. *SIAM J. Comput.*, 18(3):559–578, 1989. See also Volume 18, Number 6, page 1283, 1989.

[21] A. Borodin, S. A. Cook, and N. Pippenger. Parallel computation for well-endowed rings and space-bounded probabilistic machines. *Information and Control*, 58:113–136, 1983.

[22] J-Y. Cai and L. Hemachandra. On the Power of Parity Polynomial Time. *Mathematical Systems Theory*, 23(2):95–106, 1990.

[23] P. M. Camerini, G. Galbiati, and F. Maffioli. Random pseudopolynomial algorithms for exact matroid problems. *Journal of Algorithms*, 13(2):258–273, 1992.

[24] J. L. Carter and M. N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18:143–154, 1979.

[25] S. Chari, P. Rohatgi, and A. Srinivasan. Improved approximations of independent probability distributions. Manuscript in preparation, 1993.

[26] S. Chari, P. Rohatgi, and A. Srinivasan. Randomness-optimal unique element isolation, with applications to perfect matching and related problems. In *Proc. ACM Symposium on Theory of Computing*, pages 458–467, 1993.

[27] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–509, 1952.

[28] B. Chor and O. Goldreich. On the power of two–point sampling. *Journal of Complexity*, 5:96–106, 1989.

[29] B. Chor, O. Goldreich, J. Håstad, J. Friedman, S. Rudich, and R. Smolensky. The bit extraction problem or t–resilient functions. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 396–407, 1985.

[30] V. Chvátal. The tail of the hypergeometric distribution. *Discrete Mathematics*, 25:285–287, 1979.

[31] A. Cohen and A. Wigderson. Multigraph amplification. Manuscript, 1989.

[32] L. Csanky. Fast parallel matrix inversion algorithms. *SIAM J. Comput.*, 5:618–623, 1976.

[33] E. Dahlhaus and M. Karpinski. The matching problem for strongly chordal graphs is in NC. Technical Report 855–CS, University of Bonn, 1986.

[34] N. de Bruijn. On the number of positive integers $\leq x$ and free of prime factors $> y$. *Proc. Kon. Ned. Akad. Wet.(Indag. Math. 13)*, A54:50–60, 1951.

[35] M. Dietzfelbinger, J. Gil, Y. Matias, and N. Pippenger. Polynomial hash functions are reliable. In *Automata, Languages, and Programming (ICALP)*, Lecture Notes in Computer Science # 623, pages 235–246. Springer-Verlag, Berlin, 1992.

[36] M. Dyer, A. Frieze, and R. Kannan. A random polynomial time algorithm for approximating the volume of convex bodies. *J. Assoc. Comput. Mach.*, 38(1):1–17, 1991.

[37] G. Elekes. A geometric inequality and the complexity of computing volume. *Discrete and Computational Geometry*, 1:289–292, 1986.

[38] P. Erdős. Some remarks on the theory of graphs. *Bulletin of the American Mathematics Society*, 53:292–294, 1947.

[39] P. Erdős and M. Kac. The Gaussian law of errors in the theory of additive number theoretic functions. *American Journal of Mathematics*, 62:738–742, 1940.

[40] P. Erdős and J. L. Selfridge. On a combinatorial game. *Journal of Combinatorial Theory, Series A*, 14:298–301, 1973.

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

[42] J. Gill. Computational complexity of probabilistic Turing machines. *SIAM J. Comput.*, 6:675–695, 1977.

[43] B.V. Gladkov. Sums of random variables, any r of which are independent. *Mat. Zametkii*, 32:385–399, 1982.

[44] B.V. Gladkov. A central limit theorem for sums of random variables, any r of which are independent. *Diskretnaia Mat.*, 1:22–28, 1989. English translation by V.A. Vatutin, in Discrete Mathematics and Applications, No. 1 (1991), pages 73–79.

