Assorted results in boolean function complexity, uniform sampling and clique partitions of graphs

by

Jake Wellens

Submitted to the Department of Mathematics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Mathematics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2020

Author . . . .
Department of Mathematics
May 1, 2020

Certified by . . . .
Henry Cohn
Senior Principal Researcher at Microsoft Research New England and
Adjunct Professor of Mathematics at MIT
Thesis Supervisor

Accepted by . . . .
Jon A. Kelner
Chairman, Department Committee on Graduate Theses

Assorted results in boolean function complexity, uniform sampling and clique partitions of graphs

by

Jake Wellens

Submitted to the Department of Mathematics
on May 1, 2020, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Mathematics

Abstract

This thesis consists of three disparate parts. In the first, we generalize and extend recent ideas of Chiarelli, Hatami and Saks to obtain new bounds on the number of relevant variables for a boolean function in terms of its degree, its sensitivity, and its certificate and decision tree complexities, and we also sharpen the best-known polynomial relationships between some of these complexity measures by a constant factor. In the second part, we show that the Partial Rejection Sampling method of Guo, Jerrum and Liu can solve a handful of natural sampling problems that fall outside the guarantees of the authors' original analysis. Finally, we revise and make partial progress on a conjecture of De Caen, Erdős, Pullman and Wormald on clique partitions of a graph and its complement, building on ideas of Keevash and Sudakov.

Thesis Supervisor: Henry Cohn
Title: Senior Principal Researcher at Microsoft Research New England and Adjunct Professor of Mathematics at MIT

Acknowledgments

I owe a great deal of thanks to my thesis advisor, Henry Cohn, who was always ready and willing to discuss any mathematical thought, so long as I was ready and willing to make the trek from Building 2 to Microsoft. He allowed me the freedom to pick up any problem that sparked my interest, and was never short on suggestions for where to look next when I was ready to move on. In particular, he introduced me to partial rejection sampling, which led to the work in Chapter 3 of this thesis.

I'd also like to thank the other members of my thesis committee – Elchanan Mossel and Jon Kelner. Although they were some of the busiest people in the department, both Elchanan and Jon were always generous with their time, their mathematical wisdom and their encouragement.

To my officemates, Fred Koehler and John Urschel: thanks for enduring my boxes, my phone interviews, my "wall art" and my fidget spinner phase. Special thanks to John for convincing me to abandon PDEs in my first year and take Advanced Algorithms instead.

To my friends on the second floor, Vishesh Jain and Thao Do: thanks for all the math gossip and life advice, and for getting me to the gym. Without you two, my procrastination wouldn't have been half as productive and lively as it was.

To my friends who followed me to Cambridge from Pasadena, Jake Marcinek and Kevin Li: thanks for being, unfailingly, about that life.

To my most inspiring teachers, Peter Kaczmar and Chris Marx: thanks for teaching me how to see the forest and the trees simultaneously.

To my sister Kaitlin: thanks for always being the first mover and a role model.

To my mother Rose: thanks for your unwavering support, patience, honesty, humor and love. They don't make 'em like you anymore.

And to Lauren: thanks for being my biggest L and my biggest W.

Contents

Chapter 1

Introduction

Each of the three chapters of this thesis is, in terms of its mathematical content, almost entirely unrelated to the others – distant cousins at best. They are alike, however, in that they are comprised of refinements, generalizations and applications of clever mathematical ideas developed by other groups of people before me. We give a very brief overview of these ideas and what we do with them below – a more satisfactory introduction can be found at the beginning of each chapter.

Let us say a boolean function $f : \{0,1\}^n \to \{0,1\}$ is non-degenerate if it cannot be written as a function on $\{0,1\}^{n'}$ for $n' < n$. In 1992, Nisan and Szegedy [58] proved a lower bound on the degree of such functions as real multilinear polynomials. More specifically, if $f : \{0,1\}^n \to \{0,1\}$ is a non-degenerate function of degree d, then $n \leq d \cdot 2^{d-1}$. A recent paper of Chiarelli, Hatami and Saks [13] improved this bound to $n \leq 6.416 \cdot 2^d$, which is tight up to a constant factor. Their main idea was to consider the potential function $\sum_{i\in[n]} 2^{-\deg(D_i f)}$ instead of the quantity $\sum_{i\in[n]} \mathrm{Inf}_i[f]$, as in Nisan and Szegedy's original proof. In Chapter 2, we introduce similar potentials based on sensitivity and certificate complexity, as well as mixtures of these measures, obtaining new results of a similar flavor. Along the way, we sharpen their result by a constant, as well as many other best-known polynomial relationships between these measures.

In Chapter 3, we explore some applications of a recent paper by Guo, Jerrum and Liu [32], in which the authors introduced a general algorithmic framework called Partial Rejection Sampling for generating samples from product distributions conditional on a set of constraints being satisfied. It is shown in [32] that the procedure terminates with a uniform sample in an expected number of rounds which is logarithmic in the problem size, as long as (i) a Lovasz Local Lemma-like condition holds, and (ii) whenever two constraints overlap at all (in terms of the variables they depend on), they overlap significantly (loosely speaking). We apply their method to two sampling problems which do *not* satisfy the second condition, and yet we prove a logarithmic runtime bound for these problems.

Finally, in Chapter 4, we prove new bounds on the quantity

$$\max_{G \in \mathcal{G}_n} \operatorname{cp}(G) + \operatorname{cp}(\overline{G}),$$

where $\mathcal{G}_n$ is the set of all graphs on n vertices, and $\operatorname{cp}(G)$ is the *clique partition number* of G, which is defined as the minimal number r such that there exist r edge-disjoint cliques whose union is exactly G. (In particular, $\operatorname{cp}(G) \leq |E(G)|$, since $E(G)$ is an edge-disjoint family of 2-cliques whose union is G.) In a 1986 paper of De Caen, Erdős, Pullman and Wormald [18], the authors show that

$$\left(\frac{7}{25} + o(1)\right) n^2 \leq \max_{G \in \mathcal{G}_n} \operatorname{cp}(G) + \operatorname{cp}(\overline{G}) \leq \left(\frac{13}{30} + o(1)\right) n^2$$

and conjecture that the constant $\frac{7}{25}$ is optimal. We show that this is *not* true, by exhibiting an infinite family of graphs with $\operatorname{cp}(G) = \operatorname{cp}(\overline{G}) = \frac{1}{2}\left(\frac{7}{25} + \frac{1}{2050} + o(1)\right) n^2$. We also make use of a method of Keevash and Sudakov to improve the upper bound, showing that

$$\lim_{n \to \infty} \frac{\max_{G \in \mathcal{G}_n} \operatorname{cp}(G) + \operatorname{cp}(\overline{G})}{n^2} \in (0.28048, 0.3186).$$

(The content in Chapter 4 overlaps with a forthcoming joint work by the author, John Urschel and Dhruv Rohatgi [69]).

Chapter 2

On Some Extremal Properties of Boolean Functions

2.1 Introduction

In this chapter, we investigate several fundamental questions about boolean functions $f : \{0,1\}^n \to \{0,1\}$. Such functions are natural objects of study in many areas of computer science and mathematics, and therefore a fairly rich theory has developed around them in the past half century. Many important breakthroughs in active areas such as circuit complexity, PCPs and hardness of approximation, learning theory, cryptography and pseudorandomness – to name a few – are built upon an understanding of the combinatorial and analytic properties of boolean functions. For a proper introduction to the analysis of boolean functions and their role in theoretical computer science, we refer the reader to Ryan O'Donnell's excellent book ([61]).

Here, we concern ourselves not with these exciting applications, but rather with a few basic *extremal* questions about boolean functions. In particular, we focus on questions of the form, "if a boolean function f is small in measure A, then how large can it be in measure B?", for a variety of complexity measures A and B. In most of our results, the measure B is simply the number of *relevant variables*[1] of

[1] We say a variable $i \in [n]$ is *relevant* for f if flipping the ith bit can actually change the value of f for some inputs. More formally, a coordinate/variable i is relevant if there exists a pair of inputs

a function f, while the measure A belongs to a class of well-studied, polynomially related complexity measures, including $s(f)$, $\mathrm{bs}(f)$, $\deg(f)$, $C(f)$, $\mathrm{DT}(f)$, or *sensitivity, block sensitivity, degree, certificate complexity* and *decision tree/query complexity*, respectively[2]. These measures arise naturally when studying various simple models of computation. For example, the decision tree depth $\mathrm{DT}(f)$ measures the worst case number of bits of x made by the best adaptive query algorithm for computing $f(x)$. Perhaps a more interesting model is a CREW-PRAM (Concurrent Read Exclusive Write Parallel Random Access Machine), which has been studied by computer scientists since at least the early 1980's (see e.g. [10], [16], [68]). A CREW-PRAM for computing some function $f : \{0,1\}^n \to \{0,1\}$ consists of a collection of processors, computing synchronously in parallel, communicating via a global random access memory. Beginning with a string $x_1, \ldots, x_n$ stored in global memory cells $C_1, \ldots, C_n$, the computation must finish with $f(x_1, \ldots, x_n)$ written on cell C_1. At each step of computation, each processor can read any global memory cell, do some (arbitrary) private computing and then write to a global memory cell, with the restriction that at most one processor be allowed to write to a particular global memory cell in a single step (hence the "exclusive write"). In [16] and [68], Cook, Dwork and Reischuk show that any CREW-PRAM computing a function f requires at least $\Omega(\log s(f))$ steps in the worst case. Nisan [60] then introduced block sensitivity as a generalization of sensitivity to obtain a corresponding upper bound of $O(\log \mathrm{bs}(f))$ on CREW-PRAM complexity – hence tightness of the Cook-Dwork-Reischuk lower bound on CREW-PRAM complexity is equivalent to the statement $\mathrm{bs}(f) \leq s(f)^C$, for some constant $C < \infty$. Nisan, Szegedy and others ([58], [28], [11], e.g.) conjectured this to be true (eventually with $C = 2$). The so-called sensitivity conjecture remained open for some 25 years, until Hao Huang's recent proof ([40]) of $\mathrm{bs}(f) \leq s(f)^4$.

In light of the Cook-Dwork-Reischuk lower bound, a generic lower bound $s(f) \geq s(\mathcal{C})$, for all f in some function class $\mathcal{C}$, implies that all $f \in \mathcal{C}$ have CREW-PRAM complexity $\Omega(\log s(\mathcal{C}))$. The same is of course true for $\deg(f)$ or any of the other

x and x' in $\{0,1\}^n$ such that x and x' differ only in the ith coordinate and $f(x) \neq f(x')$.

[2] We give formal definitions of all of these quantities in section 2.2, but for a more comprehensive treatment, we recommend the surveys [11] and [35].

measures discussed above. The broadest possible class $\mathcal{C}$ of boolean functions for which it is possible to obtain nontrivial lower bounds is the set of all functions with n relevant variables, which we denote by $\mathcal{C}_n$. The following two foundational results in this direction, concerning $s(\mathcal{C}_n)$ and $\deg(\mathcal{C}_n)$, are due to H.U. Simon [72] and Nisan and Szegedy [58], respectively.

Theorem: (Simon, 1983) *For any $f \in \mathcal{C}_n$, with $s(f) = s$,*

$$n \leq \frac{s}{2} \cdot 4^s. \tag{2.1}$$

Theorem: (Nisan-Szegedy, 1994) *For any $f \in \mathcal{C}_n$, with $\deg(f) = d$,*

$$n \leq d \cdot 2^{d-1}. \tag{2.2}$$

Both theorems yield a $\Omega(\log \log n)$ lower bound on the CREW-PRAM complexity of $f \in \mathcal{C}_n$. This lower bound is tight, as witnessed by the address function

$$f(x_1, \ldots, x_k, \{y_z\}_{z \in \{0,1\}^k}) = y_{(x_1, \ldots, x_k)}$$

which has $n = 2^k + k$ relevant variables and $s(f) = \deg(f) = k + 1 = \log n + O(\log \log n)$. In fact, Wegener [75] gave an even tighter example of a *monotone* function g on $n = \Omega(\frac{1}{\sqrt{s(g)}} \cdot 4^{s(g)}) = \Omega(\frac{1}{\sqrt{\deg(g)}} \cdot 2^{\deg(g)})$ relevant variables. Hence, the logarithmic (in n) lower bounds obtained on $s(f)$ and $\deg(f)$ from (2.1) and (2.2) are tight up to a $1 + o(1)$ factor. However, viewed as upper bounds on n, (2.1) and (2.2) are not known to be asymptotically tight. In fact, Nisan and Szegedy's result was recently improved, for the first time, in [13]:

Theorem: (Chiarelli, Hatami, Saks 2019) *For any $f \in \mathcal{C}_n$, with $d := \deg(f)$,*

$$n \leq 6.614 \cdot 2^d. \tag{2.3}$$

The authors of [13] also construct an infinite family of functions of degree d with $n = (1.5 + o(1))2^d$ relevant variables, so the tightest possible bound on relevant variables in terms of degree is $n \sim C_{\text{deg}} \cdot 2^d$, for some $C_{\text{deg}} \in [1.5, 6.614]$. Computing the precise value of C_{deg} remains an open problem, one which we believe is inherently interesting, even though such improvements are inconsequential from the perspective of PRAM lower bounds.

Unlike the Nisan-Szegedy theorem, Simon's theorem has not been improved (as far as we are aware), nor have any tight examples been discovered. In particular, it does not appear to be known whether the extra factor of s can be removed from (2.1), which seems like a natural question to ask in light of the recent improvement to (2.2).

Similarly, many of the known polynomial relationships between the measures $s(f), \text{bs}(f), \deg(f), C(f)$ and $\text{DT}(f)$ have not been improved since the initial flurries of work in the 1990s and early 2000s. (The obvious exception being Huang's recent proof that $s(f)$ even belongs in this polynomial family!) There has, however, been more recent progress in constructing separations between these measures. For example, a classical result of Nisan [60] says that $C(f) \leq \text{bs}(f)^2$, and yet for many years the biggest gap exhibited by a known family of functions was $C(f) = \text{bs}(f)^{\log_{4.5} 5}$, until Gilmer, Saks and Srinivasan [27] showed in 2013 that 2 is the best possible exponent in this bound. One outstanding question in this area is that of $\text{bs}(f)$ versus $\deg(f)$. Nisan and Szegedy showed in [58] that

$$\text{bs}(f) \leq 2 \deg(f)^2, \tag{2.4}$$

and other than a constant-factor improvement to $\text{bs}(f) \leq \deg(f)^2$ by Tal [73], neither this upper bound nor the $\text{bs}(f) = \deg(f)^{\log_3 6}$ separation [59] has been improved in 25 years. The proof is completely analytic and makes use of V. A. Markov's inequality for polynomials on $\mathbb{R}$, while most of the best-known relationships between other complexity measures either have completely combinatorial/algorithmic proofs, or result from chaining together an algorithm with the bound $\text{bs}(f) \leq \deg(f)^2$. (Again, the notable exception is Huang's $\deg(f) \leq s(f)^2$, which is proved using spectral graph theory. Even so, the $\text{bs}(f) \leq s(f)^4$ corollary does go through (2.4)!) For example, the best-known bound on $\mathrm{DT}(f)$ in terms of $\deg(f)$, follows from an algorithm described in [55] which computes f using at most $\mathrm{bs}(f)\cdot\deg(f)$ queries, giving $\mathrm{DT}(f) \leq \deg(f)^3$ when combined with (2.4). Any progress on the $\mathrm{bs}(f)$ vs. $\deg(f)$ question is therefore interesting, in the author's opinion.

2.1.1 New Results

In this chapter, we make modest progress on the aforementioned problems. Our main results are summarized in the following theorem:

Main Theorem: *Let $f : \{0,1\}^n \to \{0,1\}$ be such that every $i \in [n]$ is relevant for f, and set $s := s(f)$, $d := \deg(f)$, $b := \mathrm{bs}(f)$, $C := C(f)$, and $D := \mathrm{DT}(f)$. Then*

$$n \leq \min\left\{4.394 \cdot 2^d,\ \frac{1}{2}4^C,\ 8.277 \cdot 2^{\frac{d}{2}+s},\ (\log s + 0.29) \cdot 2^{C+s}\right\}. \tag{2.5}$$

Moreover, for each k, the number of coordinates $i \in [n]$ which are not sensitive for any input x with $s_x(f) \geq k$ is at most $k^{2+o_k(1)}4^k$. If f is monotone, then

$$n \leq \min\left\{1.325 \cdot 2^d,\ \frac{1}{2} \cdot 4^s,\ \frac{1}{4} \cdot 2^D + 2\right\}. \tag{2.6}$$

In any case, for $\gamma := \sqrt{2/3} = 0.81649\ldots$, we also have

$$b \leq (\gamma + o(1)) \cdot d^2 \tag{2.7}$$

$$b \leq (\gamma + o(1)) \cdot s^4 \tag{2.8}$$

$$D \leq (\gamma + o(1)) \cdot d^3 \tag{2.9}$$

2.2 Preliminaries

Throughout the entire chapter, f will be a boolean function on $\{0,1\}^n$. We will refer to the input variables to such functions either by x_i or simply by the index i, for each $i \in \{1,\dots,n\} =: [n]$. We define $R(f)$ to be the set of *relevant variables* (sometimes *relevant coordinates*) for f, namely those $i \in [n]$ for which there exists a pair of inputs (x, x') such that $x_j = x'_j$ for all $j \neq i$ and $f(x) \neq f(x')$. Let $\delta_i(f)$ be the indicator of whether i is relevant for f. We also define $n(f) = |R(f)| = \sum_{i\in[n]} \delta_i(f)$ to be the number of relevant variables for f. We say a function f on $\{0,1\}^n$ is *non-degenerate* if $n(f) = n$, that is, every variable is relevant for f.

Sometimes it will be convenient to consider functions $g : \{\pm1\}^n \to \{\pm1\}$ instead of $f : \{0,1\}^n \to \{0,1\}$. Such sets of functions are clearly in bijection with one another, e.g. via the algebraic transformations

$$\begin{aligned} f(x_1,\dots,x_n) &\mapsto g(x) := \frac{1 - f(\frac{1-x_1}{2},\dots,\frac{1-x_n}{2})}{2} : \{\pm1\}^n \to \{\pm1\} && (2.10) \\ g(x_1,\dots,x_n) &\mapsto f(x) := 1 - 2g(1-2x_1,\dots,1-2x_n) : \{0,1\}^n \to \{0,1\} && (2.11) \end{aligned}$$

Functions on $\{\pm1\}^n$ can be expressed as a linear combination of characters χ_S for $S \subseteq [n]$, where $\chi_S(x) = \prod_{i\in S} x_i$. These characters form an orthonormal basis with respect to the inner product $\langle f, g\rangle := \frac{1}{2^n}\sum_{x\in\{\pm1\}^n} f(x)g(x) = \mathbb{E}[f(x)g(x)]$, and hence any $f : \{\pm1\}^n \to \mathbb{R}$ has a unique (Fourier) expansion of the form $f(x) = \sum_{S\subseteq[n]} \hat{f}(S)\prod_{i\in S} x_i$. (The coefficients $\hat{f}(S)$ are called the *Fourier coefficients* of f.) For each coordinate i, we define the ith directional derivative $D_i f$ via $D_i f(x) = x_i\frac{f(x)-f(x^i)}{2}$, from which it follows that $D_i f(x) = \sum_{S\ni i} \hat{f}(S)\chi_{S\setminus\{i\}}(x)$. The *ith coordinate influence* of a function f, denoted $\mathrm{Inf}_i[f]$, is defined to be

$$\mathrm{Inf}_i[f] := \Pr_{x\sim\{\pm1\}^n}[f(x) \neq f(x^i)]$$

and the *total influence* of f, denoted by $\mathbf{I}[f]$, is defined to be $\sum_{i=1}^n \mathrm{Inf}_i[f]$. Since $f(x) \neq f(x^i)$ precisely when $D_i f(x) \neq 0$, it follows that $\mathrm{Inf}_i[f] = \|D_i f\|_2^2$, recovering the well-known Fourier formulas for influence:

$$\mathrm{Inf}_i[f] = \sum_{S \ni i} \hat{f}(S)^2, \quad \mathbf{I}[f] = \sum_{S \subseteq [n]} |S| \hat{f}(S)^2$$

If f is monotone, then $\mathrm{Inf}_i[f] = \hat{f}(\{i\})$ and so $\mathbf{I}[f] = \sum_{i=1}^{n} \hat{f}(\{i\})$. The following useful fact can be observed directly from the definition of influence:

Fact 2.2.1. *For any $i \in [n]$, and any set $H \subset [n]$ with $i \notin H$,*

$$\mathrm{Inf}_i[f] = \mathbb{E}_{\alpha \sim \{0,1\}^H}[\mathrm{Inf}_i[f_\alpha]].$$

The Fourier expansion of a function f is the unique polynomial expansion of f in $\{\pm 1\}$-valued variables, and there is a corresponding (unique) polynomial for f over $\{0,1\}$, which we call the *multilinear polynomial expansion* of f. These two polynomials have the same degree, which we simply call the *degree of f*, denoted $\deg(f)$. From the Fourier formulas it is clear that $\mathbf{I}[f] \leq \deg(f)$. The following facts are well-known and easy to show by induction (see, e.g. [35] and [61]):

Fact 2.2.2. *Let f be a boolean function and let $\sum_{S \subseteq [n] c_S} \prod_{i \in S} x_i$ be its multilinear polynomial expansion over $\{0,1\}$. Then for all $S \subseteq [n]$:*

1. *$c_S \in \mathbb{Z}$, and $\hat{f}(S) \in \frac{1}{2^{\deg(f)}} \cdot \mathbb{Z}$*
2. *$c_S = (-2)^{|S|} \sum_{B : S \subseteq B \subseteq [n]} \hat{f}(B)$*
3. *$\hat{f}(S) = \sum_{B \subseteq [n], B' \subseteq S} c_B (-1)^{|B'|} (\frac{1}{2})^{-|B \setminus B'|}$*

We next define a variety of complexity measures we will encounter in this chapter. As mentioned at the start of the chapter, these concepts were originally introduced because of their close relationships with various models of computation. For any string $x \in \{0,1\}^n$ and a subset $S \subseteq [n]$, we let x^S denote the string obtained by flipping the bits of x belonging to S and leaving the rest alone. If $S = \{i\}$, we simply write x^i to denote x with the ith bit flipped. If $f(x) \neq f(x^i)$, we say that f is *sensitive* to i at x. The *sensitivity of f at an input x*, denoted $s_x(f)$, is the number of $i \in [n]$

for which f is sensitive to i at x. The maximum of $s_x(f)$ over all $x \in \{0,1\}^n$ is called the *sensitivity* of f (also *maximum sensitivity of* f), and is denoted $s(f)$. The *1-sensitivity* (resp. *0-sensitivity*) of f, denoted $s^1(f)$ (resp. $s^0(f)$), is the maximum of $s_x(f)$ over all inputs x with $f(x) = 1$ (resp. 0). Note that $\mathbf{I}[f] = \mathbb{E}_x[s_x(f)] \leq s(f)$.

The *block sensitivity at the point* x of a boolean function $f : \{0,1\}^n \to \{0,1\}$, denoted $\mathrm{bs}_x(f)$, is the maximum number k such that there exist k disjoint sets $B_1, \dots, B_k \subseteq [n]$ (called *blocks*) with the property that

$$f(x) = f(x^{B_i}), \text{ for } i = 1, \dots, k.$$

We then define the *block sensitivity* of f to be the maximum value of $\mathrm{bs}_x(f)$ over all $x \in \{0,1\}^n$, and we denote it by $\mathrm{bs}(f)$. If we restrict the blocks to be of size at most ℓ, the corresponding quantity is denoted $\mathrm{bs}_\ell(f)$. Clearly $\mathrm{bs}(f) \geq \mathrm{bs}_1(f) = s(f)$.

The *certificate complexity at the point* x of a boolean function f, denoted $C_x(f)$, is the size of the smallest set $S \subseteq [n]$ with the property that f is constant on the subcube of points which agree with x on S, i.e. $\{y : y_i = x_i \text{ for all } i \in S\}$. The *certificate complexity of* f, denoted $C(f)$, is then defined as the maximum value of $C_x(f)$ over all $x \in \{0,1\}^n$. Also, let $C_{\min}(f) := \min_{x \in \{0,1\}^n} C_x(f)$. By analogy with $s^0(f)$ and $s^1(f)$, we can also define $C^0(f)$, $C^1(f)$, $C^0_{\min}(f)$ and $C^1_{\min}(f)$ in the obvious way.

The *query complexity* or *(deterministic) decision tree complexity* of f, denoted $\mathrm{DT}(f)$, is defined to be the minimum cost of any deterministic, adaptive query algorithm which always computes f correctly. (The cost of such an algorithm is defined to be the maximal number of queries used by the algorithm to compute $f(x)$, taken over all $x \in \{0,1\}^n$.)

The *ε-approximate degree* of a boolean function $f : \{0,1\}^n \to \{0,1\}$ is the smallest d for which there exists a degree d (multilinear) polynomial $p(x_1, \dots, x_n)$ such that

$$|p(x) - f(x)| \leq \varepsilon \quad \text{for all } x \in \{0,1\}^n,$$

and we denote this quantity by $\widetilde{\deg}_\varepsilon(f)$. If we omit the ε and simply write $\widetilde{\deg}(f)$, it should be understood to mean $\widetilde{\deg}_{1/3}(f)$. This is the canonical and somewhat arbitrary choice – replacing 1/3 by any other constant can only change the value of $\widetilde{\deg}(f)$ by a constant factor.

The measures $\deg(f), \mathrm{bs}(f), C(f), \mathrm{DT}(f), \widetilde{\deg}(f)$ and (last but not least!) $s(f)$ are all known to be polynomially related.[3] Some of these relationships are known exactly, while for others there remains a gap between the best-known bound and the best-known separation. For example, Huang [40] recently proved the inequality $\deg(f) \leq s(f)^2$ for all f, and this is tight, as witnessed by the function

$$(x_{1,1} \wedge x_{2,1} \wedge \cdots x_{m,1}) \vee \cdots \vee (x_{1,m} \wedge x_{2,m} \wedge \cdots x_{m,m})$$

which has sensitivity m and degree m^2. On the other hand, the relationship $s(f) \leq \deg(f)^2$ is not known to be tight – the best known construction has $s(f) = \deg(f)^{\log_3 6}$. We summarize the current state of knowledge about these relationships in Table 2.1. Below, we list the facts that "generate" the left table, in the sense that any inequality implied by Table 2.1 can be proved by combining these inequalities in the proper sequence.

Fact 2.2.3. $\mathrm{DT}(f) \leq C(f)^2$ (Blum-Impagliazzo [8])
$s(f) \leq \mathrm{bs}(f) \leq C(f) \leq \mathrm{bs}(f)^2$ (Nisan [60])
$\mathrm{bs}(f) \leq \deg(f)^2$ (Nisan-Szegedy [58], refined by Tal [73])
$\mathrm{DT}(f) \leq \mathrm{bs}(f) \deg(f)$ (Midrijanis [55])

Fact 2.2.4 (Nisan, [60])**.** *For monotone boolean functions f, $s(f) = bs(f) = C(f) \leq \deg(f)$.*

Next we describe a construction of Wegener [75], which is a monotone function whose (block) sensitivity, degree, certificate and query complexity are all quite low compared to the number of variables. For each odd integer $k \geq 1$, we define the

[3] A number of other complexity measures, such as randomized and quantum decision tree complexities (with 0, 1 or 2-sided error) also fit into this "polynomial family", but they do not play a significant role in this thesis so we do not define them here. These definitions can be found in [11], for example.

	$s(f)$	$\deg(f)$	$\mathrm{bs}(f)$	$C(f)$
$s(f)$	$*$	2	1	1
$\deg(f)$	2	$*$	2	2
$\mathrm{bs}(f)$	4	2	$*$	1
$C(f)$	5	3	2	$*$
$\mathrm{DT}(f)$	6	3	3	2

	$s(f)$	$\deg(f)$	$\mathrm{bs}(f)$	$C(f)$
$s(f)$	$*$	$\log_3 6$ [59]	1	1
$\deg(f)$	2	$*$	2	2
$\mathrm{bs}(f)$	2	$\log_3 6$ [59]	$*$	1
$C(f)$	2.22 [6]	$\log_3 6$ [59]	2 [27]	$*$
$\mathrm{DT}(f)$	3 [6]	2 [26]	2	2

Table 2.1: Best-known relationships (**left**) and separations (**right**) between complexity measures. **On the left**: A number α in the row labeled by measure A and the column labeled by measure B means that for all boolean functions f, $A(f) \leq B(f)^{\alpha}$. So for example the 4 in the (bs, s) entry means $\mathrm{bs}(f) \leq s(f)^4$ for all f. **On the right**: A number β in the row labeled by measure A and the column labeled by measure B means that there exists (infinitely many) functions f with $A(f) = \tilde{\Omega}(B(f)^{\beta})$. The corresponding reference is given in square brackets. For example, the 2 in the (bs, s) entry means we know of an explicit family of functions with $\mathrm{bs}(f) = \Omega(s(f)^2)$. Note that we do not include a column for $\mathrm{DT}(f)$, since $\mathrm{DT}(f) \geq A(f)$ for any measure A considered here.

monotone address function

$$\mathrm{MAF}_k\left(x_1, \dots, x_k, \{y_S\}_{S \in \binom{[k]}{\lceil k/2 \rceil}}\right) := \mathrm{MAJ}(x_1, \dots, x_k) \bigvee_{S \in \binom{[k]}{\lceil k/2 \rceil}} \left(\bigwedge_{i \in S} x_i \wedge y_S\right).$$

Proposition 2.2.5. *The monotone address function $f = \mathrm{MAF}_k$ has $s(f) = \mathrm{bs}(f) = C(f) = \lceil k/2 \rceil + 1$, and $\deg(f) = \mathrm{DT}(f) = k+1$. Therefore, for $m \in \{s(\cdot), \mathrm{bs}(\cdot), C(\cdot)\}$ and $m' \in \{\deg(\cdot), \mathrm{DT}(\cdot)\}$, f has $n(f) = \Theta\left(\frac{1}{\sqrt{m(f)}} \cdot 4^{m(f)}\right) = \Theta\left(\frac{1}{\sqrt{m'(f)}} \cdot 2^{m'(f)}\right)$ relevant variables.*

Proof. The fact that $s(f) = \lceil k/2 \rceil + 1$ can be seen through a direct case analysis – we refer the reader to the proof in [75] for details. By Fact 2.2.4, this implies the same for $\mathrm{bs}(f)$ and $C(f)$. To compute $\deg(f)$, note that we can write

$$\mathrm{MAF}_k(x, y) = \sum_{S \in \binom{[k]}{\lceil k/2 \rceil}} y_S \prod_{i \in S} x_i \cdot \prod_{i \notin S} (1 - x_i) + \underbrace{\mathbf{1}\left(\sum_{i=1}^{k} x_i \geq k/2 + 1\right)}_{\deg(\cdot) \leq k},$$

and so each y_S appears in a unique degree $k+1$ monomial $y_S \cdot x_1 \cdots x_k$. Since $\mathrm{DT}(f) \geq \deg(f)$, and clearly $\mathrm{DT}(f) \leq k+1$ (f can computed by querying the x variables and then possibly the unique relevant y_S), it follows that $\mathrm{DT}(f) = k+1$ as well. The conclusion then follows from Stirling's formula. □

2.3 Improved bounds on the number of variables

2.3.1 Overview

Our goal is to generalize the ideas of Chiarelli, Hatami and Saks [13] to develop a unified framework for proving bounds on $n(f)$ in terms of various complexity measures like $\deg(f)$, $s(f)$ and $C(f)$. The key player in each proof is a certain "coordinate version" m_i of each complexity measure m, which is engineered to behave in a certain way with respect to restrictions of variables (see Definition 2.3.1). We call such m_i "restriction reducing coordinate measures" (RRCMs) and form the corresponding potential functions

$$\mathbf{M}(f) := \sum_{i\in[n]} \frac{\delta_i(f)}{2^{m_i(f)}}. \tag{2.12}$$

The defining properties of RRCMs are chosen to guarantee that, for any $H \subseteq [n]$, $\mathbf{M}$ always obeys the inequality

$$\mathbf{M}(f) \leq \sum_{i\in H} \frac{\delta_i(f)}{2^{m_i(f)}} + \mathbb{E}_{\alpha\sim\{0,1\}^H}[\mathbf{M}(f_\alpha)]. \tag{2.13}$$

This enables us to bound $\mathbf{M}(f)$ recursively, assuming we choose the set of coordinates H in such a way that the restrictions f_α are guaranteed to have *lower complexity*, in some sense. Upper bounds on $\mathbf{M}(f)$ naturally yield exponential upper bounds on $n(f)$ in terms of $m(f)$. We make these definitions precise below in the next subsection, and each subsequent subsection describes a different implementation of the general strategy above, yielding new bounds.

2.3.2 Restriction-reducing coordinate measures

Let us say a functional m on boolean functions is an *i-coordinate measure* if $\delta_i(f) = 0 \implies m(f) = 0$.

Definition 2.3.1. *We say an i-coordinate measure m_i is* restriction reducing *if, for any $j \in [n] \setminus \{i\}$, and each $b \in \{0, 1\}$:*

(1) $m_i(f_{j=b}) \leq m_i(f)$

(2) if $\delta_i(f) = 1$ and $\delta_i(f_{j=b}) = 0$, then $m_i(f_{j=1-b}) \leq m_i(f) - 1$.

We denote by $\mathcal{R}_i$ the set of restriction reducing i-coordinate measures. We abuse notation sightly and write $\{m_i\} \in \mathcal{R}_i$ to denote that $m_i \in \mathcal{R}_i$ for each $i \in [n]$. Properties (1) and (2) were essentially chosen to make the following a fact:

Fact 2.3.2. *Let $m_i \in \mathcal{R}_i$, and let $j \in [n] \setminus \{i\}$. Then*

$$\delta_i(f)2^{-m_i(f)} \leq \frac{\delta_i(f_{j=0})2^{-m_i(f_{j=0})} + \delta_i(f_{j=1})2^{-m_i(f_{j=1})}}{2}. \tag{2.14}$$

Proof. If $\delta_i(f_{j=0}) = \delta_i(f_{j=1}) = 1$, then property (1) of Definition 2.3.1 implies that both $2^{-m_i(f_{j=0})} \geq 2^{-m_i(f)}$ and $2^{-m_i(f_{j=1})} \geq 2^{-m_i(f)}$, which implies (2.14). Otherwise, suppose without loss of generality that $\delta_i(f_{j=0}) = 0$ and $\delta_i(f_{j=1}) = 1$. Then property (2) of Definition 2.3.1 implies that $2^{-m_i(f_{j=1})} \geq 2 \cdot 2^{-m_i(f)}$ which also implies (2.14). □

Fact 2.3.2 extends easily by induction to larger restrictions:

Fact 2.3.3. *For any $i \in [n]$ and any $H \subset [n]$ with $i \notin H$, and any $\{m_i\} \in \mathcal{R}_i$,*

$$\delta_i(f)2^{-m_i(f)} \leq \mathbb{E}_{\alpha \sim \{0,1\}^H} \left[\delta_i(f_\alpha)2^{-m_i(f_\alpha)}\right]. \tag{2.15}$$

Proof. We proceed by induction on $|H|$. The base case $H = \{j\}$ is Fact 2.3.2. For the inductive step, observe that if $\delta_i(f)2^{-m_i(f)} \leq \mathbb{E}_{\alpha \sim \{0,1\}^H} \left[\delta_i(f_\alpha)2^{-m_i(f_\alpha)}\right]$ holds for all f with $H = H_1$ or H_2, then it holds for $H = H_1 \sqcup H_2$, since

$$\begin{aligned}
\delta_i(f)2^{-m_i(f)} &\le \mathbb{E}_{\alpha_1 \sim \{0,1\}^{H_1}}\left[\delta_i(f_{\alpha_1})2^{-m_i(f_{\alpha_1})}\right] \\
&\le \mathbb{E}_{\alpha_1 \sim \{0,1\}^{H_1}}\left[\mathbb{E}_{\alpha_2 \sim \{0,1\}^{H_2}}\left[\delta_i(f_{\alpha_1,\alpha_2})2^{-m_i(f_{\alpha_1,\alpha_2})}\right]\right] \\
&= \mathbb{E}_{\alpha \sim \{0,1\}^{H_1 \sqcup H_2}}\left[\delta_i(f_{\alpha})2^{-m_i(f_{\alpha})}\right].
\end{aligned}$$

□

For any $\{m_i\} \in \mathcal{R}_i$, we can define the associated potential function $\mathbf{M}$ via equation (2.12). By Fact 2.3.3, $\mathbf{M}$ satisfies the inequality (2.13) for any set $H \subseteq [n]$ of restricted coordinates. Next we introduce three explicit families of RRCMs, the first of which ($\deg_i$) was introduced in [13]:

Definition 2.3.4. *For each $i \in [n]$, define the i-coordinate measures*

$$\begin{aligned}
\deg_i(f) &:= \deg(f(x) - f(x^i)) && (2.16) \\
\mathrm{sens}_i(f) &:= \max_{\{x\,:\,f(x)\neq f(x^i)\}} s_x(f) + s_{x^i}(f) && (2.17) \\
\mathrm{cert}_i(f) &:= \max_{\{x\,:\,f(x)\neq f(x^i)\}} C_x(f) + C_{x^i}(f) && (2.18)
\end{aligned}$$

Lemma 2.3.5. *For each $i \in [n]$, the coordinate measures $\deg_i$, sens_i, and cert_i all belong to $\mathcal{R}_i$.*

Proof. Since $\deg(\cdot)$, $s_x(\cdot)$ and $C_x(\cdot)$ cannot possibly increase by restricting input variables, property (1) of Definition 2.3.1 is trivially satisfied for each of the coordinate measures in question. To see that (2) holds, we abbreviate $f_{j=b}$ by f_b and assume without loss of generality that $\delta_i(f_0) = 0$.

First we argue that $\deg_i(f_1) = \deg_i(f) - 1$. We can write $f(x) = x_j f_1(x) + (1 - x_j)f_0(x)$. Since x_i does not appear in $(1 - x_j)f_0(x)$, it follows that $f(x) - f(x^i) = x_j(f_1(x) - f_1(x^i))$ from which it is clear that $\deg_i(f) = 1 + \deg_i(f_1)$.

Next we argue $\mathrm{sens}_i(f_1) = \mathrm{sens}_i(f) - 1$. Let x be any input for which $f(x) \neq f(x^i)$, and let us write y for the string which is x with the jth bit omitted. Since f_0 does not depend on i, it must be that $f_0(y^i) = f_0(y)$. Therefore all such x must have $x_j = 1$, so $f_1(y) = f(x) \neq f(x^i) = f_1(y^i)$. But then j must be sensitive for f at exactly one of x^i or x, hence $s_x(f) + s_{x^i}(f) = s_x(f_1) + s_{x^i}(f_1) + 1$.

Finally we argue $\mathrm{cert}_i(f_1) = \mathrm{cert}_i(f) - 1$, which essentially follows from the previous paragraph. Indeed, as above, all x for which i is sensitive for f must have $x_j = 1$, and j must be sensitive for exactly one of x or x^i – suppose it is x (wlog). Then any certificate for f which agrees with x must assign 1 to x_j, since if it were allowed to be flipped, the certificate could not make f constant. The claim follows. □

Lemma 2.3.6. *Let m_i be a restriction reducing i-coordinate measure and set $r := \min\{m_i(x \mapsto x_i), m_i(x \mapsto \neg x_i)\}$. Then for any boolean function f,*

$$\delta_i(f) 2^{-m_i(f)} \leq 2^{-r} \cdot \mathrm{Inf}_i[f]. \tag{2.19}$$

Hence $\mathbf{M}(f) \leq 2^{-r} \cdot \mathbf{I}[f]$ and for any $k \in \mathbb{N}$, at most $\mathbf{I}[f] \cdot 2^{k-r}$ relevant variables can have $m_i(f) \leq k$.

Proof. We proceed by induction on $n(f)$. If $n(f) = 1$, then the corollary follows from the definition of r and the fact that $\mathrm{Inf}_i[f] \leq 1$. Now suppose the desired inequality holds for all f' with $n(f') < n(f)$, and we wish to show it holds for f as well. Then by the induction hypothesis and Fact 2.3.2,

$$\delta_i(f) 2^{-m_i(f)} \leq \frac{2^{-r} \cdot \mathrm{Inf}_i[f_{j=0}] + 2^{-r} \cdot \mathrm{Inf}_i[f_{j=1}]}{2} = 2^{-r} \cdot \mathrm{Inf}_i[f] \tag{2.20}$$

where the final equality is Fact 2.2.1. If we sum this inequality over $i \in R(f)$, we obtain

$$\mathbf{M}(f) = \sum_{k=0}^{\infty} \frac{|\{j \in R(f) \,:\, m_i(f) = k\}|}{2^j} \leq 2^{-r} \mathbf{I}[f] \tag{2.21}$$

which in particular implies that at most $\mathbf{I}[f] \cdot 2^{k-r}$ variables in $R(f)$ have $m_i(f) \leq k$. □

Observation 2.3.1. Applying Lemma 2.3.6 to the measures $\deg_i$ and sens_i immediately yields both Nisan-Szegedy's and Simon's theorems. Indeed, $\min\{\deg_i(x \mapsto x_i), \deg_i(x \mapsto \neg x_i)\} = 1$ and $\min\{\text{sens}_i(x \mapsto x_i), \text{sens}_i(x \mapsto \neg x_i)\} = 2$, so

$$\begin{aligned} n(f) &\leq \mathbf{I}[f] \cdot 2^{\deg(f)-1} &(2.22)\\ n(f) &\leq \mathbf{I}[f] \cdot 4^{s(f)-1}. &(2.23) \end{aligned}$$

2.3.3 Degree

Let $\mathbf{D}(f) := \sum_{i\in[n]} \frac{\delta_i(f)}{2^{\deg_i(f)}}$, and for any $H \subseteq [n]$, let $\mathbf{D}(H, f) = \sum_{i\in H} \frac{\delta_i(f)}{2^{\deg_i(f)}}$. For any $d \in \mathbb{N}$, let $\mathbf{D}_d = \max_{\{f\,:\,\deg(f)\leq d\}} \mathbf{D}(f)$. In [13], the authors argue that one can always find a set H of $\leq \deg(f)^3$ coordinates such that (i) $\deg_i(f) = \deg(f)$ $\forall i \in H$ and (ii) $\deg(f_\alpha) < \deg(f)$ for all $\alpha \in \{0,1\}^H$. This implies $\mathbf{D}_d \leq \frac{d^3}{2^d} + \mathbf{D}_{d-1}$, and hence that $\mathbf{D}(f) < \sum_{d=1}^{\infty} \frac{d^3}{2^d} = 26$ for all f. Combined with the observation that $\mathbf{D}_d \leq \frac{d}{2}$ (see Lemma 2.3.6), this yields Chiarelli, Hatami and Saks' final bound $\mathbf{D}(f) \leq \frac{11}{2} + \sum_{d=12}^{\infty} \frac{d^3}{2^d} \approx 6.614$.

In this subsection, we implement their argument in a slightly different way to obtain a slightly stronger bound. In particular, rather than choosing H to be a minimal set of coordinates which covers all max degree monomials in f, we choose H to be the variables in a *single monomial* of f. Restricting this set of coordinates may not reduce the degree of f, but as shown below, it *will* reduce the block sensitivity of f. Hence, as we'll want to induct on both degree and block sensitivity simultaneously, we define

$$\mathbf{D}_{b,d} := \max_{\substack{f \text{ with } \text{bs}(f)\leq b \\ \text{and } \deg(f)=d}} \mathbf{D}(f).$$

We also define $B_d := \max_{\deg(f)=d} \text{bs}(f)$, and make the convention that $\mathbf{D}_{b,d} = 0$ whenever $b > B_d$.

Lemma 2.3.7. *If M is a monomial of degree $d = \deg(f)$ which appears in f with non-zero coefficient, then for any assignment $\alpha : M \to \{0,1\}$, the restricted function f_α has $\text{bs}(f_\alpha) \leq \text{bs}(f) - 1$.*

Proof. Let us write any string $x \in \{0,1\}^n$ as $x = (x_M, y)$, where $x_M \in \{0,1\}^M$ and $y \in \{0,1\}^{[n]\setminus M}$. We claim that for any (x_M, y), there is always a sensitive block for f contained entirely in M. Indeed, for any y, the function $f(\cdot, y)$ has degree d, since nothing can cancel with the maximal monomial $\prod_{i \in M} x_i$. In particular, it is not constant, so for any input x_M, there is always at least one sensitive block for $f(\cdot, y)$ at x_M. Therefore, $\mathrm{bs}_y(f_\alpha) + 1 \leq \mathrm{bs}_{(\alpha,y)}(f)$, and the lemma follows. □

Lemma 2.3.8. *For each b, d with $b \leq d^2$, we have*

$$\mathbf{D}_{b,d} \leq d \cdot 2^{-d} + \max_{k \in \{1,\ldots,d\}} \mathbf{D}_{b-1,k}$$

Proof. Suppose f has $\deg(f) = d$ and $\mathrm{bs}(f) \leq b$. Let M be any degree d monomial in f. Using (2.13),

$$\mathbf{D}(f) \leq \underbrace{|M| \cdot 2^{-d}}_{= d \cdot 2^{-d}} + \mathop{\mathbb{E}}_{\alpha \sim \{0,1\}^M}[\mathbf{D}(f_\alpha)]. \tag{2.24}$$

By Lemma 2.3.7, each f_α has $\mathrm{bs}(f_\alpha) \leq b - 1$. Since $\mathbf{D}_{b,d}$ is monotone in b (for feasible $b \leq d^2$), it follows that for each α, $\mathbf{D}(f_\alpha) \leq \mathbf{D}_{b-1,k}$, where $k = \deg(f_\alpha)$. Taking the maximum over all values of $k \in \{1, \ldots, d\}$ yields a uniform bound that holds for all restrictions f_α. □

Corollary 2.3.9. *For every f, and every $d \geq 1$,*

$$\begin{aligned} \mathbf{D}(f) \;&\leq\; \left(\mathbf{D}_{B_d,d} + \frac{(d+1)B_{d+1}}{2^{d+1}} + \sum_{k=d+2}^{\infty} \frac{k(B_k - B_{k-1})}{2^k} \right) && (2.25) \\ &\leq\; \left(\mathbf{D}_{d^2,d} + \frac{(d+1)^3}{2^{d+1}} + \sum_{k=d+2}^{\infty} \frac{2k^2 - k}{2^k} \right). && (2.26) \end{aligned}$$

Lemma 2.3.8 yields explicit bounds on $\mathbf{D}_{b,d}$ for any finite (b, d), which in turn yields an explicit bound on $\mathbf{D}(f)$ for any f via Corollary 2.3.9. Incorporating the influence bound $\mathbf{D}_{b,d} \leq \frac{d}{2}$, we build up a table of upper bounds $D(b, d)$ recursively, using the rule

$$D(b,d) = \begin{cases} \min\left\{\frac{d}{2},\ \max_{k\in\{1,\ldots,d\}}\left\{d\cdot 2^{-d} + D(b-1,k)\right\}\right\} & \text{for } b \le B_d \\ 0 & \text{for } b > B_d \end{cases} \tag{2.27}$$

Supposing $B_d = d^2$ and extracting bounds recursively already shows that $\mathbf{D}(f) < 5.0782$, but we can further improve this by obtaining sharper upper bounds on B_d. For values of $d \le 14$ (which contribute the most to $D(b,d)$ anyway), we can obtain such bounds by manually checking feasibility of a certain linear program, as shown below. (This reduction is partially inspired by ideas of Nisan and Szegedy in [58].)

Fact 2.3.10. *If there exists a function $f : \{0,1\}^n \to \{0,1\}$ of degree d with block sensitivity b, then there exists another function $g : \{0,1\}^b \to \{0,1\}$ of degree $\le d$ with $g(0) = 0$ and $g(w) = 1$ for each vector w of hamming weight 1.*

Proof. If $f(x)$ attains maximal block sensitivity at z, then $f(x \oplus z)$ attains maximal block sensitivity at 0, so without loss of generality we may assume $z = 0$, and possibly replacing f by $1 - f$ we may also assume that $f(0) = 0$. If $B_1, \ldots, B_b$ are sensitive blocks for f at 0, then define

$$g(y_1, \ldots, y_b) = f(\underbrace{y_1, \ldots, y_1}_{B_1}, \ldots, \underbrace{y_b, \ldots, y_b}_{B_b})$$

so that for each coordinate vector e_i, $g(e_i) = f(\mathbf{1}_{B_i}) = f(0^{B_i}) = 1$. □

For any $d \ge 1$, define the moment map $m_d : \mathbb{R} \to \mathbb{R}^d$ by $m(t) = (t, t^2, \ldots, t^d)$.

Proposition 2.3.11. *If there exists a degree d function $f : \{0,1\}^n \to \{0,1\}$ with block sensitivity b, then there exists $\tau \in \{0,1\}$ such that the following set of linear inequalities has a solution $p \in \mathbb{R}^d$:*

$$\begin{array}{rcl} \langle p, m_d(1)\rangle & = & 1 \\ 0 \le \langle p, m_d(k)\rangle & \le & 1 \ \ \textit{for each } k \in \{2,\ldots,b-1\} \\ \langle p, m_d(b)\rangle & = & \tau \end{array} \tag{2.28}$$

d	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$B_d \leq$	1	3	6	10	15	21	29	38	47	58	71	84	99	114

Table 2.2: LP bounds on block sensitivity for low degree functions.