[45] A. V. Goldberg, S. A. Plotkin, D. B. Shmoys, and É. Tardos. Using interior-point methods for fast parallel algorithms for bipartite matching and related problems. *SIAM J. Comput.*, 21(1):140–150, 1992.

[46] A. V. Goldberg, S. A. Plotkin, and P. M. Vaidya. Sublinear–time parallel algorithms for matching and related problems. *Journal of Algorithms*, 14(2):180–213, 1993.

[47] D. Yu. Grigoriev and M. Karpinski. The matching problem for bipartite graphs with polynomially bounded permanents is in NC. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 166–172, 1987. See also D. Yu. Grigoriev, M. Karpinski and M. F. Singer, Fast parallel algorithms for multivariate polynomial interpolation over finite fields, *SIAM J. Comput.*, Vol. 19, 1990, 1059–1063.

[48] L. K. Grover. Fast parallel algorithms for bipartite matching. In *Proceedings of the Integer Programming and Combinatorial Optimization Conference*, pages 367–384, 1992.

[49] J. Håstad. Pseudo-random generators under uniform assumptions. In *Proc. ACM Symposium on Theory of Computing*, pages 395–404, 1990.

[50] W. Hoeffding. Probability inequalities for sums of bounded random variables. *American Statistical Association Journal*, pages 13–30, 1963.

[51] M. Hofri. *Probabilistic Analysis of Algorithms.* Springer–Verlag, 1987.

[52] R. Impagliazzo, L. A. Levin, and M. Luby. Pseudorandom generation from one-way functions. In *Proc. ACM Symposium on Theory of Computing*, pages 12–24, 1989.

[53] R. Impagliazzo and D. Zuckerman. How to recycle random bits. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 248–253, 1989.

[54] A. Joffe. On a set of almost deterministic k–independent random variables. *The Annals of Probability*, 2(1):161–162, 1974.

[55] H. J. Karloff. A Las Vegas RNC algorithm for maximum matching. *Combinatorica*, 6(4):387–391, 1986.

[56] H. J. Karloff and P. Raghavan. Randomized algorithms and pseudorandom numbers. In *Proc. ACM Symposium on Theory of Computing*, pages 310–321, 1988.

[57] H. J. Karloff and D. B. Shmoys. Efficient parallel algorithms for edge coloring problems. *Journal of Algorithms*, 8:39–52, 1987.

[58] R. M. Karp. An introduction to randomized algorithms. *Discrete Applied Mathematics*, 34:165–201, 1991.

[59] R. M. Karp, N. Pippenger, and M. Sipser. A time-randomness trade-off. In *AMS Conference on Probabilistic Computational Complexity, Durham, NH*, 1985.

[60] R. M. Karp and V. Ramachandran. Parallel algorithms for shared-memory machines. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume A*, pages 871–941. Elsevier, New York, 1990.

[61] R. M. Karp, E. Upfal, and A. Wigderson. Constructing a perfect matching is in Random NC. *Combinatorica*, 6(1):35–48, 1986.

[62] R. M. Karp and A. Wigderson. A fast parallel algorithm for the maximal independent set problem. *J. Assoc. Comput. Mach.*, 32:762–773, 1985.

[63] P. W. Kastelyn. Graph theory and crystal physics. In F. Harary, editor, *Graph Theory and Theoretical Physics*, pages 43–110. Academic Press, New York, 1967.

[64] J. Köbler, U. Schöning, S. Toda, and J. Toran. Turing machines with few accepting computations and low sets for PP. *Journal of Computer and System Sciences*, 44:272–286, 1992.

[65] D. Koller and N. Megiddo. Constructing small sample spaces satisfying given constraints. In *Proc. ACM Symposium on Theory of Computing*, pages 268–267, 1993.

[66] D. C. Kozen. Semantics of probabilistic programs. *Journal of Computer and System Sciences*, 22:328–350, 1981.