Proof. If such an f exists, then let $q(x_1, \dots, x_b) = \frac{1}{b!}\sum_{\sigma \in S_b} g(x_{\sigma(1)}, \dots, x_{\sigma(b)})$, where g comes from Fact 2.3.10, and set $\tau = g(1, 1, \dots, 1)$. It is well known (see [11]) that there is a univariate polynomial $p : \mathbb{R} \to \mathbb{R}$ of degree at most d such that for any $x \in \{0,1\}^b$, $q(x_1, \dots, x_b) = p(x_1 + \dots + x_b)$. For each $k \in \{1, \dots, b\}$, $p(k)$ is therefore the average value of g on boolean vectors with hamming weight k, so in particular $p(k) \in [0, 1]$. We also know $p(0) = g(0) = 0$, $p(b) = g(1, \dots, 1) = \tau$, and $p(1) = \frac{1}{n}\sum_i g(e_i) = 1$, and hence the coefficients of p provide a solution to the set of linear inequalities. □

Using the simplex method with exact (rational) arithmetic in Maple, we compute the largest b for which the LP (2.28) is feasible for $1 \leq d \leq 14$, which yields upper bounds on B_d for small d. These bounds are summarized in Table 2.2. Recomputing the table $D(b, d)$ with B_d given by Table 2.2 for $d \leq 14$ (and $B_d = d^2$ for $d > 14$), we can recompute the table as in (2.27) with these boundary conditions. This time $D(30^2, 30) \leq 4.4157\dots$, which implies

$$\mathbf{D}(f) \leq 4.4158 \tag{2.29}$$

for all f. If we incorporate the main result of Section 2.4, which implies that

$$B_d^2 - B_d \leq \frac{2}{3}(d^4 - d^2)$$

into the table $D(b, d)$, we obtain the slightly stronger result $\mathbf{D}_\infty \leq 4.3935$, which implies

Theorem 2.3.12. *For all f, $n(f) \leq 4.3935 \cdot 2^{\deg(f)}$.*

2.3.4 Certificate complexity

Now let us define the analogous quantities for certificate complexity. Let $\mathbf{C}(f) := \sum_{i\in[n]} \frac{\delta_i(f)}{2^{\mathrm{cert}_i(f)}}$, and for any $H \subseteq [n]$, let $\mathbf{C}(H, f) = \sum_{i\in H} \frac{\delta_i(f)}{2^{\mathrm{cert}_i(f)}}$. For any $d \in \mathbb{N}$, we also define $\mathbf{C}_d = \max_{\{f\,:\,\deg(f)\le d\}} \mathbf{C}(f)$.

Theorem 2.3.13. *For any $d \geq 1$, $\mathbf{C}_d \leq \frac{1}{2}$.*

Proof. Let f be a boolean function with $\deg(f) = d$. For any certificate C for f, let H be the set of variables fixed by C. It follows from (2.13) that

$$\mathbf{C}(f) \leq \mathbf{C}(H, f) + \mathbb{E}_{\alpha\sim\{0,1\}^H}[\mathbf{C}(f_\alpha)]. \tag{2.30}$$

Since C is a certificate, we know $\deg(f_\alpha) \leq d-1$ for all α, and $\deg(f_{\alpha^*}) = 0$ for *some* $\alpha^* \in \{0,1\}^H$. So, $\mathbf{C}(f_{\alpha^*}) = 0$, and we can improve (2.30) to

$$\mathbf{C}(f) \leq \mathbf{C}(H, f) + \left(1 - 2^{-|C|}\right) \mathbf{C}_{d-1}. \tag{2.31}$$

Now take C to be the *globally smallest certificate* for f, so that $C_i(f) \geq 2|H|$ for all $i \in R(f)$, and in particular

$$\mathbf{C}(f) \leq |H| \cdot 4^{-|H|} + \left(1 - 2^{-|H|}\right) \mathbf{C}_{d-1}. \tag{2.32}$$

Since $c \cdot 2^{-c} \leq \frac{1}{2}$ for $c \geq 1$, inequality (2.32) implies that $\mathbf{C}_d \leq \alpha \cdot \frac{1}{2} + (1-\alpha) \cdot \mathbf{C}_{d-1}$ for some $\alpha \in [0,1]$. Therefore if $\mathbf{C}_{d-1} \leq \frac{1}{2}$ for some d, then also $\mathbf{C}_d \leq \frac{1}{2}$. The theorem then follows by induction on d, noting that $\mathbf{C}_1 = \frac{1}{4} < \frac{1}{2}$. □

Since $\mathbf{C}(x \mapsto x_1) = \frac{1}{4}$, Theorem 2.3.13 cannot be improved by more than a factor of 2. In any case, we have the following immediate corollary:

Theorem 2.3.14. *For any f, $n(f) \leq \frac{1}{2} \cdot 4^{C(f)}$.*

Finally, we use an implementation similar to the one above to give a proof of a stronger bound on $n(f)$ in terms of $\deg(f)$ for *monotone* functions f.

Theorem 2.3.15. *For monotone functions f, $n(f) \leq 1.325 \cdot 2^{\deg(f)}$.*

Proof. We let $\widetilde{\mathbf{D}}_d$ denote the maximum value of $\mathbf{D}(f)$ over all monotone functions of degree at most d. Given a monotone f of degree d, let H be the variables fixed by any minimal 0-certificate C. By monotonicity, $f(0_H, 1_{\overline{H}}) \equiv 0$, so by minimality of H, each $i \in H$ must be sensitive for f at the input $(0_H, 1_{\overline{H}})$. Therefore restricting the variables in $\overline{H}$ to 1 yields an OR function on H, and hence each $i \in H$ has $\deg_i(f) \geq |H|$. If we restrict all of the variables in H so that one of the restrictions is constant, we get the analogue of (2.32):

$$\mathbf{D}(f) \leq |H| \cdot 2^{-|H|} + \left(1 - 2^{-|H|}\right) \widetilde{\mathbf{D}}_{d-1}. \tag{2.33}$$

However, if we only restrict those variables i in H with $\deg_i(f) = d$, we obtain

$$\mathbf{D}(f) \leq |H| \cdot 2^{-d} + \widetilde{\mathbf{D}}_{d-1}. \tag{2.34}$$

Combining these two inequalities yields

$$\widetilde{\mathbf{D}}_d \leq \max_{1 \leq k \leq d} \left\{ \min\left(k \cdot 2^{-k} + (1 - 2^{-k})\widetilde{\mathbf{D}}_{d-1},\ k \cdot 2^{-d} + \widetilde{\mathbf{D}}_{d-1} \right) \right\}. \tag{2.35}$$

Note that $\widetilde{\mathbf{D}}_1 = \widetilde{\mathbf{D}}_2 = \frac{1}{2}$, since the only monotone functions of degree exactly two are AND_2 and OR_2. Starting with these values and using (2.35) to recursively compute bounds on $\widetilde{\mathbf{D}}_d$, we find that $\widetilde{\mathbf{D}}_{30} \leq 1.3243$, and hence $\mathbf{D}(f) \leq 1.3243 + \sum_{d=31}^{\infty} \frac{d}{2^d} < 1.325$. □

Remark: In [13], a function of degree d with $1.5 \cdot 2^d - 2$ relevant variables is constructed. Therefore, Theorem 2.3.15 implies that all monotone functions of a given degree have at least 11% fewer variables than do certain general functions of the same degree.

2.3.5 Sensitivity

Define $\mathbf{S}(f) := \sum_{i\in[n]} \frac{\delta_i(f)}{2^{\mathrm{sens}_i(f)}}$ and $\mathbf{S}(H, f) = \sum_{i\in H} \frac{\delta_i(f)}{2^{\mathrm{sens}_i(f)}}$ for any $H \subseteq [n]$. In light of the previous subsections, it seems natural to expect that one should be able to prove a bound $\mathbf{S}(f) = O(1)$ for any f using a similar inductive argument, thereby improving Simon's theorem (in the same sense that [13] improved Nisan-Szegedy's bound.) However, choosing a good H to restrict for $\mathbf{S}$ is tricky business – neither choice from the previous two subsections will work in general here. Despite this challenge, we believe such a bound does hold, so we leave it as a conjecture and provide some evidence in favor of it below.

Conjecture 2.3.16. *For any f, $n(f) \lesssim 4^{s(f)}$. More strongly, $\mathbf{S}(f) \lesssim 1$.*

Our first piece of supporting evidence for Conjecture 2.3.16 comes from a direct combination of (2.2.4) with Theorem 2.3.13:

Theorem 2.3.17. *For any monotone f, $n(f) \leq \frac{1}{2} \cdot 4^{s(f)}$.*

Theorem 2.3.17 is especially interesting in light of the fact that the tightest known example in Simon's theorem is monotone. Our next two pieces of evidence are corollaries of the following lemma, which is essentially a consequence of Huang's theorem.

Lemma 2.3.18. *For any function f, and any monomial M appearing in f, the number of variables $i \in M$ with $\mathrm{sens}_i(f) \leq k$ is at most $(k-1)^2$. The same is true of any M with $\hat{f}(M) \neq 0$.*

Proof. Let $B = \{i \in M : \mathrm{sens}_i(f) \leq k\}$, and let $M' \subseteq M$ be a *minimal* monomial containing B, in the sense that no other monomial N has $B \subseteq N \subset M'$. Let α be a partial assignment which sets all coordinates in $[n] \setminus M'$ to arbitrary values in $\{0, 1\}$, and sets those in $M' \setminus B$ to 1. Consider the polynomial f_α, which depends only on the variables in B. If f_α does not have full degree $|B|$, this could only be because the term $c_{M'} \prod_{i\in B} x_i \prod_{i\in M'\setminus B} x_i$ cancelled with another term of the form $c_N \prod_{i\in B} x_i \prod_{i\in N\setminus B} x_i$ when $M' \setminus B$ was restricted to 1. But this could only happen if $B \subseteq N \subset M'$, which by minimality of M' cannot happen. Therefore $|B| = \deg(f_\alpha)$, and by Huang's theorem [40], $\deg(f_\alpha) \le s(f_\alpha)^2$. But since $\mathrm{sens}_i(f) \le k$ for each $i \in B$, $s(f_\alpha) \le \max_{i\in B} \mathrm{sens}_i(f) - 1 \le k-1$, and hence $|B| \le (k-1)^2$.

To prove the same for a monomial M appearing in the fourier transform, we switch to ± 1 notation and observe that if $z \sim \{\pm 1\}^{[n]\setminus M}$ is a random assignment to the variables outside of M, then

$$\begin{aligned}\mathbb{E}_{z\sim\{\pm1\}^{[n]\setminus M}}[\widehat{f_z}(M)^2] &= \mathbb{E}_{z\sim\{\pm1\}^{[n]\setminus M}}\left[\left(\sum_{T\supseteq M}\hat{f}(T)\chi_{T\setminus M}(z)\right)^2\right] \\ &= \sum_{T\supseteq M}\hat{f}(T)^2 \ge \hat{f}(M)^2 > 0\end{aligned}$$

and hence there exists some restriction $f_z : \{\pm1\}^M \to \{\pm1\}$ with $\widehat{f_z}(M) \neq 0$. Therefore we can apply Huang's theorem as above and reach the same conclusion. □

As we show below, Lemma 2.3.18 implies a bound on the number of variables $i \in R(f)$ with $\mathrm{sens}_i(f) \le k$. Unlike the bound $\frac{1}{4}\mathbf{I}[f]\cdot 2^k$ (from Lemma 2.3.6), this bound only depends on k, and not the (average) sensitivity of the function f, and for $k \ll \sqrt{\mathbf{I}[f]}$ it says something much stronger.

Corollary 2.3.19. *Let $v : \mathbb{N} \to \mathbb{N}$ be any increasing function such that $\sum_{k=1}^{\infty} \frac{k}{v(k)} = C_v < \infty$. Then any boolean function f has at most $C_v \cdot v(k) \cdot 2^k$ relevant variables with $\mathrm{sens}_i(f) \le k$. In particular, the number of such variables is $O_\epsilon(k^{2+\epsilon}2^k)$ for any $\epsilon > 0$.*

Proof. For simplicity we consider only $v(k) = k^3$, the same proof works in general. Let S be any set with $\hat{f}(S) \neq 0$, and let $a_k = |\{i \in S : \mathrm{sens}_i(f) = k\}|$. Then by Lemma 2.3.18, for each ℓ we have $\sum_{k=2}^{\ell} a_k \le (\ell-1)^2$. Note that the solution to the (integer) linear program

$$\text{maximize} \sum_{k=2}^{\infty} \frac{a_k}{k^3}$$

$$\text{subject to} \begin{cases} \sum_{k=2}^{\infty} a_k = d \\ \sum_{k=2}^{\ell} a_k \le (\ell-1)^2 \quad \text{for all } \ell \ge 1 \end{cases} \tag{2.36}$$

occurs when as much weight is put on the lower values of k as possible, which means setting $a_k = 2k - 3$. Therefore,

$$\sum_{i\in S} \frac{1}{\mathrm{sens}_i(f)^3} = \sum_{k\geq 2} \frac{a_k}{k^3} \leq \sum_{k=2}^{\infty} \frac{2k-3}{k^3} =: c < \infty.$$

By Parseval's identity $\sum_{S\subseteq[n]} \hat{f}(S)^2 = 1$, this implies

$$c \geq \sum_{S\subseteq[n]} \sum_{i\in S} \frac{1}{\mathrm{sens}_i(f)^3} \hat{f}(S)^2 = \sum_{i\in R(f)} \frac{\mathrm{Inf}_i[f]}{\mathrm{sens}_i(f)^3} \tag{2.37}$$

On the other hand, by Lemma 2.3.6, $\mathrm{Inf}_i[f] \geq 2^{-\mathrm{sens}_i(f)}$. Then

$$c \geq \sum_{i\in R(f)} \frac{1}{2^{\mathrm{sens}_i(f)}\mathrm{sens}_i(f)^3} \geq \frac{1}{k^3 2^k} \cdot |\{i \in S \,:\, \mathrm{sens}_i(f) \leq k\}|, \tag{2.38}$$

from which the corollary follows. □

A similar argument also works to show that $\mathbf{S}(M, f) = O(1)$ for any monomial occurring in f.

Corollary 2.3.20. *For any function f, and any monomial M of degree d in f,*

$$\mathbf{S}(M, f) \leq \sum_{k=2}^{\lfloor\sqrt{d}+1\rfloor} \frac{2k-3}{2^k} + \frac{d - \lfloor\sqrt{d}\rfloor^2}{2^{\lfloor\sqrt{d}+2\rfloor}} < 1.5.$$

Finally, we remark that most of the known functions[4] with low sensitivity compared to the number of relevant variables have the property that almost all of their variables never get to "interact" – that is, they are never simultaneously sensitive. Below, we give a simple tensorization argument which implies that any function f with this property must obey the bound on $n(f)$ in Conjecture 2.3.16.

Lemma 2.3.21. *Suppose we can write $R(f) = Y \sqcup Z$, where for every input x, the set $s(f, x)$ of sensitive coordinates for f at x has $|s(f, x) \cap Y| \leq 1$. Then $|Y| < 4^{s(f)}$.*

[4] For example, the address function, the monotone address function, and the low-depth large-junta construction of Kane [45] all have this property.

Proof. Replacing each $y \in Y$ with a copy of f on $n(f)$ fresh variables x^y, we obtain a function f_2 with $s(f_2) \leq s(f)+s(f)-1$, since at most one of the y variables and $s(f)-1$ other variables are sensitive in f, which is really at most $s(f)$ "new" variables and $s(f)-1$ "old" variables in f_2. Also, it is clear that $n(f_2) = |Z|+|Y|(|Y|+|Z|) \geq |Y|^2$. Recursively, we let f_k be the function obtained from f by replacing each $y \in Y$ with a copy of f_{k-1} on fresh variables. By the same reasoning as the $k = 2$ case, we see that $s(f_k) \leq s(f) + s(f_{k-1}) - 1 \leq ks(f) - k$ and $s(f_k) \geq |Y|^k$. If $|Y| \geq 4^{s(f)}$, then $n(f_k) \geq 4^{ks(f)} = 4^k \cdot 4^{ks(f)-k} \geq 4^k \cdot 4^{s(f_k)}$, which contradicts Simon's theorem for k large enough so that $4^k > ks(f) - k \geq s(f_k)$. Therefore, $|Y| < 4^{s(f)}$ as claimed. □

2.3.6 Mixing measures

The goal of this subsection is to prove bounds on the number of relevant variables in terms of multiple complexity measures *simultaneously*, e.g.

$$n(f) \lesssim 2^{\frac{\deg(f)}{2}+s(f)}. \tag{2.39}$$

Note that such a bound would follow from taking the geometric mean of Theorem 2.3.12 with Conjecture 2.3.16, however, we can give a direct and unconditional proof using the methodology we have already developed, combined with the following simple observation:

Observation 2.3.2. *$\mathcal{R}_i$ is convex.*

We therefore define, for any $\beta \in [0, 1]$, the coordinate measures $ds_i^\beta, cs_i^\beta \in \mathcal{R}_i$ via

$$ds_i^\beta(f) \;:=\; \beta \cdot \deg_i(f) + (1-\beta) \cdot \mathrm{sens}_i(f) \tag{2.40}$$

$$cs_i^\beta(f) \;:=\; \beta \cdot \mathrm{cert}_i(f) + (1-\beta) \cdot \mathrm{sens}_i(f) \tag{2.41}$$

and the associated potentials

$$\mathbf{DS}^{\beta}(f) := \sum_{i\in[n]} \frac{\delta_i(f)}{2^{ds_i^{\beta}(f)}} \tag{2.42}$$

$$\mathbf{CS}^{\beta}(f) := \sum_{i\in[n]} \frac{\delta_i(f)}{2^{cs_i^{\beta}(f)}}. \tag{2.43}$$

Proposition 2.3.22. *For each $\beta \in (0,1]$, $\mathbf{DS}^{\beta}(f) = O_{\beta}(1)$. In particular, for $\beta = 1/2$, $\mathbf{DS}^{\beta}(f) < 8.277$ for all f.*

Proof. Since $\beta > 0$, we can essentially let $\deg_i$ do the legwork while sens_i simply hangs on for a free ride. Indeed, let

$$\mathbf{DS}_d^{\beta} := \max_{\{f\,:\,\deg(f)=d\}} \mathbf{DS}^{\beta}(f)$$

and suppose f is any function of degree d. Let C be any minimal certificate for f, and let H be the variables i which are fixed by C. Let $H' \subseteq H$ be those $i \in H$ with $\deg_i(f) = d$. Since any restriction α of H' lowers the degree of f_{α}, inequality (2.13) implies

$$\mathbf{DS}^{\beta}(f) \leq \frac{d^3}{2^{2-\beta}(2^{\beta})^d} + \mathbf{DS}_{d-1}^{\beta}, \tag{2.44}$$

where we have used that $ds_i^{\beta}(f) \leq 2^{-\beta d - 2(1-\beta)}$ for each of the variables $i \in H'$, and that $|H'| \leq C(f) \leq \mathrm{DT}(f) \leq \deg(f)^3$ by a result of Midrijanis [55]. By induction, we then have for all d that

$$\mathbf{DS}_d^{\beta} \leq \sum_{i=1}^{d} \frac{i^3}{2^{2-\beta}(2^{\beta})^i} < \infty \tag{2.45}$$

which implies the desired conclusion with constant $\frac{1}{2^{2-\beta}} \sum_{i=1}^{\infty} \frac{i^3}{(2^{\beta})^i}$, but this can be improved dramatically. By Lemma 2.3.6, $\mathbf{DS}^{\beta}(f) \leq \frac{1}{2^{2-\beta}}\mathbf{I}[f]$, so $\mathbf{DS}^{\beta}(f) \leq \min_{k\geq 1}\{\frac{k}{2^{2-\beta}} + \sum_{i\geq k+1} \frac{i^3}{2^{2-\beta}(2^{\beta})^i}\}$. For $\beta = \frac{1}{2}$, this minimum occurs at $k = 32$, giving

$$\mathbf{DS}^{\frac{1}{2}}(f) \leq \frac{1}{2^{1.5}}\left(32 + \sum_{i=33}^{\infty} \frac{i^3}{2^{i/2}}\right) = 11.602.$$

We can also keep track of block sensitivity and turn the crank of Lemma 2.3.8, which ultimately yields a bound $\mathbf{DS}^{\frac{1}{2}}(f) \leq 8.277$. $\square$

Corollary 2.3.23. *For any f, $n(f) \leq 8.277 \cdot 2^{\frac{\deg(f)}{2}+s(f)}$.*

Corollary 2.3.23 implies in particular, that for every $\epsilon > 0$, either $n(f) \leq \frac{1}{\varepsilon} \cdot 4^{s(f)}$ or $n(f) < 70 \cdot \varepsilon \cdot 2^{\deg(f)}$. In other words, any function f which fails to satisfy Conjecture 2.3.16 has $n(f) = o(2^{\deg(f)})$. We can prove a similar result for $\mathbf{CS}^{\beta}(f)$, although the bound is slightly worse at $\beta = 1/2$ (by a logarithmic factor). The idea is similar to the proof of Theorem 2.3.13, although the recursive bound that arises is a bit more complicated. To deal with this, we make use of a small technical lemma.

Lemma 2.3.24. *Suppose a non-negative sequence $\{A_d\}_{d\in\mathbb{N}}$ satisfies*

$$A_{d+1} \leq \max_{h\in\mathbb{N}} \left\{ B \cdot h \cdot \alpha^h + \left(1 - \frac{1}{2^h}\right) A_d \right\} \tag{2.46}$$

for some constants B and $\alpha < 1$, and every $d \geq 1$. Then $A_d = O_{B,\alpha}(\log d)$. If $\alpha < 1/2$, then $A_d = O_{B,\alpha}(1)$.

Proof. First we treat the easy case $\alpha < 1/2$. In this case, we can write $B \cdot h \cdot \alpha^h = (B \cdot h \cdot \gamma^h) \cdot \frac{1}{2^h}$, for some $\gamma < 1$. Since $m := \max_{h\in\mathbb{N}} B \cdot h \cdot \gamma^h < \infty$, we can set $C = \max\{A_1, m\}$. If $A_d \leq C$, then for every h, $\frac{B\cdot h\cdot\gamma^h}{2^h} + (1-\frac{1}{2^h})A_d \leq C\cdot(\frac{1}{2^h}+(1-\frac{1}{2^h})) = C$, and hence $A_{d+1} \leq C$, and the conclusion follows by induction.

Now suppose $\alpha = \frac{1}{2}$. We can loosen the upper bound by allowing h to take on any positive real value. By reparameterizing h as $h \ln 2$ and modifying the constant B, the upper bound is maximized when

$$\frac{d}{dx}\left[e^{-x}(Bx - A_d)\right] = 0 \implies x = 1 + A_d/B \tag{2.47}$$

and hence $A_{d+1} \leq (B + A_d)e^{-(1+A_d/B)} + A_d(1 - e^{-(1+A_d/B)}) = A_d + Be^{-(1+A_d/B)}$. Note that the function $a \mapsto a + Be^{-(1+a/B)}$, is increasing for $a \geq 0$, so any bound of the form $A_d \leq A^*$, implies the bound $A_{d+1} \leq A^* + Be^{-(1+A^*/B)}$.

We'll prove by induction on d that $A_d \leq C\sum_{i=1}^{d}\frac{1}{i}$, where $C := \max\{A_1, B\}$. The base case is clear. Suppose the bound holds for some $d \geq 1$. Since $A_d \leq C\sum_{i=1}^{d}\frac{1}{i}$, the above reasoning implies

$$\begin{aligned} A_{d+1} &\leq C\sum_{i=1}^{d}\frac{1}{i} + (B/e)e^{-(C/B)\sum_{i=1}^{d}\frac{1}{i}} \\ &\leq C\sum_{i=1}^{d}\frac{1}{i} + (B/e)e^{-\ln(d+1)} \\ &\leq C\sum_{i=1}^{d}\frac{1}{i} + \frac{B/e}{d+1} \\ &\leq C\sum_{I=1}^{d+1}\frac{1}{i}, \end{aligned}$$

where we have used the Riemann sum inequality $\sum_{i=1}^{d}\frac{1}{i} \geq \ln(d+1)$. □

Proposition 2.3.25. *For each* $\beta \in (\frac{1}{2}, 1]$, $\mathbf{CS}^{\beta}(f) = O_{\beta}(1)$. *For* $\beta = \frac{1}{2}$, $\mathbf{CS}^{\frac{1}{2}}(f) \leq \log s(f) + \frac{\gamma}{2}$, *where* $\gamma \approx 0.5772$ *is the Euler-Mascheroni constant.*

Proof. Let

$$\mathbf{CS}_d^{\beta} := \max_{\{f\,:\,\deg(f)=d\}} \mathbf{CS}^{\beta}(f)$$

and let f be any function with degree d. Let C be a globally minimal certificate for f, and let H be the variables i which are fixed by C. We note that it follows from global minimality of the certificate C that $\mathrm{cert}_i(f) \geq 2|H|$ for all $i \in H$. Then apply (2.13) to obtain the analogue of (2.32):

$$\mathbf{CS}^{\beta}(f) \leq |H|4^{-\beta\cdot|H|-(1-\beta)} + (1-2^{-|H|})\mathbf{CS}_{d-1}^{\beta} \tag{2.48}$$

Hence the sequence $\{\mathbf{CS}_d^{\beta}\}_{d\in\mathbb{N}}$ satisfies the conditions of Lemma 2.3.24, with $B = \frac{1}{2}$ and $\alpha = 4^{-\beta}$. Hence, for $\beta > \frac{1}{2}$, $\alpha < \frac{1}{2}$ and $\mathbf{CS}^{\beta}(f) = O_{\beta}(1)$. For $\beta = \frac{1}{2}$, we are in the $\alpha = \frac{1}{2}$ case of the lemma, and since $\max\{B, \mathbf{CS}_1^{\frac{1}{2}}\} = \frac{1}{2}$, we conclude

$$\begin{aligned} \mathbf{CS}^{\frac{1}{2}}(f) &\leq \frac{1}{2}\sum_{i=1}^{d}\frac{1}{i} \\ &\leq \frac{1}{2}\log\deg(f) + \gamma/2 \\ &\leq \log s(f) + \gamma/2, \end{aligned}$$

where the final inequality is Huang's theorem, $s(f) \geq \sqrt{\deg(f)}$. □

Corollary 2.3.26. *For any $\beta \in (\frac{1}{2}, 1]$ and any f, $n(f) \lesssim_\beta 4^{\beta \cdot C(f) + (1-\beta) \cdot s(f)}$, and $n(f) \leq (\log s(f) + \frac{\gamma}{2}) \cdot 4^{\frac{C(f)+s(f)}{2}}$.*

2.3.7 Decision tree depth

For decision tree depth – unlike the other complexity measures considered thus far – getting a tight bound on $n(f)$ is trivial. Indeed, a depth d binary tree has at most $2^d - 1$ nodes, so $n(f) \leq 2^{\mathrm{DT}(f)} - 1$, and this is obtained by the function which queries a different variable at each node. However, the question becomes nontrivial when restricted to *monotone* boolean functions. Let us denote the set of monotone boolean functions of depth d by $\mathcal{M}_d$ and define the quantities

$$\mathbf{R}_d^{\mathrm{DT}} := \max_{f \in \mathcal{M}_d} n(f) \quad \text{and} \quad \mathbf{R}^{\mathrm{DT}} := \limsup_{d \to \infty} \frac{\mathbf{R}_d^{\mathrm{DT}}}{2^d}.$$

It is quite possible (and, we believe, probably true) that $\mathbf{R}^{\mathrm{DT}} = 0$ – see Section 2.5 for comments. In this section, we give a proof that

$$\mathbf{R}^{\mathrm{DT}} \leq \frac{1}{4}. \tag{2.49}$$

Here we do not use the general restriction-reduction strategy of the previous sections. Instead, our main idea is in the following lemma, which essentially says that unless both of the subfunctions f_0, f_1 of a node in a monotone decision tree have very short certificates, they must share a significant number of relevant variables.[5]

Lemma 2.3.27. *Let f_0, f_1 be the two subfunctions from the root node in a monotone decision tree. If neither f_0 nor f_1 is constant, then*

$$C^0_{\min}(f_0) + C^1_{\min}(f_1) \leq |R(f_0) \cap R(f_1)| + 1.$$

Proof. We first claim that every assignment to $R(f_0) \cap R(f_1)$ must either force $f_0 = 0$

[5] This property, of course, does not hold in general for non-monotone decision trees!

or force $f_1 = 1$. To see this, let $C := R(f_0) \cap R(f_1)$. Let us decompose any assignment α to $R(f)$ into $(\alpha_x, \alpha_0, \alpha_1, \alpha_C)$, where each component is the assignments to x (the root node), $R(f_0) \setminus R(f_1)$, $R(f_1) \setminus R(f_0)$, and C respectively. Suppose for the sake of contradiction that there is some assignment β_C to C which does not force $f_0 = 0$ or $f_1 = 1$ – then we can pick assignments α, α' such that: (i) $f_0(\alpha_0, \beta_C) = 1$ and (ii) $f_1(\alpha'_1, \beta_C) = 0$. But then $f(0, \alpha_0, \alpha'_1, \beta_C) = 1$ and $f(1, \alpha_0, \alpha'_1, \beta_C) = 0$, which violates monotonicity of f since $(0, \alpha_0, \alpha'_1, \beta_C) \prec (1, \alpha_0, \alpha'_1, \beta_C)$, which proves our claim.

Now fix some ordering $x_1, \ldots, x_{|C|}$ of C and consider the $|C| + 1$ assignments

$$\alpha_i := (\underbrace{1, \ldots, 1}_{i}, \underbrace{0 \ldots, 0}_{|C|-i}), \text{ for } i = 0, 1, \ldots, |C|.$$

By the claim above, each α_i forces either $f_0 = 0$ or $f_1 = 1$. In particular, we know that α_0 forces $f_0 = 0$ and $\alpha_{|C|}$ forces $f_1 = 1$. (Indeed, if α_0 does not force $f_0 = 0$, then $f_1 \equiv 1$, which we assumed is not the case, and likewise for $\alpha_{|C|}$.) Therefore, since $\alpha_i \prec \alpha_{i+1}$, there must be some $0 \leq i \leq |C| - 1$ for which α_i forces $f_0 = 0$ and α_{i+1} forces $f_1 = 1$. Hence, by monotonicty, there is a 1-certificate for f_1 fixing only the variables $\{x_1, \ldots, x_{i+1}\}$ to 1, and a 0-certificate for f_0 fixing only the variables $\{x_{i+1}, \ldots, x_{|C|}\}$ to 0. This implies $C^0_{\min}(f_0) + C^1_{\min}(f_1) \leq i + 1 + |C| - i = |C| + 1$. □

We also need the following standard fact, whose easy proof we omit:

Fact 2.3.28. *Let g be any function which does not depend on the variable a. Then* $\mathrm{DT}(a \vee g) = \mathrm{DT}(a \wedge g) = 1 + \mathrm{DT}(g)$.

Lemma 2.3.29. *For $d \geq 2$, $\mathbf{R}^{\mathrm{DT}}_d \leq \max\left\{2\mathbf{R}^{\mathrm{DT}}_{d-1} - 2,\ 2 + 2\mathbf{R}^{\mathrm{DT}}_{d-2},\ 1 + \mathbf{R}^{\mathrm{DT}}_{d-1}\right\}$.*

Proof. Let $f \in \mathcal{M}_d$, for $d \geq 2$. We consider the possible values of $c_0 := C^0_{\min}(f_0)$ and $c_1 := C^1_{\min}(f_1)$. If either $c_0 = 0$ or $c_1 = 0$ (i.e. one of the subfunctions is constant), then $n(f) \leq 1 + \mathbf{R}^{\mathrm{DT}}_{d-1}$.

Otherwise, $c_0, c_1 \geq 1$ and Lemma 2.3.27 applies. If $\min\{c_0, c_1\} \geq 2$, then $c_0 + c_1 \geq$

4, and so by the lemma,

$$\begin{aligned} n(f) &\leq 1 + n(f_0) + n(f_1) - |R(f_0) \cap R(f_1)| \\ &\leq n(f_0) + n(f_1) - 2 \qquad (2.50) \\ &\leq 2\mathbf{R}_{d-1}^{\mathrm{DT}} - 2. \end{aligned}$$

If $c_0 = c_1 = 1$, then we can write $f_0 = a \wedge g$ and $f_1 = a \vee h$ for some functions g and h which do not depend on a. By Fact 2.3.28, $\mathrm{DT}(g) = \mathrm{DT}(f_0) - 1 \leq d - 2$, and similarly $\mathrm{DT}(h) \leq d - 2$, so $n(f) \leq 1 + 1 + n(g) + n(h) \leq 2 + 2\mathbf{R}_{d-2}^{\mathrm{DT}}$.

Finally, it remains to consider the case when $\{c_0, c_1\} = \{1, 2\}$. Without loss of generality, suppose $c_0 = 1$. It follows that we can write $f_0 = a \wedge g$, for some function g which does not depend on a. By Fact 2.3.28, $\mathrm{DT}(g) = \mathrm{DT}(f_0) - 1 \leq d - 2$, and hence

$$\begin{aligned} n(f) &\leq 1 + n(f_0) + n(f_1) - |R(f_0) \cap R(f_1)| \\ &\leq 1 + \mathbf{R}_{d-1}^{\mathrm{DT}} + 1 + \mathbf{R}_{d-2}^{\mathrm{DT}} - 3 \qquad (2.51) \\ &= \mathbf{R}_{d-1}^{\mathrm{DT}} + \mathbf{R}_{d-2}^{\mathrm{DT}} - 1 \\ &\leq 2\mathbf{R}_{d-1}^{\mathrm{DT}} - 2. \end{aligned}$$

□

Proof of (2.49): Since $\mathbf{R}_1^{\mathrm{DT}} = 1$, Lemma 2.3.29 immediately implies $\mathbf{R}_2^{\mathrm{DT}} \leq 2$, $\mathbf{R}_3^{\mathrm{DT}} \leq 4$, $\mathbf{R}_4^{\mathrm{DT}} \leq 6$, and $\mathbf{R}_5^{\mathrm{DT}} \leq 10$. It is also easy to construct explicit examples showing that these all of these inequalities are actually equalities – in fact, if $g(x) \in \mathcal{M}_{d-2}$, then the function

$$f(a, b, x, y) = ((\neg a) \wedge (b \wedge g(x))) \vee (a \wedge (b \vee g(y))) \in \mathcal{M}_d$$

has $2n(g) + 2$ relevant variables. Therefore $\mathbf{R}_d^{\mathrm{DT}} \geq 2\mathbf{R}_{d-2}^{\mathrm{DT}} + 2$, and the bound $\mathbf{R}_d^{\mathrm{DT}} \leq 2\mathbf{R}_{d-1}^{\mathrm{DT}} - 2$ becomes the dominant bound in the lemma for $d \geq 4$. We can rewrite this inequality as

$$(\mathbf{R}_d^{\mathrm{DT}} - 2) \leq 2(\mathbf{R}_{d-1}^{\mathrm{DT}} - 2) \quad \text{for } d \geq 5,$$

and therefore $(\mathbf{R}_d^{\mathrm{DT}} - 2) \leq 2^{d-5}(10 - 2) = 2^{d-2} \implies \mathbf{R}_d^{\mathrm{DT}} \leq 2^{d-2} + 2$ and $\mathbf{R}^{\mathrm{DT}} \leq \frac{1}{4}$. □

2.4 A constant factor improvement in the sensitivity conjecture

In their seminal 1994 paper, Nisan and Szegedy [58] proved an upper bound on the block sensitivity of any boolean function f in terms of its degree, namely

$$\mathrm{bs}(f) \leq 2\deg(f)^2. \tag{2.52}$$

In [73], Avishay Tal gives a tensorization argument showing that the constant factor 2 in (2.52) can be reduced to 1:

$$\mathrm{bs}(f) \leq \deg(f)^2. \tag{2.53}$$

In this section, we improve upon the original argument of Nisan and Szegedy to further improve the constant in (2.53):

Theorem 2.4.1. *For any boolean function f,*

$$\mathrm{bs}(f)^2 - \mathrm{bs}(f) \leq \frac{2}{3}(\deg(f)^4 - \deg(f)^2) \tag{2.54}$$

and hence

$$\mathrm{bs}(f) \leq \sqrt{2/3} \cdot \deg(f)^2 + 1. \tag{2.55}$$

For many pairs of complexity measures, the proof of the best-known relationships between them make use of the inequality (2.53) as an intermediate step. Upgrading those proofs (see [55], [40] and [60]) with Theorem 2.4.1 immediately improves those relations by a constant factor:

Corollary 2.4.2. *For any boolean function f,*

$$\begin{aligned} \mathrm{bs}(f) &\leq \sqrt{2/3} \cdot s(f)^4 + 1 & (2.56)\\ \mathrm{DT}(f) &\leq \sqrt{2/3} \cdot \deg(f)^3 + \deg(f) & (2.57)\\ C(f) &\leq \sqrt{2/3} \cdot s(f)^5 + s(f) & (2.58) \end{aligned}$$

In particular, (2.56) improves on Huang's recent result $\mathrm{bs}(f) \le s(f)^4$, which constitutes the best-known progress on the (strong) sensitivity conjecture, namely $\mathrm{bs}(f) \lesssim s(f)^2$. We note that while the bound in Theorem 2.4.1 can probably be improved further, there is a limit to this approach. The family obtained by tensorizing the function

$$f(x_1, \dots, x_6) := \left(\sum_{i=1}^{6} x_i\right) - \left(\sum_{1 \le i < j \le 6} x_i x_j\right) + x_1x_3x_4 + x_1x_2x_5 + x_1x_4x_5 \\ + x_2x_3x_4 + x_2x_3x_5 + x_1x_2x_6 + x_1x_3x_6 + x_2x_4x_6 + x_3x_5x_6 + x_4x_5x_6$$

certifies that $\mathrm{bs}(f) \ge \deg(f)^{1.63}$ is possible,[6] and since Huang's theorem ($\deg(f) \le s(f)^2$) is tight, combining the two inequalities can never yield a bound stronger than $\mathrm{bs}(f) \le s(f)^{3.26}$. We also remark that, as a consequence of Theorem 2.4.1, any function family generated by tensorizing a single example will always have a truly subquadratic separation between $\mathrm{bs}(f)$ and $\deg(f)$. So if it *is* possible to quadratically separate $\mathrm{bs}(f)$ from $\deg(f)$, this will require a different proof technique.

2.4.1 Proof of Theorem 2.4.1

We begin by recalling Fact 2.3.10, which says that the maximal block sensitivity among functions of degree d is actually obtained by a function f with (i) $f(0) = 0$ and (ii) $f(x) = 1$ for all vectors x of hamming weight 1. Let us say any f satisfying properties (i) and (ii) is in *standard form*. It is easy to see that any function $f(x)$ in standard form has a real multilinear polynomial expansion which looks like

$$f(x_1, \dots, x_b) = x_1 + \cdots + x_b + \sum_{i<j} c_{ij} x_i x_j + \text{ (higher degree terms)} \tag{2.59}$$

where $b = \mathrm{bs}(f) = s(f)$. As it turns out, the coefficients c_{ij} on the quadratic terms $x_i x_j$ in such functions can only take one of two values:

Lemma 2.4.3. *If $f(x_1, \dots, x_b)$ is in standard form, then each quadratic term $x_i x_j$*

[6] This example is due to Kushilevitz [59], and achieves the best-known separation between $\mathrm{bs}(f)$ and $\deg(f)$.

appears with coefficient $c_{ij} \in \{-1, -2\}$ in the polynomial expansion of f.

Proof. For any pair i, j of coordinates, let $e_{i,j}$ be the vector which has ones in the ith and the jth coordinates and zeroes elsewhere. Since f is boolean-valued, $f(e_{i,j}) \in \{0, 1\}$. On the other hand, we can compute $f(e_{i,j})$ by plugging into the polynomial (2.59), which yields $1 + 1 + c_{ij} \in \{0, 1\}$, since all higher degree terms evaluate to 0. □

If we plug any real numbers $(\mu_1, \dots, \mu_b)$ in $[0, 1]^b$ into equation (2.59) for the x_i, we can interpret the result as the expected value of $f(x)$ where the bits x_i of x are independently sampled Bernoulli(μ_i)'s. In particular, taking all $\mu_i = \mu$, we obtain a univariate polynomial $p_f(\mu)$ whose relevant properties are summarized in the lemma below.

Lemma 2.4.4. *If $f(x_1, \dots, x_b)$ is in standard form, then the polynomial $p_f(\mu)$ satisfies*

1. $\deg(p_f) \le \deg(f)$
2. $\sup_{x \in [0,1]} |p_f(x)| \le 1$
3. $|p''_f(0)| \ge b(b-1)$.

Proof. Item (1) follows directly from the definition of p_f, while item (2) follows from the interpretation of $p_f(\mu)$ as the expected value of the boolean function f. To see (3), observe that (2.59) implies that

$$p_f(\mu) = b \cdot \mu + \left(\sum_{i<j} c_{ij} \right) \mu^2 + \text{ (higher degree terms)}, \tag{2.60}$$

and hence by Lemma 2.4.3,

$$p''_f(0) = 2 \cdot \sum_{i<j} c_{ij} \in \left[-4\binom{b}{2}, -2\binom{b}{2} \right],$$

which clearly implies (3). □

In light of Lemma 2.4.4, to bound b in terms of $\deg(f)$, it suffices to bound $|p''_f(0)|$ in terms of $\deg(p_f)$. This is accomplished by the following fact, which is a direct consequence of V. A. Markov's inequality [52].

Fact 2.4.5. *If $p(x)$ is a degree d polynomial satisfying $0 \leq p(x) \leq 1$ for all $x \in [0,1]$, then*

$$|p''(0)| \leq \frac{2d^2(d^2-1)}{3}.$$

Proof. Recall the famous Markov brothers' inequality, which states that if $q(x)$ is a degree d real polynomial, then for each $k \geq 1$,

$$\sup_{x\in[-1,1]} |q^{(k)}(x)| \leq \frac{d^2(d^2-1^2)(d^2-2^2)\cdots(d^2-(k-1)^2)}{1\cdot 3\cdot 5\cdots(2k-1)} \sup_{x\in[-1,1]} |q(x)|.$$

In particular, for $k=2$

$$\sup_{x\in[-1,1]} |q''(x)| \leq \frac{d^2(d^2-1)}{3} \sup_{x\in[-1,1]} |q(x)|. \tag{2.61}$$

To translate (2.61) from $[-1,1]$ to $[0,1]$, we simply let $q(x) := \frac{1}{2} - p\left(\frac{1+x}{2}\right)$. Since $x \mapsto \frac{1+x}{2}$ maps $[-1,1]$ to $[0,1]$, we know that

$$\sup_{x\in[-1,1]} |q(x)| = \sup_{x\in[0,1]} |\frac{1}{2} - p(x)| \leq \frac{1}{2}.$$

Similarly, since $q''(x) = -\frac{1}{4}p''(\frac{1+x}{2})$, we also have

$$\begin{aligned} |p''(0)| \leq \sup_{x\in[0,1]} |p''(x)| &= 4 \sup_{x\in[-1,1]} |q''(x)| \\ &\leq 4\cdot\frac{d^2(d^2-1)}{3}\cdot\sup_{x\in[-1,1]} |q(x)| \\ &\leq \frac{2d^2(d^2-1)}{3}, \end{aligned}$$

which is what we wanted to show. □

Combining (1), (2), and (3) from Lemma 2.4.4 with Fact 2.4.5 yields (2.54). This then implies (2.55), because if b is an integer with $b = (\sqrt{2/3})d^2 + \ell$ for some $\ell \geq 1$, then

$$b^2 - b = (2/3)d^4 - (\sqrt{2/3})d^2 + \ell^2 - \ell > (2/3)d^4 - (2/3)d^2,$$

which contradicts (2.54). Therefore (2.55) holds, and Theorem 2.4.1 is proved. $\square$

2.4.2 Block sensitivity vs. approximate degree:

In [58], the authors also prove a bound on block sensitivity in terms of the *approximate* degree, namely

$$\mathrm{bs}(f) \leq 6 \cdot (\widetilde{\deg}_{1/3}(f))^2. \tag{2.62}$$

Again we can streamline their argument to improve the constant, this time from 6 to 5. We remark that, although $\widetilde{\deg}(f \circ g) = O(\widetilde{\deg}(f) \cdot \widetilde{\deg}(g))$ (by a result of Sherstov [71]), the implicit constant in the $O(\cdot)$ obstructs us from reducing the constant in (2.62) to 1 via tensorization. Another difference between (2.62) and (2.52) is that (2.62) is known to be tight up to the constant – it is shown in [58] that OR_n can be 1/3-approximated by a Chebyshev polynomial of degree $2\sqrt{n}$, and hence the 6 cannot be replaced by anything smaller than $\frac{1}{4}$ in (2.62).

Theorem 2.4.6. *For any boolean function f,*

$$\mathrm{bs}(f) \leq 5 \cdot (\widetilde{\deg}_{1/3}(f))^2$$

Proof. By reasoning as in Fact 2.3.10, we may assume that f is in standard form with (block) sensitivity b. Let $p(x_1, \ldots, x_b)$ be a polynomial of degree $d = \widetilde{\deg}_{1/3}(f)$ satisfying $|p(x) - f(x)| \leq 1/3$ for all $x \in \{0,1\}^b$. Write

$$p(x) = c_0 + c_1x_1 + \cdots + c_bx_b + (\text{higher order terms}),$$

and observe that

$$|p(0) - f(0)| \leq 1/3 \implies |c_0| \leq 1/3 \tag{2.63}$$

$$|p(e_i) - f(e_i)| \leq 1/3 \implies c_i + c_0 \geq 2/3. \tag{2.64}$$

Therefore each $c_i \geq 1/3$, and so $\sum_{i=1}^{b} c_i \geq b/3$. Viewing p as a function on $[0,1]^b$ via its multilinear extension, and considering the univariate function $q(t) := \frac{1}{2} - p(\frac{1+t}{2}, \frac{1+t}{2}, \dots, \frac{1+t}{2})$, we have that

$$\sup_{-1\leq t\leq 1} |q(t)| = \sup_{0\leq t\leq 1} |\frac{1}{2} - p(t,t,\dots,t)| = \sup_{x\in[0,1]^b} |\frac{1}{2} - p(x)| \leq \frac{1}{2} + \frac{1}{3} = 5/6,$$

where the middle inequality is due to convexity/multilinearity. On the other hand, $q'(0) = \frac{1}{2}\sum_{i=1}^{b}(\partial_i p)(0) = \frac{1}{2}\sum_{i=1}^{s} c_i \geq b/6$. By Markov's inequality (in the $k = 1$ case), this implies

$$b/6 \leq \frac{5d^2}{6} \implies b \leq 5d^2.$$

□

2.5 Open problems and future directions

In addition to Conjecture 2.3.16, we suggest some other questions left open by our work:

Asymptotically stronger bounds on $n(f)$ for monotone functions: For monotone functions f, our work shows stronger bounds on $n(f)$ in terms of $\deg(f)$, $s(f)$ and $\mathrm{DT}(f)$ than are known for general functions. However, these bounds still fall short of the best construction, which (for each of the three measures above) is given by the monotone address function MAF_k, for which

$$n(f) = \Theta\left(\frac{2^{\mathrm{DT}(f)}}{\sqrt{\mathrm{DT}(f)}}\right) = \Theta\left(\frac{2^{\deg(f)}}{\sqrt{\deg(f)}}\right) = \Theta\left(\frac{4^{s(f)}}{\sqrt{s(f)}}\right).$$

We conjecture that this is the best possible for monotone functions.

Approximate junta size: If $s(f) = s$, then is f ε-close to a $O_\varepsilon(4^s)$ junta? Verbin, Servedio and Tan conjectured that for *monotone* f with $\mathrm{DT}(f) = d$, f must be ε-close to a $\mathrm{poly}_\varepsilon(d)$ junta, which would imply the same for $s(f)$. However, Kane [45] showed this was false, by constructing a (random) monotone function with $\mathrm{DT}(f) = d$ which is not 0.1-close to any $\exp(\sqrt{d})$-junta. This is tight up to a constant in the exponent by Freidgut's theorem and the OS inequality ($\mathbf{I}[f] \leq \sqrt{\mathrm{DT}(f)}$ for monotone f, see [64]). Since $s(f) \leq \mathrm{DT}(f)$, Kane's construction is also a monotone function with $s(f) = s$ that is not 0.1-close to any $\exp(\sqrt{s})$-junta.

Do large juntas have smaller separations? If $n(f)$ (the number of relevant variables) is exponential in $s(f), \deg(f), C(f), \mathrm{DT}(f)$, then how are these measures related? For example, if $n(f) = 2^{\Omega(\deg(f))}$ then $s(f) = \Omega(\deg(f))$, by Simon's theorem; if $n(f) = 2^{\Omega(s)}$, then $\deg(f) = \Omega(s(f))$ by Nisan-Szegedy. Do the other directions hold? What can be said if $n(f) \geq 2^{s(f)^{1/100}}$?