[67] D. C. Kozen, U. V. Vazirani, and V. V. Vazirani. *NC* algorithms for comparability graphs, interval graphs, and testing for unique perfect matching. In *Proceedings, FST & TCS Conference*, Lecture Notes in Computer Science # 206, pages 496–503. Springer-Verlag, Berlin, 1985.

[68] C. Kruskal, L. Rudolph, and M. Snir. A complexity theory of efficient parallel algorithms. *Theoretical Computer Science*, 71:95–132, 1990.

[69] F. T. Leighton, B. Maggs, and S. Rao. Universal packet routing algorithms. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 256–269, 1988.

[70] G. F. Lev, N. Pippenger, and L. G. Valiant. A fast parallel algorithm for routing in permutation networks. *IEEE Transactions on Computers*, 30:93–100, 1981.

[71] N. Linial and N. Nisan. Approximate inclusion-exclusion. *Combinatorica*, 10(4):349–365, 1990.

[72] N. Linial and M. Saks. Decomposing graphs into regions of small diameter. In *Proc. ACM/SIAM Symposium on Discrete Algorithms*, pages 320–330, 1991.

[73] C. H. C. Little. An extension of Kastelyn's method of enumerating the 1–factors of planar graphs. In D. Holton, editor, *Combinatorial Mathematics, Proc. Second Australian Conference*, pages 63–72. Lecture Notes in Mathematics 403, Springer–Verlag, Berlin, 1974.

[74] L. Lovász and M. Plummer. *Matching Theory.* North–Holland, Amsterdam, 1986.

[75] M. Luby. A fast parallel algorithm for the maximal independent set problem. *SIAM J. Comput.*, 15(4):1036–1053, 1986.

[76] M. Luby. Removing randomness in parallel computation without a processor penalty. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 162–173, 1988.

[77] N. A. Lynch. Upper bounds for static resource allocation in a distributed system. *Journal of Computer and System Sciences*, 23:254–278, 1981.

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

[79] K. Mehlhorn and U. Vishkin. Randomized and deterministic simulations of PRAMs by parallel machines with restricted granularity of parallel memories. *Acta Informatica*, 21:339–374, 1984.

[80] G. L. Miller and J. Naor. Flow in planar graphs with multiple sources and sinks. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 112–117, 1989.

[81] R. Motwani, J. Naor, and M. Naor. The probabilistic method yields deterministic parallel algorithms. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 8–13, 1989.

[82] K. Mulmuley. Randomized geometric algorithms and pseudo-random generators. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 90–100, 1992.

[83] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.

[84] S.V. Nagaev and I.F. Pinelis. Some inequalities for the distribution of the sums of independent random variables. *Theory of Probability and its Applications*, 22:248–256, 1977.

[85] J. Naor and M. Naor. Small–bias probability spaces: efficient constructions and applications. In *Proc. ACM Symposium on Theory of Computing*, pages 213–223, 1990.

[86] H. Narayanan, H. Saran, and V. V. Vazirani. Randomized parallel algorithms for matroid union and intersection, with applications to arborescences and edge–disjoint spanning trees. In *Proc. ACM/SIAM Symposium on Discrete Algorithms*, pages 357–366, 1992.

[87] N. Nisan. Pseudorandom generators for space–bounded computation. In *Proc. ACM Symposium on Theory of Computing*, pages 204–212, 1990.

[88] N. Nisan. $RL \subseteq SC$. In *Proc. ACM Symposium on Theory of Computing*, pages 619–623, 1992.

[89] N. Nisan, E. Szemerédi, and A. Wigderson. Undirected connectivity in $O(\log^{1.5} n)$ space. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 24–29, 1992.

[90] N. Nisan and D. Zuckerman. More deterministic simulation in Logspace. In *Proc. ACM Symposium on Theory of Computing*, pages 235–244, 1993.

[91] A. Panconesi and A. Srinivasan. Fast randomized algorithms for distributed edge coloring. In *Proc. ACM Symposium on Principles of Distributed Computing*, pages 251–262, 1992.

[92] A. Panconesi and A. Srinivasan. Improved distributed algorithms for coloring and network decomposition problems. In *Proc. ACM Symposium on Theory of Computing*, pages 581–592, 1992.

[93] C. H. Papadimitriou and M. Yannakakis. The complexity of restricted spanning tree problems. *J. Assoc. Comput. Mach.*, 29(2):285–309, 1982.

[94] R. Paturi. On the degree of polynomials that approximate symmetric boolean functions. In *Proc. ACM Symposium on Theory of Computing*, pages 468–474, 1992.

[95] D. Peleg and E. Upfal. A time–randomness tradeoff for oblivious routing. *SIAM J. Comput.*, 19:256–266, 1990.

[96] N. Pippenger and J. Spencer. Asymptotic behavior of the chromatic index for hypergraphs. *Journal of Combinatorial Theory, Series A*, 51:24–42, 1989.

[97] S. A. Plotkin, D. B. Shmoys, and É. Tardos. Fast approximation algorithms for fractional packing and covering problems. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 495–504, 1991. Expanded version available as Technical Report 999, School of Operations Research and Industrial Engineering, Cornell University, 1992.

[98] M. O. Rabin. Probabilistic algorithm for testing primality. *Journal of Number Theory*, 12:128–138, 1980.

[99] M. O. Rabin. N–process mutual exclusion with bounded waiting by $4 \log_2 N$–valued shared variable. *Journal of Computer and System Sciences*, 25(1):66–75, 1982.

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

[101] P. Raghavan. Lecture notes on randomized algorithms. Technical Report RC 15340 (#68237), IBM T.J.Watson Research Center, January 1990. Also available as CS661 Lecture Notes, Technical report YALE/DCS/RR-757, Department of Computer Science, Yale University, January 1990.

[102] R. Raman. The power of Collision: Randomized parallel algorithms for chaining and integer sorting. In *Proceedings, 10th Annual FST & TCS Conference*, Lecture Notes in Computer Science # 472, pages 161–175. Springer-Verlag, Berlin, December 1990. Also available as University of Rochester CS Dept. TR 336, March 1990 (Revised January 1991).

[103] S. Ramanujan. Collected papers, 1927.

[104] A. Ranade. How to emulate shared memory. *Journal of Computer and System Sciences*, 41:307–326, 1991.

[105] H. Robbins. A remark on Stirling's formula. *Amer. Math. Monthly*, 62:26–29, 1955.

[106] V. Rödl. On a packing and covering problem. *European Journal of Combinatorics*, 5:69–78, 1985.

[107] M. Santha and U. V. Vazirani. Generating quasi-random sequences from semi-random sources. *Journal of Computer and System Sciences*, 33:75–87, 1986.

[108] J. P. Schmidt, A. Siegel, and A. Srinivasan. Chernoff-Hoeffding bounds for applications with limited independence. In *Proc. ACM/SIAM Symposium on Discrete Algorithms*, pages 331–340, 1993. Expanded version available as Technical Report TR 92-1305, Department of Computer Science, Cornell University, October 1992.

[109] J.P. Schmidt and A. Siegel. On aspects of universality and performance for closed hashing. In *Proc. ACM Symposium on Theory of Computing*, pages 355–366, 1989.

[110] J.P. Schmidt and A. Siegel. The analysis of closed hashing under limited randomness. In *Proc. ACM Symposium on Theory of Computing*, pages 224–234, 1990.

[111] L. J. Schulman. Small spaces uniform on neighborhoods. In *Proc. ACM Symposium on Theory of Computing*, pages 17–25, 1992.

[112] D. B. Shmoys, C. Stein, and J. Wein. Improved approximation algorithms for shop scheduling problems. In *Proc. ACM/SIAM Symposium on Discrete Algorithms*, pages 131–140, 1991.

[113] A. Siegel. Toward a usable theory of Chernoff Bounds for heterogeneous and partially dependent random variables. Manuscript, September 1992.