2.6 Appendix A: A simplified presentation of the Ajtai-Linial construction

One of the earliest and most important motivating examples for studying the analysis of boolean functions is the Kahn-Kalai-Linial theorem [44], which states that for any $f : \{0,1\}^n \to \{0,1\}$, there is a coordinate i with $\mathrm{Inf}_i[f] = \Omega(\mathrm{Var}(f) \cdot \frac{\log n}{n})$. This improves over the trivial $\mathrm{Inf}_i[f] \geq \frac{\mathrm{Var}(f)}{n}$ (Poincare inequality) by a $\log n$ factor. This factor is asymptotically optimal, as witnessed by the balanced Tribes function $F = \bigvee_{j=1}^{n/b} \bigwedge_{k=1}^{b} x_j^k$ (where $b = \log_2 n - \log_2 \log_2 n + o(1)$ is chosen so that $\mathbb{E}[F] = (1-2^{-b})^{n/b} = \frac{1}{2} + o(1)$). One important application of the KKL theorem (and in fact, the first ever) is to the problem of *collective coin flipping*, introduced by Ben-Or and Linial [7]: imagine a two candidate election in which n binary votes $x_1, \ldots, x_n$ are cast, most of which are fair coin flips, but there is some unknown coalition $Q \subset [n]$ of voters whose votes are corrupted by an adversary, who can observe the votes from $[n] \setminus Q$ and cause Q to vote however she pleases. The outcome of the election is then defined to be $f(x)$. Is there a voting function $f(x)$ such that, with high probability over the fair coin flips, the adversary cannot change the outcome of the election?

The answer of course, depends on the size of Q. If $|Q| = o(\sqrt{n})$, for example, a simple majority vote will do the trick. Let us write $I_Q[f]$ to denote the *influence of the coalition Q on f*, that is, the probability that Q can change the outcome of f, i.e.

$$I_Q[f] := \Pr_{\alpha \sim \{0,1\}^{[n]\setminus Q}}[f_\alpha(\cdot) \text{ is not constant}]. \tag{2.65}$$

If $f = \mathrm{MAJ}_n$, then $I_Q[f] = \Pr\left[\left|\frac{n}{2} - \sum_{i \notin Q} x_i\right| \leq |Q|\right]$, which, by the central limit theorem, implies

$$I_Q[f] \lesssim \int_0^{\frac{|Q|}{\sqrt{n}}} e^{-t^2}\, dt \lesssim \frac{|Q|}{\sqrt{n}},$$

so $I_Q[\mathrm{MAJ}_n] \leq \epsilon$ for $|Q| \lesssim \epsilon\sqrt{n}$. In general, if f has the property that $I_Q[f] = o(1)$ for any $Q \subset [n]$ with $|Q| = q(n)$, we say that f is *resilient to coalitions of size $q(n)$*. So MAJ_n is resilient to coalitions of size $o(\sqrt{n})$, yet it is easy to see that *any* coalition of size $\Omega(\sqrt{n})$ will have $\Omega(1)$ influence on f. The constant function $f \equiv 1$

is trivially resilient to coalitions of all sizes, but it is also a pretty terrible way to decide an election – we rule out such functions by restricting to those f which are *nearly balanced*, which we take to mean $\min\{\Pr[f = 1], \Pr[f = 0]\} \geq 1/3$, although the constant $1/3$ is completely arbitrary. The KKL theorem provides[7] a very general upper bound on how resilient a nearly-balanced function can be:

Theorem 2.6.1 (Kahn-Kalai-Linial, 1988)**.** *If $f : \{0,1\}^n \to \{0,1\}$ is nearly balanced, then for each $\epsilon > 0$, there is some coalition Q of size $|Q| = O(\frac{n \log(\frac{1}{\epsilon})}{\log n})$ with $I_Q[f] \geq 1 - \epsilon$.*

Whether Theorem 2.6.1 can be improved remains an interesting open problem. However, a construction of Ajtai and Linial [1] shows that it cannot be improved by very much:

Theorem 2.6.2 (Ajtai-Linial, 1993)**.** *There is a nearly balanced function f which is resilient to coalitions of size $q(n)$, for any $q(n) = o(\frac{n}{\log^2 n})$.*

In what follows, we give a "lecture-friendly" proof of Theorem 2.6.2. We use the original randomized construction of Ajtai and Linial, and simplify the resilience analysis in a few small ways, inspired in part by Meka's recent derandomization of this construction [53]. We emphasize that our proof is merely a *simplification* of the original proof, not a dramatically different one. However, while Theorem 2.6.2 is well-cited and often stated in undergraduate lectures on the analysis of boolean functions, the author could not find a proof of it, or even a proof sketch, in any lecture notes or textbooks available online. Perhaps this is because Ajtai and Linial's original argument, while mostly elementary, is long and somewhat challenging to follow, which hopefully ours is not. As the problem of improving Theorem 2.6.2 is still open, perhaps including the complete proof in lecture would inspire more young minds to work on this interesting open problem.

[7] The coalition Q is essentially formed by taking the maximal influence coordinate, restricting and looping until the function becomes highly biased. (Technically this procedure only works for monotone functions, but it is a standard fact that monotonization can only make a function more resilient.)

2.6.1 The construction

Given n, pick $b = \log_2 n - 2\log_2\log_2 n + o(1)$ such that

$$1 - \frac{\ln 2}{n} \leq 1 - (1 - 2^{-b})^{n/b} \leq 1 - \frac{\ln 2}{n} + \frac{\ln^2 n}{n^2}.$$

For a collection $\mathbf{P} = \{P^1, \ldots, P^n\}$ of partitions $P^i = \{P^i_1, \ldots, P^i_{n/b}\}$ of $[n]$ into parts of size[8] b, and a sequence $\mathbf{g} = (g^1, \ldots, g^n)$ of n strings $g^i \in \{0,1\}^n$, we define the functions

$$F_{P^i,g^i}(x) := \bigvee_{j=1}^{n/b} \bigwedge_{k \in P^i_j} (x_k = g^i_k) \text{ and } F_{\mathbf{P},\mathbf{g}}(x) := \bigwedge_{i=1}^{m} F_{P^i,g^i}(x). \tag{2.66}$$

We'll show that, for some choice of $\mathbf{P}$ and $\mathbf{g}$, the function $F_{\mathbf{P},\mathbf{g}}$ is (a) nearly balanced and (b) resilient to all coalitions of size $o(\frac{n}{\log^2 n})$. First we give a short proof of (a):

Lemma 2.6.1. *For every* $\mathbf{P}$*, there exists some choice of* $\mathbf{g}$ *for which*

$$\Pr_x[F_{\mathbf{P},\,\mathbf{g}}(x) = 1] = \frac{1}{2} + o_n(1).$$

Proof. Fix any $\mathbf{P} = \{P^1, \ldots, P^n\}$. For any g^i, we have $\Pr_x[F_{P^i,g^i}(x) = 1] = 1 - (1 - 2^{-b})^{n/b}$, which by our choice of b, is equal to $1 - \frac{\ln 2}{n} + O(\frac{\ln^2 n}{n^2})$. Observe that, for a fixed x, when the strings g^i are chosen independently and uniformly at random, the events $\{F_{P^i,g^i}(x) = 1\}$ are independent across different values of i. Therefore,

$$\begin{aligned} \mathbb{E}_{\mathbf{g}} \Pr_x[F_{\mathbf{P},\mathbf{g}}(x) = 1] &= \mathbb{E}_x \Pr_{\mathbf{g}}[F_{\mathbf{P},\mathbf{g}}(x) = 1] \\ &= \mathbb{E}_x \left[\prod_{i=1}^{n} \Pr_{g^i}[F_{P^i,g^i}(x) = 1]\right] \\ &= \left(1 - \frac{\ln 2}{n}\right)^n = \frac{1}{2} + O\left(\frac{\log^2 n}{n}\right), \end{aligned}$$

so on average, $F_{\mathbf{P},\mathbf{g}}$ has the correct bias. Fix $\delta = \Theta(\frac{\log^2 n}{n})$. If there is no $\mathbf{g}$ such that $\Pr_x[F_{\mathbf{P},\mathbf{g}}(x) = 1] \in (\frac{1}{2} - \delta, \frac{1}{2} + \delta)$, then by connectedness of the hypercube graph,

[8] We ignore questions of divisibility here, as such issues are inconsequential to the analysis.

there must be some $\mathbf{g}$ such that flipping a single bit g_k^i causes the expectation of $F_{\mathbf{P},\mathbf{g}}$ to change by at least 2δ. This in turn implies that the influence of coordinate i on F_{P^i,g^i} is at least δ, contradicting the fact that $I_i(F_{P^i,g^i}) \le \mathrm{Var}(F_{P^i,g^i}) = O(\frac{1}{n})$. □

2.6.2 Proof of resilience

To prove theorem 2.6.2, it remains to bound the influence of a q-sized coalition Q on $F_{\mathbf{P},\mathbf{g}}$, for $q = \frac{\varepsilon n}{\log^2 n}$. Recall that $I_Q[f]$ is the probability that a random assignment to the variables outside of Q doesn't leave f fixed. So by a union bound,

$$I_Q[F_{\mathbf{P},\mathbf{g}}] \le \sum_{i=1}^{n} I_Q[F_{P^i,g^i}] = \sum_{i=1}^{n} I_Q[F_{P^i,0}], \tag{2.67}$$

where in the final equality we have used the easy observation that $I_Q[F_{P^i,g^i}]$ does not depend on g^i, so we may set it to 0 for convenience. Let us simply write F_{P^i} for $F_{P^i,0}$. A partial assignment $x \in \{0,1\}^{\overline{Q}}$ can only leave F_{P^i} unfixed if (1) for every part P_j^i that does not intersect Q, we have $x_k = 1$ for some $k \in P_j^i$, and (2) for some part P_j^i which intersects Q, we have $x_k = 0$ for all $k \in P_j^i \setminus Q$. These two events are independent, and we can bound their probabilities somewhat crudely. Since there are at least $\frac{n}{b} - q$ parts which do not intersect Q, the probability of (1) is at most $(1-2^{-b})^{n/b-q} \lesssim \frac{1}{n}$. The probability of (2) is, by a union bound, at most

$$\sum_{j:P_j^i\cap Q\neq\emptyset} \frac{1}{2^{b-|P_j^i\cap Q|}} = 2^{-b} \sum_{j:P_j^i\cap Q\neq\emptyset} 2^{|P_j^i\cap Q|}.$$

Therefore,

$$I_Q[F_{P^i}] \le (1-2^{-b})^{n/b-q} 2^{-b} \sum_{j:P_j^i\cap Q\neq\emptyset} 2^{|P_j^i\cap Q|} \lesssim \frac{\log^2 n}{n^2} \sum_{j=1}^{n/b} 1_{Q\cap P_j^i\neq\emptyset} 2^{|P_j^i\cap Q|}. \tag{2.68}$$

If we define $S_i = S_i(\mathbf{P}, Q) := \frac{\log^2 n}{n^2} \sum_{j=1}^{n/b} 1_{Q\cap P_j^i\neq\emptyset} 2^{|Q\cap P_j^i|}$, then (2.67) and (2.68) imply

$$I_Q[F_{\mathbf{P},\mathbf{g}}] \lesssim \sum_{i=1}^{n} S_i. \tag{2.69}$$

Consider choosing the collection $\mathbf{P} = \{P^1, \ldots, P^n\}$ uniformly at random. Then the random variables S_i are i.i.d., with mean

$$\begin{aligned}
\mathbb{E}[S_i] &= \frac{n}{b} \cdot \frac{\log^2 n}{n^2} (\mathbb{E}_{P^1}[2^{|P^1 \cap Q|}] - \Pr_{P^1}[P^1 \cap Q \emptyset]) \\
&= \frac{\log^2 n}{nb} \sum_{k=1}^{b} 2^k \Pr_{P^1}[|P^1 \cap Q| = k]. \qquad (2.70)
\end{aligned}$$

The number of $P^1 \in \binom{[n]}{b}$ with $|P^1 \cap Q| = k$ is exactly $\binom{q}{k}\binom{n-q}{b-k} \le (\frac{eq}{k})^k \binom{n}{b-k}$, and hence

$$\begin{aligned}
\Pr[|P^1 \cap Q| = k] \le \frac{(\frac{eq}{k})^k \binom{n}{b-k}}{\binom{n}{b}} &= \left(\frac{eq}{k}\right)^k \cdot \frac{b(b-1)\cdots(b-k+1)}{(n-b+k)\cdots(n-b+1)} \qquad (2.71) \\
&\le \left(\frac{eq}{k}\right)^k \left(\frac{2b}{n}\right)^k \le \left(\frac{6qb}{n}\right)^k \qquad (2.72)
\end{aligned}$$

which by (2.70) implies

$$\mathbb{E}[S_i] \le \frac{\log^2 n}{nb} \sum_{k=1}^{\infty} \left(\frac{12qb}{n}\right)^k \le \frac{2\log n}{n} \cdot \sum_{k=1}^{\infty} \left(\frac{6\varepsilon}{\log n}\right)^k \le \frac{100\varepsilon}{n}. \qquad (2.73)$$

This already implies that, for a random $\mathbf{P}$ and a random Q of size q, $I_Q[F_{\mathbf{P},\mathbf{g}}] \lesssim \varepsilon$ for any $\mathbf{g}$. However, we need this bound to hold for *all* Q simultaneously, not just an average Q. There is an easy Markov/Chernoff fix, as long as we're willing to pay a $\sqrt{\varepsilon}$ price in the influence bound. Indeed, by Markov's inequality, $\Pr\left[S_i > \frac{100\sqrt{\varepsilon}}{n}\right] \le \sqrt{\varepsilon}$, and hence by a Chernoff bound we have, for each Q,

$$\Pr_{\mathbf{P}}\left[\left|\left\{i \in [n] \,:\, S_i(\mathbf{P}, Q) > \frac{100\sqrt{\varepsilon}}{n}\right\}\right| > 10\sqrt{\varepsilon} n\right] \le 2^{-\varepsilon n}. \qquad (2.74)$$

Since there are only $\binom{n}{q} = \binom{n}{\frac{\varepsilon n}{\log^2 n}} \le \left(\frac{e\log^2 n}{\varepsilon}\right)^{\varepsilon n/\log^2 n} \le 2^{\frac{n \log\log n}{\log^2 n}}$ sets Q to worry about, taking a union bound gives

$$\begin{aligned}
\Pr_{\mathbf{P}}\left[\exists Q \in \binom{[n]}{q} \text{ for which } \left|\left\{i \in [n] \,:\, S_i(\mathbf{P}, Q) > \frac{100\sqrt{\varepsilon}}{n}\right\}\right| > 10\sqrt{\varepsilon} n\right] &\le 2^{-\varepsilon n + \frac{n\log\log n}{\log^2 n}} \\
&< 2^{-\varepsilon n/2}.
\end{aligned}$$

So fix any such $\mathbf{P}$ satisfying $\left|\{i \in [n] : S_i(\mathbf{P}, Q) > \frac{100\sqrt{\varepsilon}}{n}\}\right| \leq 10\sqrt{\varepsilon}n$ for all Q. Let $S(Q) := \{i \in [n] : S_i(\mathbf{P}, Q) > 10\sqrt{\varepsilon}\}$. Then for any Q and any $\mathbf{g}$, we have

$$\begin{aligned}
I_Q[F_{\mathbf{P},\mathbf{g}}] &\leq \sum_{i \in S(Q)} I_Q[F_{P^i}] + \sum_{i \notin S(Q)} I_Q[F_{P^i}] \\
&\lesssim \sum_{i \in S(Q)} S_i + \sum_{i \notin S(Q)} \mathrm{Var}[F_{P^i}] && (2.75) \\
&\lesssim \sum_{i \in S(Q)} \frac{100\sqrt{\varepsilon}}{n} + \sum_{i \notin S(Q)} \frac{1}{n} && (2.76) \\
&\lesssim \sqrt{\varepsilon} + \sqrt{\varepsilon} && (2.77)
\end{aligned}$$

Taking the particular $\mathbf{g}$ from Lemma 2.6.1, we see that the function $F_{\mathbf{P},\mathbf{g}}$ satisfies Theorem 2.6.2. □

Remark: Technically, Ajtai and Linial's original argument proves a slightly stronger result. They showed that, for almost all $\mathbf{P}$ and almost all $\mathbf{g}$, all sets Q of size $\frac{\varepsilon n}{\log^2 n}$ have $I_Q[F_{\mathbf{P},\mathbf{g}}] \lesssim \varepsilon$, whereas our proof shows that, for almost all $\mathbf{P}$, there exists a $\mathbf{g}$ such that all sets Q of size $\frac{\varepsilon n}{\log^2 n}$ have $I_Q[F_{\mathbf{P},\mathbf{g}}] \lesssim \sqrt{\varepsilon}$.

Chapter 3

Applications of Partial Rejection Sampling

3.1 Introduction

Given a set of "bad" events $A_1, \dots, A_m$ depending on variables $X_1, \dots, X_n$, how can we sample a *uniformly random* assignment σ to the variables subject to the constraint that

$$\sigma \in \bigcap_{i=1}^{m} \overline{A_i} \ ?$$

The most naive approach would be to repeatedly sample an assignment σ uniformly at random until, by a stroke of luck[1], $\sigma \in \cap_{i=1}^{m} \overline{A_i}$. While this algorithm is attractively simple and always outputs a sample from the desired distribution upon termination, it takes an often intractably large $\frac{1}{\Pr[\cap_{i=1}^{m} \overline{A_i}]}$ rounds in expectation. Is there a general-purpose way to make rejection sampling more efficient?

In a 2017 paper [32], Guo, Jerrum and Liu gave a general algorithm for uniform sampling which captures (at least partially) the simplicity of rejection sampling, without necessarily taking an enormous number of rounds to terminate when $\Pr[\cap_{i=1}^{m} \overline{A_i}]$ is small. The underlying idea of their method, called *partial rejection sampling*, is

[1] One must of course assume *something* about the events A_i to ensure that $\Pr[\cap_{i=1}^{m} \overline{A_i}] > 0$. A common choice is given by the hypothesis of the Lovasz Local Lemma, which roughly says that this probability is positive when $\Pr[A_i]$ is small compared to the number of events on which A_i depends.

simple: instead of resampling all of the variables every time at least one bad event occurs, sample just a subset of them, which includes all of the variables involved in violated constraints – which we call $\mathsf{Badvar}(\sigma)$ – and not too many more than that. Indeed, consider drawing a random assignment σ_0 from the product distribution on the X_j, and continually resampling the variables in $\mathsf{Badvar}(\sigma_i)$ to obtain σ_{i+1}. This is the algorithm analyzed in Moser and Tardos's celebrated constructive proof of the Lovasz Local Lemma [57], in which the authors give a bound on the expected number t of rounds of resampling until $\mathsf{Badvar}(\sigma_t) = \emptyset$, which is finite when the events A_i satisfy the conditions of the Lovasz Local Lemma (LLL). The most general condition on the collection of events A_i which guarantees that $\Pr[\cap_{i=1}^{m}\overline{A_i}] > 0$ is called *Shearer's condition* [70], and in 2011, Kolipaka and Szegedy [49] strengthened the arguments of Moser and Tardos, showing that Shearer's condition is sufficient for the same algorithm to terminate efficiently. The instances for which the lower bound on $\Pr[\cap_{i=1}^{m}\overline{A_i}]$ from Shearer's condition is tight are exactly those in which *dependent bad events are disjoint* – these have been called *extremal* instances [49]. Somewhat amazingly, Guo, Jerrum and Liu showed that, on extremal instances, the random assignment output by the Moser-Tardos algorithm is actually *uniform* among all satisfying assignments to the X_j. A number of classical sampling algorithms actually fit into this framework, such as the "sink-popping" algorithm of Cohn, Pemantle and Propp [15] for generating a uniformly random sink-free orientation of an undirected graph, or Wilson's "cycle-popping" algorithm for sampling uniform spanning trees [78].

For non-extremal instances, however, the output of the Moser-Tardos algorithm may not be uniform. Guo et al. provide one method of growing the set $\mathsf{Badvar}(\sigma)$ to include certain "neighboring" variables (discussed in Section 3.2.2), giving a *generalized partial rejection* algorithm, and in [32] it is shown that this is always a uniform sampler. The authors provide general conditions under which this sampler is also efficient: loosely speaking, this occurs when any dependent pair of bad events *share many of their variables*. As an example, the authors of [32] use PRS to sample satisfying assignments of k-CNFs with the property that no variable occurs in more than $\approx 2^{k/2}$ clauses, and every pair of dependent clauses *share* at least $k/2$ variables. Their methods also yield an algorithm for sampling from the hard-core distribution on independent sets in graphs of maximum degree d up to fugacity $\frac{1}{2\sqrt{e} \cdot d-1}$. This is about a factor of 9 short of the hardness threshold $\frac{e}{d}$, which is attained by the correlation decay-based algorithm of Weitz [76], although as the authors of [32] point out, the PRS-based algorithm "has the advantage of being simple, exact, and running in linear time in expectation."

In this chapter, we consider a few sampling problems which are not extremal in the sense of [49], nor do they have the large-overlap feature mentioned above, and yet we exploit other properties of their underlying dependency graphs to prove that generalized partial rejection sampling solves them efficiently. While these problems are probably not of great theoretical or practical importance, we believe they suggest that partial rejection sampling may be a viable approach for structured sampling problems, even when that structure does not obey the hypotheses of the main theorems in [32]. The two types of problems we consider are:

1. **Sampling w-free strings:** Given input (Σ, w, n), where Σ is a finite alphabet and w is some string over Σ, generate a uniformly random element of Σ^n which does not contain w as a contiguous substring.

2. **Sampling H-free subgraphs of a grid:** Given an input (G, H, λ), where G and H are subgraphs of a grid graph and $\lambda \in (0,1)$, generate a sample from the following distribution, which is supported on H-free (non-induced) subgraphs of G:

$$\Pr(G') \propto \begin{cases} \lambda^{e(G')}(1-\lambda)^{e(G)-e(G')} \propto \left(\frac{\lambda}{1-\lambda}\right)^{e(G')} & \text{if } G' \subseteq G \text{ is } H\text{-free} \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

(We only deal explicitly with the case that G is a subgraph of either the triangular or the square grid, and H is either a triangle or a square, respectively.)

3.1.1 Our results

The main theorem of [32] (Theorem 26 therein), applied directly to Problem 1 above, only guarantees that general PRS is efficient for $|\Sigma| > 16|w|$, which is quite undesirable. By tailoring the runtime analysis to the associated dependency graph, we are able to prove efficiency of PRS under much less severe restrictions:

Theorem 3.1.1. *For $|\Sigma| \geq 3$ and any w, the expected number of rounds used by PRS to sample a w-free string in Σ^n is $O(\log n)$ and the expected number of resampled events is $O(n)$.*

We remark that a variety of efficient samplers exist for this problem. Indeed, a number of efficient combinatorial algorithms are known for counting length n strings in regular languages (see, e.g. [41], [46], [62], [21]). These methods, however, do have a few disadvantages: for one, their complexity grows with $|\Sigma|$ and $|w|$, while PRS actually uses *fewer* rounds of resampling for larger alphabets and longer forbidden words. Secondly, these combinatorial methods can be rather difficult to implement, and many of them involve doing arithmetic with numbers whose bit complexity grows quite rapidly. Even in the parameter regime where some of these methods may outperform PRS, they cannot compete in terms of simplicity of implementation and ease of explanation. Simple MCMC-based approximate samplers may also work here, but it is unclear how to make them exact. We also remark that our methods can easily be adapted to a more general definition of "string", including necklaces, figure-eights, and any path-like object with a bounded number of self-intersections.

In contrast to string sampling, the exact counting problem for subgraph sampling is NP-hard in general, and perhaps remains hard even on instances which are subgraphs of a grid. However, when G and H have the property that no edge of G appears in more than 2 copies of H inside G, then there is an FPRAS for *approximately* sampling from this distribution when $\lambda \leq 1/2$, based on an algorithm of Lin, Liu and Lu [50]. Both the square and triangular grid versions of the problem we consider in this chapter satisfy this property. However, even on these grids the FPRAS can be quite inefficient – the runtime bound in [50] becomes $O(n^{11.16}(1/\epsilon)^{2.58})$ when applied to an $n \times n$ grid.