[114] A. Siegel. On universal classes of fast hash functions, their time-space tradeoff, and their applications. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 20–25, 1989.

[115] M. Sipser. Expanders, randomness or time vs. space. *Journal of Computer and System Sciences*, 36:379–383, 1988.

[116] R. Solovay and V. Strassen. A fast Monte Carlo test for primality. *SIAM J. Comput.*, 6(1):84–85, 1977. See also SIAM J. Comput., Volume 7, 1978.

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

[118] A. Srinivasan. On the distribution of the number of prime factors of integers. In preparation, 1993.

[119] C. Stein. *Approximation algorithms for multicommodity flow and shop scheduling problems.* Ph.D. dissertation, Laboratory for Computer Science, Massachusetts Institute of Technology, 1992. Available as $MIT/LCS/TR-550$.

[120] E. Styer and G. L. Peterson. Improved algorithms for distributed resource allocation. In *Proc. ACM Symposium on Principles of Distributed Computing*, pages 105–116, 1988.

[121] J. Tarui. Randomized polynomials, Threshold circuits and the Polynomial hierarchy. In *Annual Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science #480, pages 238–250, 1991.

[122] W. Thrash. A note on the least common multiples of dense sets of integers. Technical Report #93-02-04, Department of Computer Science & Engineering, University of Washington, Seattle, February 1993.

[123] P. Tiwari. Parallel algorithms for instances of linear matroid parity with small number of solutions. Technical Report RC 12766, IBM T.J.Watson Research Center, May 1987.

[124] S. Toda. PP is as hard as the Polynomial-time Hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991.

[125] M. Tompa. Lecture notes on probabilistic algorithms and pseudorandom generators. Technical Report #91-07-05, Department of Computer Science & Engineering, University of Washington, Seattle, July 1991.

[126] P. Turán. On a theorem of Hardy and Ramanujan. *Journal of the London Mathematics Society*, 9:274–276, 1934.

[127] P. M. Vaidya. Reducing the parallel complexity of certain linear programming problems. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 583–589, 1990.

[128] L. G. Valiant. A scheme for fast parallel communication. *SIAM J. Comput.*, 11:350–361, 1982.

[129] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[130] L. G. Valiant and G. J. Brebner. Universal schemes for parallel communication. In *Proc. ACM Symposium on Theory of Computing*, pages 263–277, 1981.

[131] L. G. Valiant and V. V. Vazirani. *NP* is as easy as detecting unique solutions. *Theoretical Computer Science*, 47(1):85–93, 1986.

[132] U. V. Vazirani. *Randomness, Adversaries and Computation.* Ph.D. dissertation, EECS, University of California at Berkeley, 1986.

[133] U. V. Vazirani and V. V. Vazirani. Random polynomial time is equal to slightly-random polynomial time. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 417–428, 1985. See also U. V. Vazirani and V. V. Vazirani, Random polynomial time is equal to semi-random polynomial time, Technical Report 88-959, Department of Computer Science, Cornell University, 1988.

[134] V. V. Vazirani. *NC* algorithms for computing the number of perfect matchings in $K_{3,3}$-free graphs and related problems. *Information and Computation*, 80:152–164, 1989.

[135] V. G. Vizing. On an estimate of the chromatic class of a p-graph. *Diskret. Anal.*, 3:25–30, 1964. In Russian.

[136] M.N. Wegman and J.L. Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, 22:265–279, 1981.

[137] D. J. A. Welsh. *Matroid Theory.* Academic Press, New York, 1976.

[138] A. Wigderson and D. Zuckerman. Expanders that beat the eigenvalue bound: Explicit construction and applications. In *Proc. ACM Symposium on Theory of Computing*, pages 245–251, 1993.

[139] A. C. Yao. Theory and applications of trapdoor functions. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 80–91, 1982.

[140] D. Zuckerman. Simulating BPP using a general weak random source. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 79–89, 1991.