Again we can try appealing directly to Theorem 26 in [32] to see how PRS performs. When sampling triangle free subgraphs of a triangular grid graph, the theorem guarantees efficiency of PRS for $\lambda \leq \frac{1}{9e} \approx 0.04088$, while for square-free subgraphs of the square grid, it requires $\lambda \leq \frac{1}{12e} \approx 0.03065$. However, when we ran an implementation of PRS on these problems, the efficiency thresholds for λ appeared to be about 0.471 and 0.456 respectively. Inspired by these simulations, we greatly improve the guarantees from [32]:

Theorem 3.1.2. *If $\lambda \leq \lambda_\Delta \approx 0.3748$ and G is a subgraph of the $n \times n$ triangular grid, then running on the instance (G, K_3, λ), PRS takes $O(\log n)$ rounds in expectation. If $\lambda \leq \lambda_\Box \approx 0.4063$ and G is a subgraph of the $n \times n$ square grid, then running on the instance (G, C_4, λ), PRS takes $O(\log n)$ rounds in expectation*[2].

In an appendix to this chapter, we consider an application of PRS to a problem outside of the so-called "variable framework". In [31], Guo and Jerrum give a PRS-based algorithm for sampling from the *hard-spheres* model, which is a particular distribution of non-overlapping spheres of equal radii in $\mathbb{R}^d$, parameterized by an intensity parameter λ. We give a geometric argument which, in any dimension d, slightly increases the range of λ for which their algorithm is guaranteed to be efficient.

3.2 The method of Guo, Jerrum, Liu

In this section, we review the ideas from Guo et al. [32] that we'll need to make use of. We also fix our notation to be largely consistent with theirs. Let $X_1, ..., X_n$ be independent random variables distributed according to some product distribution π^n and let $\{A_1 ..., A_m\}$ be a set of ("bad") events dependent on some of the X_i. Let $\mathsf{var}(A_i)$ be the set of random variables on which the event A_i depends. For an assignment σ to the X_i, we define $\mathsf{Bad}(\sigma)$ to be the set of A_i which hold under σ, and

[2] The overall runtime of the algorithm depends on certain implementation choices, but the most naive implementation requires time $O(n^3 \log n)$, space $O(n^2)$, and uses $O(n^2)$ random bits in expectation.

we let $\mathsf{Badvar}(\sigma)$ be the set of all variables involved in some occurring bad events, that is $\mathsf{Badvar}(\sigma) = \cup_{i \in \mathsf{Bad}(\sigma)} \mathsf{var}(A_i)$. We define the *dependency graph* $G = (V, E)$, where $V = \{A_1, ..., A_n\}$, and $A_i \sim A_j$ (i.e. A_i and A_j are neighbors) if $\mathrm{var}(A_i) \cap \mathrm{var}(A_j) \neq \emptyset$. In other words, $A_i \sim A_j$ if they both depend on some common random variable X_k. For any subset S of bad events, we define $\Gamma(S) = \{A_i \,|\, A_i \sim A_j \text{ for some } A_j \in S, A_i \notin S\}$ and $\Gamma^+(S) = S \cup \Gamma(S)$.

3.2.1 Extremal Partial Rejection Sampling

As mentioned in the introduction, an instance of this setup is called *extremal* if $A_i \sim A_j \implies \Pr[A_i \cap A_j] = 0$. Algorithm 1 below, which is the parallel version of the Moser-Tardos algorithm, generalizes the sink-popping algorithm of Cohn, Pemantle and Propp [15], in which the term "partial rejection sampling" first appeared.

Algorithm 1: Partial Rejection Sampling for extremal instances

```
sample $\sigma = (x_1, \ldots, x_n) \sim \pi^n$
while $\mathsf{Bad}(\sigma) \neq \emptyset$ do
    resample all variables in $\mathsf{Badvar}(\sigma)$
    update $\sigma$
end
output $\sigma$
```

As shown in [32], Algorithm 1 produces a uniform sample from the product distribution on $(X_1, \ldots, X_n)$ conditioned on the event $\overline{\bigcup_{i=1}^m A_i}$. Moreover, by differentiating a relevant generating function, the authors produce an exact formula for the expected number of resampled events during the entire run of this Algorithm on any extremal instance:

Theorem 3.2.1 (Guo, Jerrum, Liu 2017)**.** *Let q_S be the probability that the set of occurring bad events is exactly S, and suppose $q_\emptyset > 0$. Then the expected number of resampled events during Algorithm 1 is $\sum_{i=1}^n \frac{q_{\{i\}}}{q_\emptyset}$.*

3.2.2 General partial rejection sampling

We now describe the general version of partial rejection sampling given in the latter half of [32]. For any set of events S, we let σ_S denote the restriction of the assignment σ to the variables in $\mathsf{var}(S) := \cup_{i \in S} \mathsf{var}(A_i)$. We say[3] that σ_S *blocks* an event A_i if $\sigma_S \implies \overline{A}_i$, that is, A_i cannot occur given that the variables in $\mathsf{var}(S)$ are set according to σ_S. In general, it is not enough to sample only the variables participating in $\mathsf{Bad}(\sigma)$ – in each round we must sample the variables from a larger set of events we call $\mathsf{Res}(\sigma)$. Algorithm 3 gives one method of doing this. In words, it grows $R = \mathsf{Res}(\sigma)$ iteratively, beginning with $R = \mathsf{Bad}(\sigma)$, choosing an unmarked event $i \in U = \Gamma(R)$, and checking whether A_i is blocked by σ_R. If it is, we mark it "don't resample," while if it is not blocked, we add it to R. After we have looped through all of the unmarked events in U, we update $U = \Gamma(R)$, as R might've grown during the last iteration. We continue until we reach a point when $\Gamma(R)$ contains no unmarked events.

Algorithm 2: Computing the resample set $\mathsf{Res}(\sigma)$

```
initialize $R = \mathsf{Bad}(\sigma)$, $N = \emptyset$, and $U = \Gamma(R) \setminus N$
while $U \neq \emptyset$ do
    for $i \in U$ do
        if $\sigma_R$ blocks $i$ then
            update $N = N \cup \{i\}$
        end
        else
            update $R = R \cup \{i\}$
        end
    end
    update $U = \Gamma(R) \setminus N$
end
output $R = \mathsf{Res}(\sigma)$
```

[3] Note that our terminology here differs from that in [32], as we find theirs to be less intuitive.

Algorithm 3: General Partial Rejection Sampling

```
sample $\sigma = (x_1, \ldots, x_n) \sim \pi^n$
while $\mathsf{Bad}(\sigma) \neq \emptyset$ do
    compute $\mathsf{Res}(\sigma)$ via Algorithm 2
    resample the variables in $\cup_{i \in \mathsf{Res}(\sigma)} \mathsf{var}(A_i)$
    update $\sigma$
end
output $\sigma$
```

It is important to note that when looping over the set U in line 3 of Algorithm 2, we must use a fixed, deterministic ordering of U, although this ordering is arbitrary. Let σ_t denote the assignment after t invocations of Algorithm 2 during Algorithm 3, and let Res_t denote the resampled events at this stage. In [32] it is shown that

$$(\sigma_{t+1} \mid \mathsf{Res}_0, \mathsf{Res}_1, \ldots, \mathsf{Res}_t) \sim \pi^n(\cdot \mid \cap_{i \in [m] \setminus \Gamma^+(\mathsf{Res}_t)} \overline{A}_i), \tag{3.2}$$

which implies that $\sigma \sim \pi^n(\cdot \mid \cap_{i \in [m]} \overline{A}_i)$ upon termination, so this is indeed a uniform sampler. As an interesting corollary of 3.2, we see that the sample output by Algorithm 3 is independent of how many rounds it had to run for. This means in particular that a time-constrained user could interrupt and/or restart the algorithm without introducing impatience bias into the sample.

Unlike Algorithm 1, it is likely impossible to give a simple closed form for the expected runtime of Algorithm 3, but Guo, Jerrum and Liu are able to prove a logarithmic upper bound on the number of rounds under a certain set of conditions. Let $p = \max_{i \in [m]} \Pr[A_i]$ and[4] $\Delta = \max_{i \in [m]} |\Gamma(A_i)|$ be the maximum degree in the dependency graph. Let R_{ij} be the event that the partial assignments on $\mathsf{var}(A_i) \cap \mathsf{var}(A_j)$ can be extended to an assignment satisfying A_j. Let $r_{ij} = \Pr[R_{ij}]$ and finally set $r = \max_{i \sim j} r_{ij}$.

Theorem 3.2.2. *([32], Theorem 26): Let m be the number of events and n the number of variables. For any $\Delta \geq 2$, if $6ep\Delta^2 \leq 1$ and $3er\Delta \leq 1$, then the expected number of rounds is $O(\log m)$ and the expected number of resampled events is at most $O(m)$.*

[4] Unless otherwise indicated, all probabilities will be with respect to the product measure π^n.

As mentioned in the introduction, applying this theorem directly to our problems of interest yields quite suboptimal results. So, we now open up the proof of this theorem from [32], laying out the terminology and the general strategy which we will fine-tune in our applications.

The proof of Theorem 3.2.2 works by showing that the expected size of the resampling set Res_t is exponentially decaying in t. The main observation is that when an event i is added during stage ℓ of Algorithm 5, there must be a chain of ℓ events ending at i and beginning at a bad event which *occurs*, such that the partial assignments never block the next event in the chain. The probability of such an occurrence can be bounded by $pr^{\ell-1}$, and the number of such potential paths can be bounded by something like $\Delta(\Delta-1)^{\ell}$. This motivates the condition $r\Delta \lesssim 1$. We justify these observations below.

Lemma 3.2.1. *If i is added to R_ℓ during a run of Algorithm 5 on an assignment σ, then there exists a path $i_0, i_1, \ldots, i_\ell = i$ in the dependency graph such that, for each $0 \le k \le \ell$, $i_k \in R_k$, the events $R_{i_{k-1}i_k}$ occur, and $i_{k'} \sim i_k \iff |k-k'| \le 1$.*

Proof. If i is added in to R_ℓ, it must be because it was unblocked by some neighboring event $i_{\ell-1}$ which was added in the previous round. By induction, there is a path $i_0, i_1, \ldots, i_{\ell-1}$ as in the lemma. If $i \sim i_k$ for $k < \ell - 1$, then Algorithm 5 would've added i to either R_{k+1} or N_{k+1} during stage $k+1 < \ell$, which is impossible since i is added during stage ℓ. Therefore the path $i_0, \ldots, i_{\ell-1}, i_\ell$ satisfies the desired conditions. □

We shall henceforth refer to such paths as *bad paths*, as is done in [32]. More precisely, we define a *path* to be a sequence of events $P = i_0, i_1, \ldots, i_\ell$ such that $i_{k'} \sim i_k \iff |k-k'| \le 1$. Then we see that a path P is *bad* under an assignment σ if (1) $i_0 \in \mathsf{Bad}(\sigma)$ and (2) $R_{i_{k-1}i_k}$ occurs for all k. Let E_P denote the event that the path P is bad. Note that the events $R_{i_{k-1}i_k}$ are independent for different values of k – indeed, $R_{i_{k-1}i_k}$ depends only on $\mathsf{var}(A_{i_{k-1}}) \cap \mathsf{var}(A_{i_k})$, and so our definition of a path implies that $\mathsf{var}(A_{i_k}) \cap \mathsf{var}(A_{i_{k'}}) = \emptyset$ unless $|k - k'| \leq 1$. Hence

$$\begin{aligned} \Pr[E_P] &= \Pr[(i_0 \in R_0) \wedge R_{i_0 i_1} \wedge (i_1 \in R_1) \wedge R_{i_1 i_2} \wedge \cdots \wedge (i_\ell \in R_\ell) \wedge R_{i_{\ell-1} i_\ell}] \\ &\leq \Pr[A_{i_0} \wedge R_{i_0 i_1} \wedge \cdots \wedge R_{i_{\ell-1} i_\ell}] \leq \Pr[A_{i_0} \wedge R_{i_0 i_1}] \prod_{k=2}^{\ell} r_{i_{k-1} i_k} \end{aligned} \tag{3.3}$$

To show that Res_t is shrinking exponentially, we'd like to prove a bound of the form

$$\mathbb{E}[|\mathsf{Res}_{t+1}| \mid \mathsf{Res}_t] \leq C|\mathsf{Res}_t|, \tag{3.4}$$

for some $C < 1$. Since $\mathsf{Bad}(\sigma_{t+1}) \subseteq \Gamma^+(\mathsf{Res}_t)$, the number of paths $P = i_0, \ldots, i_\ell$ of length ℓ is at most $|\Gamma^+(\mathsf{Res}_t)|\Delta(\Delta-1)^\ell \leq (1+\Delta)\Delta(\Delta-1)^{\ell-1}|\mathsf{Res}_t|$. By a union bound, the expected number of elements that are added to $\mathsf{Res}(\sigma_{t+1})$ during the ℓth iteration of the for loop in Algorithm 2 is at most $\sum_P \Pr[E_P \mid \mathsf{Res}_t]$. If, for a moment, we assume that the bound in (3.3) held for the conditional probability $\Pr[E_P \mid \mathsf{Res}_t]$, it would follow that

$$\mathbb{E}[|\mathsf{Res}_{t+1} \mid \mathsf{Res}_t] \leq \left(p\Delta(1+\Delta) \sum_{\ell=1}^{\infty} (r(\Delta-1))^{\ell-1} \right) |\mathsf{Res}_t| \tag{3.5}$$

which would imply Theorem 3.2.2 (with even weaker hypotheses). However, we still need to show a bound on the conditional probabilities $\sum_P \Pr[E_P \mid \mathsf{Res}_t]$. Given a set $S \subset [m]$ of bad events, we let $B(S)$ denote the event that none of the events in S occur, i.e. $\mathsf{Bad}(\sigma) \subseteq S$. Hence by (3.2), we are trying to bound $\Pr[E_P \mid B(\Gamma^+(\mathsf{Res}_t))]$. For this, the authors of [32] make use of the following lemma from [34], which essentially says that if an event E does not share many dependent variables in common with a set S, then conditioning on $B(S)$ doesn't increase the likelihood of E by too much. It can be proven inductively in a way similar to the Lovasz Local Lemma.

Lemma 3.2.2. *(Theorem 2.1 in [34]): If $x \in \mathbb{R}^+$ is such that $x(1-x)^\Delta \geq p$, then*

$$\Pr[E \mid B(S)] \leq \Pr[E]\,(1-x)^{-|\Gamma(E)\cap S|} \tag{3.6}$$

for any event E and any set $S \subset [m]$.

Incorporating the factor from Lemma 3.2.2 into (3.5), and using the assumptions $\max\{6ep\Delta^2, 3er\Delta\} \leq 1$, one can show that (3.4) holds with a constant $C < 1$. We omit the details of this easy computation, and close this section with a proof that (3.4) implies a logarithmic runtime in expectation.

Lemma 3.2.3. *If (3.4) holds with some $C < 1$, then the expected number of resampled events during Algorithm 3 is at most $\frac{m}{1-C}$, and the expected number of rounds of resampling is at most $\log_{1/C} m + \frac{C}{1-C}$.*

Proof. By (3.2), (3.4) is equivalent to $\mathbb{E}[|\mathsf{Res}_{t+1}| \mid \mathsf{Res}_0, \mathsf{Res}_1, \ldots, \mathsf{Res}_t] \leq C|\mathsf{Res}_t|$, so by the tower property of conditional expectations, $\mathbb{E}|\mathsf{Res}_t| \leq C^t \mathbb{E}|\mathsf{Res}_0| = C^t \cdot m$, and the expected total number of resampling events is $\sum_{t=0}^{\infty} \mathbb{E}|\mathsf{Res}_t| \leq m \cdot \sum_{t=0}^{\infty} C^t = \frac{m}{1-C}$. Let T be the number of rounds before Algorithm 3 terminates. Then

$$\mathbb{E}[T] \leq \log_{1/C} m + \sum_{t > \log_{1/C} m} \Pr[T > t] \leq \log_{1/C} m + \underbrace{\sum_{t > \log_{1/C} m} \mathbb{E}|\mathsf{Res}_t|}_{\leq \sum_{t=1}^{\infty} C^t}$$

and hence $\mathbb{E}[T] \leq \log_{1/C} m + \frac{C}{1-C}$. □

3.3 Sampling w-free strings

In this section, we'll prove Theorem 3.1.1. We first consider a special case of the w-free string sampling problem, for which we can use the simpler Algorithm 1 and give a short proof of its efficiency.

3.3.1 Extremal case: non-translatable w

For a fixed string $w \in \Sigma^*$, let $\Sigma_w^n \subset \Sigma^n$ be the set of length n strings over the alphabet Σ which do not contain w as a (contiguous) substring. In light of Theorem 3.2.1, we also define $\Sigma_{w,1}^n \subset \Sigma^n$ to be the set of strings containing *exactly one* copy of w as a substring. A string $w \in \Sigma^*$ will be called *non-translatable* if no non-trivial prefix of w is also a suffix of w. Equivalently, w is non-translatable iff it cannot be written as

$w = xyx$ for a non-empty string x and some string y. (For example, $w = abcabcd$ is non-translatable but $w = abcabca$ isn't.) Instantiating Algorithm 1 in this context yields the following sampling algorithm:

Algorithm 4: w-free string sampler (non-translatable)

```
sample $s = s_1 \cdots s_n \in \Sigma^n$ uniformly at random
while $s \in \Sigma^* \cdot w \cdot \Sigma^*$ do
    resample all characters involved in each copy of $w$ inside $s$
    update $s$
end
output $s$
```

Proposition 3.3.1. *When Algorithm 1 halts, the string s is a uniformly random element of Σ^n_w. Moreover, the expected number of rounds is $\frac{|\Sigma^n_{w,1}|}{|\Sigma^n_w|}$.*

Proof. The non-translatability of w is precisely what makes this problem amenable to the partial rejection sampling framework of Guo, Jerrum and Liu [32]. More explicitly, let us define the random variables X_i to be the ith character of our string s and the "bad events" A_i to be the event that $X_i X_{i+1} \cdots X_{i+|w|-1} = w$. Clearly A_i and A_j are independent for $|i - j| \geq |w|$. If $1 \leq |i - j| < |w|$, then A_i and A_j are disjoint, since w is non-translatable. Therefore the guarantees of Algorithm 1 apply to Algorithm 4. □

Lemma 3.3.2. *When $|\Sigma| \geq 3$,*

$$\frac{|\Sigma^n_{w,1}|}{|\Sigma^n_w|} \leq n.$$

Proof. Let $k = |w|$. We'll define an injective map from $\Sigma^n_{w,1}$ to $\Sigma^n_w \times [n]$. Let $s = s_1 s_2 \cdots s_n$ be any string containing exactly one copy of w, and let i be the unique index such that $s_i s_{i+1} \cdots s_{i+k-1} = w$. We claim there is some way to modify s by changing only the characters $s_i, \ldots, s_{i+k-1}$ to make it w-free. There are $|\Sigma|^k$ potential ways to reassign these k characters, and we'll argue that at most

$$2\frac{|\Sigma|^k - 1}{|\Sigma| - 1} - 1$$

of them can create an instance of w somewhere else in s. Since this is strictly less than $|\Sigma|^k$ for $|\Sigma| \geq 3$, this will imply the existence of a w-free modification of s on these k characters.

Suppose an assignment of $s_i \cdots s_{i+k-1}$ creates an instance of w in s. Since $s_1 \cdots s_{i-1}$ and $s_{i+k-1} \cdots s_n$ are w-free, the copy of w must overlap this interval, and therefore must have its leftmost endpoint in one of the $2k-1$ locations $i-k+1, i-k+2, \ldots, i+k-1$. The number of assignments to $s_i, \ldots, s_{i+k-1}$ for which $s_j s_{j+1} \cdots s_{j+k-1} = w$ is at most

$$|\Sigma|^{k-|[j,j+k-1]\cap[i,i+k-1]|}$$

and therefore the total number of assignments which can create a w is at most

$$|\Sigma|^{k-1} + \cdots + |\Sigma|^1 + |\Sigma|^0 + |\Sigma|^1 + \cdots + |\Sigma|^{k-1} = 2\frac{|\Sigma|^k - 1}{|\Sigma| - 1} - 1.$$

Now we're ready to define the injection: given a string $s \in \Sigma^n_{w,1}$, let $i(s)$ be the coordinate of the left endpoint of its w-substring, and let s' be (one of) the w-free modifications of s which we have just shown to exist. Then $s \mapsto (s', i(s))$ is clearly an injection into $\Sigma^n_w \times [n]$, since given (s', i), we can recover s by changing the $i, i+1, \ldots, i+k-1$ characters of s' to the corresponding characters of w. □

Remark: The bound in the lemma is tight up to a factor which is constant in n. Indeed, if we send a pair $(s, i) \in \Sigma^n_w \times [n]$ to the string which has w spliced into it at location i, we obtain an element of $\Sigma^n_{w,1}$ (since w is non-translatable this only creates a single copy of it). The number of pre-images of any $s \in \Sigma^n_{w,1}$ under this mapping is at most $|\Sigma|^{|w|}$, which implies the bound

$$\frac{|\Sigma^n_{w,1}|}{|\Sigma^n_w|} \geq n \cdot |\Sigma|^{-|w|}. \tag{3.7}$$

The assumption $|\Sigma| \geq 3$ in the lemma is also necessary. Indeed, consider $\Sigma = \{a, b\}$ and $w = ab$. Then $|\Sigma^n_w| = n + 1$, but $|\Sigma^n_w| \gtrsim n^3$, since it contains all strings of the form $b^i a^j b^k a^{n-(i+j+k)}$. However, for non-translatable w with $|w| \geq 5$, this superlinear behavior cannot happen: it is a direct consequence of Corollary 15 in [32] that the expected number of rounds required by Algorithm 4 is at most $\frac{n}{\frac{|\Sigma|^{|w|}}{e(2|w|-1)} - 1}$ whenever $|\Sigma|^{|w|} > e(2|w| - 1)$. This also shows that, for large enough k, our lower bound (3.7) on the runtime of Algorithm 1 is essentially tight.

3.3.2 Proof of Theorem 3.1.1

For generic w, we'll have to use Algorithm 3. However, since in this case $\Delta = 2|w| - 2$ and $r = \frac{1}{|\Sigma|}$, in order for Theorem 3.2.2 to apply, we would need

$$|\Sigma| > 16|w|. \tag{3.8}$$

It is interesting to ask whether a condition like (3.8) is necessary for Algorithm 3 to run efficiently in this setting. We now show that for $|\Sigma| \geq 3$ it is not necessary.

Let $\Delta = 2(|w| - 2)$ be the maximum degree in the dependency graph, and let $p = |\Sigma|^{-|w|}$ be the probability of the bad event $A_i = \{s_{[i:i+|w|]} = w\}$. In everything that follows, we assume z is some positive number satisfying $z(1-z)^{\Delta} \geq p$, and $\alpha := (1-z)^{-1}$. Set $\delta \leq \Delta$ to be the number of *compatible shifts* of w in either direction, so that δ is, for a typical bad event, the number of neighboring bad events which are not blocked by it (and $\delta = 0 \iff w$ is non-translatable). Let $\sigma_0, \sigma_1 \ldots, \sigma_t$ be the current strings after each iteration of the while loop of Algorithm 3, and let $\mathsf{Bad}_t = \mathsf{Bad}(\sigma_t)$ and $\mathsf{Res}_t = \mathsf{Res}(\sigma_t)$. Finally, let C_t denote the number of connected components of Res_t in the dependency graph.

Lemma 3.3.3. *At any stage t of Algorithm 3, we have*

$$\mathbb{E}[C_{t+1} \,|\mathsf{Res}_t] \leq \mathbb{E}[|\mathsf{Bad}_{t+1}| \,|\, \mathsf{Res}_t] \leq \frac{p\Delta}{2}|\mathsf{Res}_t| + \left(2p\alpha\left(\frac{\alpha^{\Delta/2} - 1}{\alpha - 1}\right) - \frac{p\Delta}{2} + p\right) C_t.$$

Proof. The first inequality is obvious from the nature of Algorithm 2 – it builds Res_{t+1} starting from Bad_{t+1} and attaching unblocked neighbors, so the number C_t of connected components of Res_{t+1} is at most $|\mathsf{Bad}_{t+1}|$. To prove the second inequality, we need to look more closely at the structure of $\Gamma^+(\mathsf{Res}_t)$, which we know contains Bad_{t+1}. Let S be some connected component of Res_t, and we will consider its contribution $\Gamma^+(S)$ to $\Gamma^+(\mathsf{Res}_t)$. Since S is connected, it looks like a line segment with gaps of at most $|w| - 2$ between adjacent events, which will get filled in upon moving to $\Gamma^+(S)$. At each end of the segment, there are up to $|w| - 1$ extra events. Thus,

$$|\Gamma^+(S)| \leq |S| + (|w| - 2)|S| + 2(|w| - 1) = \frac{\Delta}{2}|S| + \Delta. \tag{3.9}$$

More precisely, $\Gamma^+(S)$ contains at most Δ events with any "unfresh" variables – just the ones on the fringes. There are at most $(|w|-1)(|S|-1)+1 = \frac{\Delta}{2}|S| - \frac{\Delta-2}{2}$ interior (i.e. fresh) events, which therefore contribute at most an expected $\frac{p\Delta}{2}|S| - \frac{p(\Delta-2)}{2}$ events to Bad_{t+1}, while the unfresh events contribute at most

$$p \cdot 2 \cdot \sum_{k=1}^{\Delta/2} \alpha^k \leq 2p\alpha \cdot \left(\frac{\alpha^{\Delta/2} - 1}{\alpha - 1}\right)$$

in expectation by Lemma 3.2.2. Summing these contributions over all components S of Res_t, we obtain

$$\mathbb{E}[|\mathsf{Bad}_{t+1}| \mid \mathsf{Res}_t] \quad \leq \quad \frac{p\Delta}{2}|\mathsf{Res}_t| + \left(2p\alpha\left(\frac{\alpha^{\Delta/2} - 1}{\alpha - 1}\right) - \frac{p\Delta}{2} + p\right) C_t$$

□

Lemma 3.3.4. *If*

$$\frac{r\alpha^{|w|} - \alpha r^{|w|}}{\alpha - r} < 1,$$

then at any stage t of Algorithm 3, we have

$$\mathbb{E}[|\mathsf{Res}_{t+1}| \mid \mathsf{Res}_t] \leq B_{t+1} \cdot \left(1 + \frac{\alpha^{\Delta/2} \cdot \delta}{1 - \frac{r\alpha^{|w|} - \alpha r^{|w|}}{\alpha - r}}\right)$$

where B_{t+1} is the upper bound on $\mathbb{E}[|\mathsf{Bad}_{t+1}| \mid \mathsf{Res}_t]$ obtained in Lemma 3.3.3.

Proof. By Lemma 3.2.1, we know that for any event i which is added to Res_{t+1}, there must be a bad path $i_0, i_1, \ldots, i_\ell = i$. Paths of length 0 correspond exactly to Bad_{t+1}, whose expected size we know how to bound from Lemma 3.3.3. For a given i_0, there are at most δ choices for the next event i_1, which also fixes the "direction" of the path (the path moves "right" iff $i_0 < i_1$). From here, the path is parametrized by a tuple

$$(d_1, d_2, \dots, d_{\ell-1})$$

where each $d_i \in \{1, \dots, \frac{\Delta}{2}\}$, and $i_{k+1} = i_k + d_k$ for rightward paths, while $i_{k+1} = i_k - d_k$ for leftward paths. For a tuple $(d_1, \dots, d_{\ell-1})$, the probability (conditional on Res_t) that the corresponding path is bad is at most

$$\begin{aligned}
\Pr[A_{i_0} \wedge R_{i_0 i_1} \wedge \cdots \wedge R_{i_{\ell-1} i_\ell} \,|\, \mathsf{Res}_t] &= \Pr[A_{i_0} \wedge R_{i_1 i_2} \wedge \cdots \wedge R_{i_{\ell-1} i_\ell} \,|\, B([m] \setminus \Gamma^+(\mathsf{Res}_t))] \\
&\le \Pr[A_{i_0} \wedge R_{i_1 i_2} \wedge \cdots \wedge R_{i_{\ell-1} i_\ell}] \cdot \alpha^{|E \setminus \Gamma^+(\mathsf{Res}_t)|} \\
&\le p \cdot r^{(\ell-1)|w| - \sum_{k=1}^{\ell-1} d_k} \cdot \alpha^{|E \setminus \Gamma^+(\mathsf{Res}_t)|}
\end{aligned}$$

where $E = \Gamma^+(\{i_0, i_1, \dots, i_\ell\})$. We can bound

$$\begin{aligned}
|E \setminus \Gamma^+(\mathsf{Res}_t)| &\le |\Gamma^+(i_0) \setminus \Gamma^+(\mathsf{Res}_t)| + |\Gamma^+(i_1) \setminus \Gamma^+(i_0)| + \sum_{k \ge 1}^{\ell-1} |\Gamma^+(i_{k+1}) \setminus \Gamma^+(i_k)| \\
&\le |\Gamma^+(i_0) \setminus \Gamma^+(\mathsf{Res}_t)| + \frac{\Delta}{2} + \sum_{k=1}^{\ell-1} d_k
\end{aligned}$$

and so

$$\Pr[A_{i_0} \wedge R_{i_0 i_1} \wedge \cdots \wedge R_{i_{\ell-1} i_\ell} \,|\, \mathsf{Res}_t] \le p \cdot \alpha^{\Delta/2 + |\Gamma^+(i_0) \setminus \Gamma^+(\mathsf{Res}_t)|} \cdot r^{(\ell-1)|w|} \cdot \left(\frac{\alpha}{r}\right)^{\sum_{k=1}^{\ell-1} d_k}.$$

Summing this bound over all $i_0 \in \Gamma^+(\mathsf{Res}_t)$ and admissible choices of i_1, we obtain the bound

$$\mathbb{E}[|\mathsf{Res}_{t+1} \setminus \mathsf{Bad}_{t+1}| \,|\, \mathsf{Res}_t] \le \alpha^{\Delta/2} \cdot B_{t+1} \cdot \delta \sum_{\ell \ge 1} r^{(\ell-1)|w|} \sum_{(d_1, \dots, d_{\ell-1})} \left(\frac{\alpha}{r}\right)^{\sum_{k=1}^{\ell-1} d_k} \tag{3.10}$$

Observe that for any x, we have

$$\sum_{(d_1, \dots, d_{\ell-1})} x^{\sum_{k=1}^{\ell-1} d_k} = (x + x^2 + \cdots + x^{|w|-1})^{\ell-1} = x^{\ell-1} \left(\frac{x^{|w|-1} - 1}{x - 1}\right)^{\ell-1} \tag{3.11}$$

Plugging in $x = \alpha/r$, we see that

$$\begin{aligned}
\sum_{\ell\geq 1} r^{(\ell-1)|w|} \sum_{(d_1,\ldots,d_{\ell-1})} \left(\frac{\alpha}{r}\right)^{\sum_{k=1}^{\ell-1} d_k} &= \sum_{\ell\geq 1} r^{(\ell-1)|w|} \cdot \left(\frac{\alpha}{r}\right)^{\ell-1} \cdot \left(\frac{\left(\frac{\alpha}{r}\right)^{|w|-1} - 1}{\frac{\alpha}{r} - 1}\right)^{\ell-1} \\
&= \sum_{\ell\geq 1} \left(\frac{r\alpha^{|w|} - \alpha r^{|w|}}{\alpha - r}\right)^{\ell-1} \qquad (3.12) \\
&= \frac{1}{1 - \frac{r\alpha^{|w|} - \alpha r^{|w|}}{\alpha - r}} \qquad (3.13)
\end{aligned}$$

whenever the series converges, i.e. when $\frac{r\alpha^{|w|}-\alpha r^{|w|}}{\alpha-r} < 1$.

So the upper bound in (3.10) becomes

$$\mathbb{E}[|\mathsf{Res}_{t+1} \setminus \mathsf{Bad}_{t+1}| \mid \mathsf{Res}_t] \leq \frac{\alpha^{\Delta/2} \cdot B_{t+1} \cdot \delta}{1 - \frac{r\alpha^{|w|}-\alpha r^{|w|}}{\alpha-r}} \qquad (3.14)$$

Adding in the remaining $\mathbb{E}[|\mathsf{Bad}_{t+1} \mid \mathsf{Res}_t] \leq B_{t+1}$ paths of length 0 yields the lemma.

□

Lemma 3.3.5. *Set*

$$X_t = \frac{p\Delta}{2}|\mathsf{Res}_t| + \left(2p\alpha\left(\frac{\alpha^{\Delta/2}-1}{\alpha-1}\right) - \frac{p\Delta}{2} + p\right) C_t.$$

Then at any stage t of Algorithm 3, we have

$$\mathbb{E}[X_{t+1} \mid \mathsf{Res}_t] \leq \left(\frac{p\Delta}{2}\left(\frac{\alpha^{\Delta/2}\delta}{1 - \frac{r\alpha^{|w|}-\alpha r^{|w|}}{\alpha-r}}\right) + 2p\alpha\left(\frac{\alpha^{\Delta/2}-1}{\alpha-1}\right) + p\right) X_t$$

Proof. This follows immediately by taking the appropriate linear combination of the bounds in Lemmas 3.3.3 and 3.3.4. □

Corollary 3.3.1. *For any string w with $|w| \geq 2$ and alphabet with $|\Sigma| \geq 3$, Algorithm 3 outputs a uniformly random element of Σ_w^n in $O(\log n)$ rounds.*

Proof. Let $|\Sigma| \geq 3$. The upper bound in Lemma 3.3.5 is clearly increasing in δ, so we can assume $\delta = \Delta$. For $|\Sigma| \geq 4$, plugging this value of δ into Lemma 3.3.5, along with $\alpha = (1+1/\Delta)$ already yields the corollary (via Lemma 3.2.3). For $|\Sigma| = 3$, this works for $|w| \notin \{2,3,4\}$. For these remaining lengths, we can choose α more optimally: set z equal to the smallest value of x for which $x(1-x)^{\Delta} \geq 3^{-|w|}$, and pick $\alpha = (1-z)^{-1}$. This tightens the bounds enough to work in these cases, as can be easily verified. □

One can easily extend the arguments in this section to the setting where the string being sampled is actually a circle, or a figure eight, or any closed "curve" with a bounded number of self-intersections. In this case, the dependency graph is essentially a union of intersecting cycles (with extra local edges, as in the string case, which corresponds to a path with extra local edges). It is easy to see that the proofs of Lemmas 3.3.3 and 3.3.4 go through in this setting, paying only a constant factor $O(\eta)$ in the bounds, where η is the number of self intersections. We leave the tedious details to the skeptical reader.

Corollary 3.3.2. *Let C be a curve of length n with η self-intersections. Then Algorithm 3 finds a labeling of C by Σ which avoids a pattern w in $O(\log n)$ rounds in expectation, provided that $|\Sigma|^{|w|} \gtrsim \eta$.*

Note that when $\delta = 0$, Algorithm 3 reduces to Algorithm 4, and hence Lemma 3.3.5 can be used to obtain a runtime bound for Algorithm 1. In this case, Algorithm 1 takes $O(\log n)$ rounds when

$$\frac{2}{|\Sigma|^{|w|}} \cdot \left(\frac{1}{2} + \frac{\alpha^{|w|} - \alpha}{\alpha - 1}\right) < 1$$

for any $\alpha = (1-z)^{-1}$ with $z(1-z)^{2|w|-2} \geq |\Sigma|^{-|w|}$. When $|\Sigma| = 2$, this says nothing for $|w| = 2,3,4$, but for $|w| \geq 5$ we can take $z = 0.0453$ and $\alpha = 1.0475$, so that

$$\frac{2}{2^5} \cdot \left(\frac{1}{2} + \frac{1.0475^5 - 1.0475}{0.0475}\right) \approx 0.312 < 1.$$

Corollary 3.3.3. *For non-translatable $w \in \Sigma^n$, Algorithm 1 takes $O(\log n)$ rounds in expectation, except possibly for $|\Sigma| = 2$ and $|w| = 2,3,4$.*

3.4 Sampling H-free subgraphs of a grid graph

Given (G, H, λ), our goal is to sample from the following distribution, supported on H-free subgraphs of G:

$$\Pr(G') \propto \begin{cases} \lambda^{e(G')}(1-\lambda)^{e(G)-e(G')} \propto \left(\frac{\lambda}{1-\lambda}\right)^{e(G')} & \text{if } G' \subseteq G \text{ is } H\text{-free} \\ 0 & \text{otherwise} \end{cases} \tag{3.15}$$

We can think of this problem as a special case of the hard core model[5] in $|H|$-uniform hypergraphs – indeed, the vertices correspond to the edges of G, and the hyperedges correspond to the copies of H in G. When $\lambda = 1/2$, the problem is simply asking for a uniformly random H-free subgraph of an input graph. Viewed another way, each instance of $(H, \frac{1}{2})$ corresponds to a monotone $e(H)$-CNF

$$\bigwedge_{(e_1,\ldots,e_{e(H)})\cong H} (\neg e_1 \vee \neg e_2 \vee \cdots \vee \neg e_{e(H)})$$

whose clauses correspond to copies of H and whose variables correspond to edges. Here we want to sample a uniformly random satisfying assignment. On instances G in which each edge appears in at most k different copies of H inside G, we say the corresponding CNF is a *read-k-monotone CNF*. When $k = 2$ (as it will be in the case of planar grid graphs), an algorithm of Lin, Liu and Lu [50] gives a FPTAS for approximately counting the set of satisfying assignments to a read-twice-monotone-CNF. Using the standard reduction from approximate counting to approximate sampling for self-reducible problems, this implies the following theorem:

Theorem 3.4.1 (Lin, Liu, Lu 2012)**.** *For $\lambda \leq 1/2$, there is an algorithm which ε-approximately samples from the distribution (3.15), provided that each edge of G appears in at most 2 different copies of H. The running time is $O(n^{3.58}m^2 \cdot (1/\varepsilon)^{2.58})$, where $n = |E(G)|$ and m is the number of copies of H in G.*

Liu and Lu [51] subsequently gave a FPTAS for counting solutions to read-k-

[5] In most work on the hardcore model, the fugacity parameter λ plays the role that the quantity $\frac{\lambda}{1-\lambda}$ plays in our notation.

monotone CNF, for $k \leq 5$. However, their algorithm is somewhat impractical, with a provable runtime bound on the order of $(n/\epsilon)^{144}$. For $k \geq 6$, there is no FPTAS (resp. FPRAS) for this problem unless NP = P (resp. NP = RP).

3.4.1 Triangle-free subgraphs of the triangular grid

We now analyze the performance of Algorithm 3 on instances of the form (T, K_3, λ), where T is a subgraph of the $m \times m$ triangular grid graph. (Throughout this section, when we speak of Algorithm 3, we implicitly assume that it is running on such an instance.) Here, the bad events correspond to the presence of each triangle, so $\Delta = 3$ and $p = \lambda^3$. Since any pair of dependent triangles share exactly one edge, $r_{ij} = \lambda$ for every $i \sim j$. Applying Theorem 3.2.2 directly, we see that for

$$\lambda \leq \frac{1}{9e} \approx 0.04088$$

Algorithm 3 solves this problem efficiently. Our present goal is to extend this range of λ.

Lemma 3.4.1. *At any stage t during Algorithm 3,*

$$|\Gamma^+(\mathsf{Res}_t)| \leq 2.5 \cdot |\mathsf{Res}_t|$$

Proof. Let $\Delta \in \mathsf{Bad}_t$ be an occurring triangle in round t. It suffices to prove the expansion inequality for the connected component S of Res_t containing Δ. There are a handful of cases to consider, depending on the number of neighbors of Δ in the dependency graph, which of course depends on the graph T. Since all edges of Δ are present by assumption, each of its neighbors will be added to S during stage 1 of Algorithm 5, so $S_1 := \Gamma^+(\{\Delta\}) \subseteq S$. In each case $|S_1| \in \{1, 2, 3, 4\}$, we can check that

$$\frac{|\Gamma^+(S_1)|}{|S_1|} \leq 2.5$$

holds. Now we observe that attaching a triangle to any non-empty set S' of triangles increases $|S'|$ by 1 and $|\Gamma^+(S')|$ by at most 2. Since $\frac{a}{b} \leq 2.5 \implies \frac{a+2k}{b+k} \leq 2.5$ for any

$k \geq 0$, the inequality remains true for S, and hence for Res_t. □

Proposition 3.4.2. *If*

$$2.5\lambda^3 + \frac{7.5\lambda^3}{1 - \frac{2\lambda}{1+\lambda}} < 1$$

then Algorithm 3 takes an expected $O(\log m)$ rounds. In particular, we may take $\lambda \leq 0.3748$.

Proof. At a high level, instead of counting bad paths, we directly estimate the expected number of events added to Res_t during each phase of Algorithm 5, conditioned on the previous rounds. More precisely, let R^t_ℓ be the intermediate set at stage ℓ in Algorithm 5, during round t of Algorithm 3 (so in particular $R^t_0 = \mathsf{Bad}_t$ and $R^t_\infty = \mathsf{Res}_t$). We will show that for $\ell \geq 1$,

$$\mathbb{E}[|R^{t+1}_{\ell+1}| \mid R^{t+1}_\ell, \ldots, R^{t+1}_0, \mathsf{Res}_t] \leq C(\lambda)|R^{t+1}_\ell| \tag{3.16}$$

for some constant $C(\lambda)$. Since $\mathbb{E}[|R^{t+1}_0| \mid \mathsf{Res}_t] \leq \lambda^3|\Gamma^+(\mathsf{Res}_t)| \leq 2.5\lambda^3|\mathsf{Res}_t|$ (by Lemma 3.4.1) and $|R^{t+1}_1| \leq 3|R^{t+1}_0|$, we have

$$\mathbb{E}[[|R^{t+1}_1| \mid \mathsf{Res}_t] \leq 7.5\lambda^3|\mathsf{Res}_t|.$$

To upper bound $\mathbb{E}[|R^{t+1}_2| \mid R^{t+1}_1, R^{t+1}_0, \mathsf{Res}_t]$, we observe that each event (i.e. triangle) in R^{t+1}_2 is one of the $\leq 2|R^{t+1}_1|$ triangles which share an edge with a triangle in R^{t+1}_1 and do not belong to $R^{t+1}_0 \cup R^{t+1}_1$. For each triangle Δ in R^{t+1}_1, we know that a certain edge (and possibly one more) is present (the common edge(s) between Δ and R^{t+1}_0), and that *not all three edges in Δ are present* – otherwise it would have been in R^{t+1}_0. If Δ shares two edges with R^{t+1}_0, we know for sure the third edge is not present, and so it contributes nothing to R^{t+1}_2. Otherwise, we claim that

$$\begin{aligned} \Pr(\Delta \text{ has another edge} \mid R^{t+1}_1, R^{t+1}_0, \mathsf{Res}_t) &\leq \Pr(\Delta \text{ has another edge} \mid \Delta \notin \mathsf{Bad}_{t+1}) \\ &= \frac{2\lambda}{1+\lambda} \end{aligned}$$

and hence $\mathbb{E}[|R^{t+1}_2| \mid R^{t+1}_1, R^{t+1}_0, \mathsf{Res}_t] \leq \frac{2\lambda}{1+\lambda} \cdot |R^{t+1}_1|$. The same argument works for 2 and 1 replaced by $\ell+1$ and ℓ, and so (3.16) holds with $C(\lambda) = \frac{2\lambda}{1+\lambda}$. By the tower property of conditional expectations, it follows that

$$\mathbb{E}[|R_{\ell+1}^{t+1}| \mid \mathsf{Res}_t] \leq C(\lambda)^{\ell} \cdot 7.5\lambda^3 |\mathsf{Res}_t|. \tag{3.17}$$

Summing over $\ell \geq 0$, we see that

$$\mathbb{E}[|\mathsf{Res}_{t+1}| \mid \mathsf{Res}_t] \leq \left(2.5\lambda^3 + \frac{7.5\lambda^3}{1-\frac{2\lambda}{1+\lambda}}\right) |\mathsf{Res}_t|$$

from which the proposition follows. □

3.4.2 Square-free subgraphs of the square grid

We now analyze the performance of Algorithm 3 on instances of the form (G, C_4, λ) problem, where G is a subgraph of the $m \times m$ square grid graph. Here the bad events correspond to the presence of each square, so $\Delta = 4$ and $p = \lambda^4$. Since any pair of dependent squares share exactly one edge, $r_{ij} = \lambda$ for every $i \sim j$. Applying Theorem 3.2.2 directly, we see that for

$$\lambda \leq \frac{1}{12e} \approx 0.03065$$

Algorithm 3 solves this problem efficiently. Our present goal is to extend this range of λ.

Lemma 3.4.3. *At any stage t during Algorithm 3,*

$$|\Gamma^+(\mathsf{Res}_t)| \leq 3 \cdot |\mathsf{Res}_t|$$

Proof. The idea is the same as in the proof of Lemma 3.4.1. Let $\square \in \mathsf{Bad}_t$, and let S be the component of Res_t containing $\square$, which also must contain $S_1 := \Gamma^+(\{\square\})$. It is easy to check that for every possible S_1 we have

$$\frac{|\Gamma^+(S_1)|}{|S_1|} \leq 3,$$

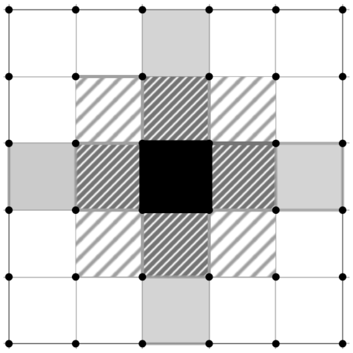

Figure 3-1: All squares which can be added to Res_{t+1} in stage 1 or 2 of Algorithm 5 due to $i_0 \in \mathsf{Bad}_{t+1}$.

and since attaching an additional square to any nonempty set S' of squares increases $|S'|$ by 1 and $|\Gamma^+(S')|$ by at most 3, the inequality persists all the way to S and hence to Res_t. □

Proposition 3.4.4. *Let G be a subgraph of the $m \times m$ square grid. If*

$$3\lambda^4 \left(5 + \frac{4\lambda - 4\lambda^2 - 12\lambda^3}{1 - 2\lambda - \lambda^2}\right) < 1$$

then sampling a square-free subgraph with Algorithm 3 takes an expected $O(\log m)$ rounds. In particular we may take $\lambda \leq 0.4063$.

Proof. For each $i_0 \in \Gamma^+(\mathsf{Res}_t)$, we bound the expected number of squares which are added to Res_{t+1} as a result of a bad path rooted at i_0. All probabilities will be conditional on Res_t, that is, conditional on a certain set of squares *not* being present – however, by monotonicity[6], we can ignore this conditioning and exploit the resulting independence. The square i_0 (represented in black in Figure 3-1 above) is present with probability at most λ^4. If i_0 is present, it automatically brings the ($\leq$) four adjacent (heavy-hatched) squares into Res_{t+1}. Each of the four light-hatched squares can only be added to Res_{t+1} if the two edges they share with the heavy-hatched squares are present, which happens with probability at most λ^2. If those edges are not present, the squares will be *blocked* by the heavy-hatched squares and hence not added to Res_{t+1}. Each of the light-grey squares will be blocked unless their edge bordering a

[6] more specifically, the FKG inequality

heavy-hatched square is present, which happens with probability at most λ. Thus, the expected number of squares added due to i_0 during stages 1 and 2 of Algorithm 5 is at most

$$\lambda^4 \left(\underbrace{5}_{i_0 \text{ and its 4 neighbors}} + \underbrace{4\lambda^2}_{\text{light-hatched squares}} + \underbrace{4\lambda}_{\text{light grey squares}} \right)$$

Next we bound the expected number of squares added due to paths $i_0, i_1, \dots, i_\ell$, with $\ell > 2$ and $i_k \sim i_{k'} \iff |k - k'| = 1$. This last condition is stronger than that of a self-avoiding walk – even the *neighbors* of previously visited sites must be avoided on the next step – so we'll call them *very self-avoiding walks*, or vSAWs. We can break up the set of such walks into two categories – S and D – based on their first two moves: either the first two moves are the same (e.g. left, left or up, up) or the two moves are different (e.g. up, right). We claim that for $\ell \geq 2$,

$$S_{\ell+1} \leq S_\ell + 2D_\ell \tag{3.18}$$

$$D_{\ell+1} \leq S_\ell + D_\ell \tag{3.19}$$

where S_ℓ and D_ℓ count the number of length-ℓ vSAWs of type S and D respectively starting from a fixed square. Indeed, if the first two moves of a length $\ell + 1$ vSAW are the same – say left, left – then up, down, left are the next available moves, which correspond to walks of length ℓ of type D, D, and S respectively, proving (3.18). To prove (3.19), suppose without loss that the first two moves are left, up. Then the next move is either right or up, since a left move would land on a neighbor of the original square. This yields (3.19). Combining all of the above, we know that each $i_0 \in \Gamma^+(\mathsf{Res}_t)$ contributes at most

$$5\lambda^4 + \lambda^4 \cdot \left\langle \sum_{\ell \geq 0}^{\infty} \begin{pmatrix} \lambda & 2\lambda \\ \lambda & \lambda \end{pmatrix}^{\ell} \begin{pmatrix} 4\lambda \\ 4\lambda^2 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\rangle = \lambda^4 \left(5 + \frac{4\lambda - 4\lambda^2 - 12\lambda^3}{1 - 2\lambda - \lambda^2} \right)$$

to $\mathbb{E}[|\mathsf{Res}_{t+1}| \mid \mathsf{Res}_t]$ (where we have used the identity $\sum_{\ell \geq 0} A^\ell = (I - A)^{-1}$). Summing up these contributions over all $|\Gamma^+(\mathsf{Res}_t)| \leq 3|\mathsf{Res}_t|$ (Lemma 3.4.3), we have shown

$$\mathbb{E}[|\mathsf{Res}_{t+1}| \mid \mathsf{Res}_t] \leq 3\lambda^4 \left(5 + \frac{4\lambda - 4\lambda^2 - 12\lambda^3}{1 - 2\lambda - \lambda^2}\right).$$

By Lemma 3.2.3, this implies the proposition. □

A very similar argument also gives an improvement over Theorem 3.2.2 for sampling cube-free subgraphs of a three dimensional cubic lattice. In this setting, $p = \lambda^{12}$, $r = \lambda$ and $\Delta = 18$, so the bound from Theorem 3.2.2 is $1/54e < 0.00682$. In the subsequent proposition we sketch a proof of a significantly improved bound. A distinguishing feature of this problem is that, in the cube, each edge belongs to *four* bad events (versus only *two* in grid graphs). In other words, the natural CNF representation is read-four-times, so Theorem 3.4.1 does not apply.

Proposition 3.4.5. *Let G be any subgraph of the $m \times m \times m$ cube. If*

$$11\lambda^{12} \left(1 + \frac{18 - 138\lambda - 36\lambda^4}{1 - 7\lambda - \lambda^4 - 25\lambda^5}\right) < 1$$

then sampling a cube-free subgraph with Algorithm 3 takes an expected $O(\log m)$ rounds. In particular we may take $\lambda \leq 0.142588$.

Proof. Starting from an occurring cube, there are 18 ways a bad path could proceed – 6 lateral moves, one through each face (call these F moves) and 12 diagonal moves, one through each edge (call these D moves). A moment's reflection shows that at any point along a bad path, if the last move was an F move, then for the next move, there is at most 1 available F move and at most 8 available D moves; while if the last move was a D move, then there are at most 4 available F moves and 7 available D moves. Each F move is possible if a particular square is present, which happens with probability at most λ^4 (even when conditioned on Res_t, by monotonicity). Similarly, each D move is possible if a particular edge is present, which happens with probability at most λ. Therefore, the sum over all potential bad paths, weighted by their probability of being bad given Res_t is at most

$$\lambda^{12} \cdot |\Gamma^+(\mathsf{Res}_t)| + \lambda^{12} \cdot |\Gamma^+(\mathsf{Res}_t)| \left\langle \sum_{\ell \geq 0}^{\infty} \begin{pmatrix} \lambda^4 & 8\lambda \\ 4\lambda^4 & 7\lambda \end{pmatrix}^{\ell} \begin{pmatrix} 6 \\ 12 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\rangle \tag{3.20}$$

In the spirit of Lemmas 3.4.1 and 3.4.3, one can prove a bound of $|\Gamma^+(\mathsf{Res}_t)| \leq 11|\mathsf{Res}_t|$. We omit the details, since this constant has only negligible effect on the cutoff value for λ. The rest of (3.20) can be evaluated explicitly (using the identity $\sum_{\ell \geq 0} A^\ell = (I - A)^{-1}$) to yield the proposition. □

3.5 Open questions

- Numerical experiments suggest a phase transition in the expected runtime of Algorithm 3 on grid graphs as a function of the edge probability λ. However, we are currently unable to prove even much weaker statements, such as the existence of an $\epsilon > 0$ for which this algorithm takes super-polynomial time when $\lambda \in (1 - \epsilon, 1)$. The critical value of λ reflects something about the intrinsic geometry of the lattice, and could perhaps have an interpretation in the language of statistical physics.

- Our simulations also suggest that Theorem 3.1.1 holds even for $|\Sigma| = 2$, as long as $|w| \geq 3$. Proving this would likely require a bound stronger than the one from Lemma 3.2.2.

- Consider a scenario with two sets of bad events, A_i and B_i, such that (i) $A_i \sim A_j \implies \Pr[A_i \cap A_j] = 0$, (ii) $B_i \sim B_j \implies \Pr[B_i \cap B_j] = 0$, and (iii) $A_i \cap B_i = \emptyset$. (For example: in a randomly oriented graph G, let A_i and B_i be the events that vertex i is a sink or a source, respectively.) Is there a way to sample a uniformly random assignment satisfying $\bigcap_i (A_i \cap B_i)$ that leverages the extremality of A and B separately?

- It would of course be interesting to apply partial rejection sampling to other problems outside the regime of Theorem 3.2.2, to see how it performs in practice. It probably won't come as close to hardness thresholds as MCMC methods can, but our results in this chapter offer some evidence that PRS should at least be considered when one needs a quick and dirty uniform sampler.

3.6 Appendix: Hard sphere model

Originating (like so many things) from statistical physics, the *hard sphere model* is a simple probability distribution used to model positions of particles of a contained gas, supported on configurations of non-overlapping disks of radius r in a bounded region of $\mathbb{R}^d$. More precisely, the centers of the spheres are sampled from a Poisson process on $\Lambda \subset \mathbb{R}^d$, conditioned on the spheres being non-overlapping. When $d = 2$, this is called the hard disks model, and was the original object of study in the seminal work of Metropolis et al. on MCMC methods [54].

In recent work of Guo and Jerrum [31], the authors show that for sufficiently small intensities $\lambda \leq \overline{\lambda}_d$, a PRS-based algorithm (Algorithm 5 described below) is efficient in the sense that in expectation, it only requires $O(\log 1/r)$ rounds of resampling.

Theorem 3.6.1. (Guo, Jerrum [31]) *Algorithm 5 for the hard disks model with parameter $\lambda_r = \frac{\lambda}{\pi r^2}$ takes $O(\log 1/r)$ rounds of resampling in expectation for $\lambda \leq \overline{\lambda}_2 :=$ 0.21027. The same conclusion holds for the hard sphere model in $d > 2$ dimensions with parameter $\lambda_r = \frac{\lambda}{\textbf{\textit{vol}}(B_r^d(0))}$ for $\lambda \leq \overline{\lambda}_d := 2^{-(d+\frac{1}{2})}$.*

Combined with a result of Jenssen, Joos and Perkins [42] which relates the intensity parameter λ to the expected packing density α for the hard sphere model, Theorem 3.6.1 yields an efficient method of generating sphere packings of density $\Omega(2^{-d})$. This matches the best known algorithms up to a constant factor, which all fall short of the density $\Omega(d2^{-d})$ packings which are known to exist.[7] The authors of [31] conjecture that $\overline{\lambda}_2$ can be taken to be ≈ 0.5, according to their simulations. In this note, we slightly improve the value of $\overline{\lambda}_d$ for all d:

Theorem 3.6.2. *The constant $\overline{\lambda}_d$ in Theorem 3.6.1 can be improved by factor of $(1 + 2^{-O(d)})$. In particular, $\overline{\lambda}_2 \geq 0.2344+$.*

[7] Very recently, Helmuth, Perkins and Petti [38] have shown that a certain Markov chain (called *single-center dynamics*) mixes rapidly to the hard sphere distribution with parameter λ, as long as $\lambda \leq 2^{1-d}$. This improves by a constant the best expected packing density of any known efficient algorithm, to $(0.8526 + o_d(1))2^{-d}$.

3.6.1 A partial rejection sampler for hard spheres

We now describe the sampling algorithm used by Guo and Jerrum in [31], and introduce some notation. Let $B_r^d(x)$ be the open ball of radius r centered at $x \in \mathbb{R}^d$. If P is a finite subset of $[0,1]^d$, we define the sets

$$\begin{aligned} \mathsf{BadPairs}(P) &:= \{\{x,y\} : x, y \in P \text{ and } \|x-y\|_2 \le 2r\} \\ \mathsf{BadPoints}(P) &:= \bigcup_{S \in \mathsf{BadPairs}(P)} S \end{aligned}$$

It is shown in [31] that the set of points $P = \{x_1, \ldots, x_k\}$ output by Algorithm 5 below is distributed as a Poisson process on $[0,1]^d$ with intensity λ_r, conditioned on $\|x_i - x_j\|_2 > 2r$ for all $i \neq j$, and moreover that the analogue of (3.2) holds at each step.

Algorithm 5: Partial rejection sampling for hard spheres in $[0,1]^d$

```
sample $P$ from a Poisson process of intensity $\lambda_r = \frac{\lambda}{\mathbf{vol}(B_r^d(0))}$
while $\mathsf{BadPoints}(P) \neq \emptyset$ do
    set $S = \bigcup_{x \in \mathsf{BadPoints}(P)} B_{2r}^d(x)$
    Let $P^S$ be a sample from the Poisson process of intensity $\lambda_r$ on S
    update $P \leftarrow P^S \cup (P \setminus \mathsf{BadPoints}(P))$
end
output $P$
```

To prove that the while loop terminates after $O(d \log \frac{1}{r})$ iterations in expectation, the authors argue that the number of $\mathsf{BadPairs}$ is decreasing exponentially for sufficiently small λ. More precisely, if $P_0, P_1, \ldots, P_t$ denotes the set P after each update, then the authors show $\mathbb{E}[|\mathsf{BadPairs}(P_{t+1})| \,|\, P_t, \ldots, P_0] = \mathbb{E}[|\mathsf{BadPairs}(P_{t+1})| \,|\, P_t] \le C(\lambda)|\mathsf{BadPairs}(P_t)|$, for some $C(\lambda)$ which goes to 0 as $\lambda \to 0$. Since $|\mathsf{BadPairs}(P_0)| \le \frac{1}{2}\lambda_r^2$, it then follows from Lemma 3.2.3 that the expected number of loops is $O(\log(\lambda_r)) = O(d \log \frac{1}{r})$ for λ sufficiently small.

3.6.2 Pushing spheres apart

To improve upon Guo and Jerrum's analysis of Algorithm 5 (i.e. to improve the upper bounds on $C(\lambda)$), we need the following intuitive geometric fact:

Fact 3.6.1. *Let $\gamma > 1$, and $x_1, \ldots, x_n \in \mathbb{R}^d$. Then*

$$\mathbf{vol}\left(\bigcup_{i=1}^{n} B_r^d(x_i)\right) \leq \mathbf{vol}\left(\bigcup_{i=1}^{n} B_r^d(\gamma x_i)\right).$$

While intuitively obvious, proving Fact 3.6.1 is not trivial. In fact it is a special case of the *Kneser-Poulsen conjecture*. Let us say the point set $Y = \{y_1, \ldots, y_n\} \subset \mathbb{R}^d$ is an *expansion* of $X = \{x_1, \ldots, x_n\}$ if $\|x_i - x_j\|_2 \leq \|y_i - y_j\|_2$ for all i, j. The Kneser-Poulsen conjecture posits that, for any radii $r_1, \ldots, r_n$, and point sets $X, Y \subset \mathbb{R}^d$ such that Y is an expansion of X,

$$\mathbf{vol}\left(\bigcup_{i=1}^{n} B_{r_i}^d(x_i)\right) \leq \mathbf{vol}\left(\bigcup_{i=1}^{n} B_{r_i}^d(y_i)\right). \tag{3.21}$$

The conjecture remains open, although certain cases are known. We say Y is a *continuous expansion* of $X = \{x_1, \ldots, x_n\}$ if there exist continuous functions $x_i(t)$ such that $X(t) = \{x_1(t), \ldots, x_n(t)\}$ is an expansion of $X(t')$ for all $t > t'$, and $X(0) = X$, $X(1) = Y$. It is known that when Y is a continuous expansion of X then inequality (3.21) holds – this was proved first for disks in $\mathbb{R}^2$ of equal radii by Bollobas [9], then for arbitrary radii in $\mathbb{R}^2$ and eventually in any dimension by Csikós [17], which is enough to imply Fact 3.6.1. The following consequence of this fact is what we'll actually use:

Lemma 3.6.2. *Let $C = \cup_{i=1}^{\ell} B_{2r}^d(x_i)$ be the union of ℓ spheres of radius $2r$ in $\mathbb{R}^d$. Then for any $s > 0$,*

$$\int_C \int_C 1_{\|x-y\| \leq s}\, dx\, dy \geq \frac{\mathbf{vol}(C)}{\mathbf{vol}(B_{2r}^d(0))} \cdot \int_{B_{2r}^d(0)} \int_{B_{2r}^d(0)} 1_{\|x-y\| \leq s}\, dx\, dy.$$

Proof. The desired inequality is clearly equivalent to

$$\mathop{\mathbb{E}}_{x \in C} \mathbf{vol}(C \cap B_s^d(x)) \geq \mathop{\mathbb{E}}_{x \in B_{2r}^d(0)} \mathbf{vol}(B_{2r}^d(0) \cap B_s^d(x))$$

where $f(x) := \mathbf{vol}(C \cap B_s^d(x))$ and $g(x) := \mathbf{vol}(B_{2r}^d(0) \cap B_s^d(x))$. This is in turn equivalent to the inequality

$$\frac{1}{\mathbf{vol}(C)}\int_0^\infty \mathbf{vol}(x\in C : f(x)>t)\,dt \geq \frac{1}{\mathbf{vol}(B_{2r}^d(0))}\int_0^\infty \mathbf{vol}(x\in B_{2r}^d(0) : g(x)>t).$$

We claim that the integrand on the left dominates the one on the right pointwise – that is, for each $t\geq 0$, we have

$$\frac{\mathbf{vol}(x\in C : f(x)>t)}{\mathbf{vol}(C)} \geq \frac{\mathbf{vol}(x\in B_{2r}^d(0) : g(x)>t)}{\mathbf{vol}(B_{2r}^d(0))}. \tag{3.22}$$

Observe that for each t, the set $\{x\in B_{2r}^d(0) : g(x)>t\}$ is an open ball $B_{2\alpha r}^d(0)$ for some $\alpha=\alpha(t)\leq 1$. Then clearly $\cup_{i=1}^{\ell} B_{2\alpha r}^d(x_i)\subseteq\{x\in C : f(x)>t\}$. Hence to prove (3.22), it suffices to show that

$$\frac{\mathbf{vol}(\cup_{i=1}^{\ell} B_{2\alpha r}^d(x_i))}{\mathbf{vol}(B_{2\alpha r}^d(0))} \geq \frac{\mathbf{vol}(\cup_{i=1}^{\ell} B_{2r}^d(x_i))}{\mathbf{vol}(B_{2r}^d(0))}. \tag{3.23}$$

Consider applying the transformation $x\mapsto x/2\alpha r$ on $\mathbb{R}^d$. This sends $B_{2\alpha r}^d(x_i)\mapsto B_1^d(x_i/2\alpha r)$ and scales all volumes by $1/(2\alpha r)^d$. Similarly the transformation $x\mapsto x/2r$ takes $B_{2r}^d(x_i)\mapsto B_1^d(x_i/2r)$ and scales volumes by $1/(2r)^d$. Hence, (3.23) is equivalent to

$$\mathbf{vol}\left(\bigcup_{i=1}^{\ell} B_1^d(x_i/2\alpha r)\right) \geq \mathbf{vol}\left(\bigcup_{i=1}^{\ell} B_1^d(x_i/2r)\right) \tag{3.24}$$

which is Fact 3.6.1, with $\gamma = 1/\alpha$. □

3.6.3 Proof of Theorem 3.6.2

Recall that P_t is the set of points sampled during round $t\geq 0$ of the algorithm. Suppose $|\mathsf{BadPairs}(P_t)|=k_t$. Then the resampling set $S_t\subset[0,1]^d$ is $\bigcup_{x\in\mathsf{BadPoints}(P_t)} B_{2r}^d(x)$. Let

$$\begin{aligned}
k' &= \mathbb{E}[k_{t+1}\,|\,\mathsf{BadPoints}(P_t)]\\
j' &= \mathbb{E}[\#\{(x,y)\in S_t\times[0,1]^d : \|x-y\|\leq 2r\}\cap P_{t+1}^2\,|\,\mathsf{BadPoints}(P_t)]\\
\ell' &= \mathbb{E}[\#\{(x,y)\in S_t\times S_t : \|x-y\|\leq 2r\}\cap P_{t+1}^2\,|\,\mathsf{BadPoints}(P_t)]
\end{aligned}$$

Since each unordered bad pair $\{x, y\}$ with $x, y \in S_t$ gets counted twice in j' and twice in ℓ', we have

$$k' = j' - \frac{\ell'}{2}. \tag{3.25}$$

It is shown in [31] that the hard disks process can be coupled to a Poisson process in such a way that the latter configuration always contains the former. In particular,

$$\begin{aligned} j' &\leq \int_{S_t} \lambda_r \int_{[0,1]^d} \lambda_r 1_{\|x-y\| \leq 2r}\, dx\, dy \\ &\leq \lambda_r^2 \cdot \mathbf{vol}(S_t) \cdot \mathbf{vol}(B_{2r}^d(0)) = \frac{2^d \lambda^2}{\mathbf{vol}(B_r^d(0))} \cdot \mathbf{vol}(S_t) \end{aligned} \tag{3.26}$$

On the other hand, by Lemma 3.6.2, we have

$$\begin{aligned} \frac{\ell'}{2} &= \frac{1}{2} \int_{S_t} \lambda_r \int_{S_t} \lambda_r 1_{\|x-y\| \leq 2r}\, dx\, dy \\ &\geq \frac{\mathbf{vol}(S_t)}{\mathbf{vol}(B_{2r}^d(0))} \cdot \frac{1}{2} \int_{B_{2r}^d(0)} \lambda_r \int_{B_{2r}^d(0)} \lambda_r 1_{\|x-y\| \leq 2r}\, dx\, dy & (3.27) \\ &= \frac{2^{d-1} \lambda^2 \Pr_{x,y \in B_1^d(0)} \left[\|x-y\|_2 \leq 1\right]}{\mathbf{vol}(B_r^d(0))} \cdot \mathbf{vol}(S_t) & (3.28) \end{aligned}$$

and hence

$$k' \leq \frac{\mathbf{vol}(S_t)}{\mathbf{vol}(B_r^d(0))} \cdot \lambda^2 \cdot 2^d \cdot \left(\frac{1}{2} - \Pr_{x,y \in B_1^d(0)} \left[\|x-y\|_2 \leq 1\right]\right). \tag{3.29}$$

Recall that S_t is the union of at most $2k_t$ copies of B_{2r}^d, each of which must overlap with at least one other disk – in fact, for each $B_{2r}^d(x)$ in S_t, there must be an overlapping $B_{2r}^d(x')$ in S_t with $\|x - x'\| < 2r$. Therefore $\mathbf{vol}(S_t)$ is maximized when S_t is a union of k_t connected components, each of which is a translated copy of $B_{2r}^d(0) \cup B_{2r}^d(2r\mathbf{e}_1)$. Therefore

$$k' \leq \lambda^2 \cdot 2^{2d} \cdot \underbrace{\frac{\mathbf{vol}(B_1^d(0) \cup B_1^d(\mathbf{e}_1))}{\mathbf{vol}(B_1^d(0))}}_{=:V_d} \cdot \underbrace{\left(\frac{1}{2} - \Pr_{x,y \in B_1^d(0)} \left[\|x-y\|_2 \leq 1\right]\right)}_{=:P_d} \cdot k_t \tag{3.30}$$

d	2	3	4	5	6	7	8	9	10	11	12
C_d	1.32619	1.24419	1.18882	1.14913	1.11957	1.09693	1.07925	1.06523	1.05399	1.04488	1.03744
$\overline{\lambda}_d$	0.23444	0.10997	0.052539	0.025393	0.01237	0.00606	0.00298	0.00147	0.000728	0.000361	0.000179

Table 3.1: Values of C_d and $\overline{\lambda}_d$ for $d = 2, \ldots, 12$.

and therefore Theorem 3.6.2 holds with $\overline{\lambda}_d = C_d \cdot 2^{-(d+\frac{1}{2})}$ where $C_d := \sqrt{\frac{2}{V_d \cdot P_d}}$. With elementary geometry in $\mathbb{R}^2$, one can compute exactly $\overline{\lambda}_2 = \left(\left(\frac{16\pi}{3} + 2\sqrt{3}\right) \cdot \frac{4\pi + 3\sqrt{3}}{2\pi^2}\right)^{-\frac{1}{2}} = 0.2344+$. For $d > 2$, formulas involving regularized beta functions

$$I_x(a,b) = \frac{\int_0^z u^{a-1}(1-u)^{b-1}\, du}{\int_0^1 u^{a-1}(1-u)^{b-1}\, du}$$

are known for both V_d and P_d. Indeed,

$$\begin{aligned} V_d &= 2 - 2 \cdot \frac{\mathbf{vol}(\text{spherical cap of height } \frac{1}{2})}{\mathbf{vol}(B_1^d(0))} \\ &= 2 - I_{3/4}\left(\frac{d+1}{2}, \frac{1}{2}\right) \end{aligned} \tag{3.31}$$

and

$$P_d = 1 - \frac{d}{2} \int_0^1 x^{d-1} \cdot I_{1-\frac{x^2}{4}}\left(\frac{d+1}{2}, \frac{1}{2}\right)\, dx. \tag{3.32}$$

Note that $2^{-(d+\frac{1}{2})}$ is the value of $\overline{\lambda}$ obtained in [31], and that $C_d > 1$ for all d, so this is an improvement for all d. With some more work, one can show that $P_d \leq 1 - \Omega(\alpha^d)$ and $V_d \leq 2 - \Omega(\alpha^d)$, for some $\alpha \approx 0.865$, and therefore $C_d = 1 + 2^{-O(d)}$.

Chapter 4

Clique partitions of a graph and its complement

4.1 Introduction

For a graph G, let $\mathrm{cp}(G)$ denote its *(edge) clique partition number*, which is the minimum number of cliques in G needed to cover the edges of G exactly once. We call such a covering a clique partition of G. Similarly, we let $\mathrm{cc}(G)$ denote the clique *covering* number, which is the minimum number of cliques needed to cover the edges of G at least once. For each of these quantities, the algorithmic task of computing it for arbitrary graphs is an NP-complete problem [63] – nonetheless, they have been extensively studied from a combinatorial perspective. In this chapter, we concern ourselves primarily with the following question, raised in 1986 by De Caen, Erdős, Pullman and Wormald [18]: *what is the maximum possible value of* $\mathrm{cp}(G) + \mathrm{cp}(\overline{G})$, *where G is any graph on n vertices?* In the same paper, the authors solved the analogous problem for $\mathrm{cc}(G)$ up to $1 + o_n(1)$ factors, showing that

$$\max_{G\in\mathcal{G}_n} \mathrm{cc}(G) + \mathrm{cc}(\overline{G}) = \frac{n^2}{4}(1 + o(1)) \tag{4.1}$$

(where $\mathcal{G}_n$ is the set of all graphs on n vertices.) Note that the complete bipartite graph $K_{n/2,n/2}$ has $\mathrm{cc}(K_{\frac{n}{2},\frac{n}{2}}) = \lfloor \frac{n^2}{4} \rfloor$, which is the maximum value of $\mathrm{cc}(G)$ for $G \in$

$\mathcal{G}_n$ [22], and $\mathrm{cc}(K_{\frac{n}{2},\frac{n}{2}}) + \mathrm{cc}(\overline{K_{\frac{n}{2},\frac{n}{2}}}) = \lfloor \frac{n^2}{4} \rfloor + 2$, which almost matches the upper bound on $\mathrm{cc}(G) + \mathrm{cc}(\overline{G})$. L. Pyber [67] later closed the $o(n^2)$ gap, showing that $\max_{G \in \mathcal{G}_n} \mathrm{cc}(G) + \mathrm{cc}(\overline{G}) = \lfloor \frac{n^2}{4} \rfloor + 2$ for $n \geq n_0 = 2^{1500}$.

However, the corresponding problem for clique *partitions* appears to be much more difficult. In the same paper [18], De Caen et al. were able to show

$$\frac{7n^2}{25} + O(n) \leq \max_{G \in \mathcal{G}_n} \mathrm{cp}(G) + \mathrm{cp}(\overline{G}) \leq \frac{13n^2}{30} + O(n), \tag{4.2}$$

and left closing the rather sizeable gap as an open problem. They also conjectured that the lower bound was close to the truth, namely that

$$\max_{G \in \mathcal{G}_n} \mathrm{cp}(G) + \mathrm{cp}(\overline{G}) \sim \frac{7n^2}{25}. \tag{4.3}$$

The lower bound in (4.2) comes from a construction which is a type of blow-up of C_5, while the upper bound comes from Turan's theorem and Ramsey theory. Indeed, for $G \in \mathcal{G}_n$, we can think of G and $\overline{G}$ as the black and white edges, respectively, of a black/white coloring of the edges of K_n, and consider any maximal collection of black and white triangles in this coloring. Then the set of uncovered edges forms a K_6-free graph on n vertices, since otherwise the Ramsey bound $R(3,3) = 6$ would yield either another black or white triangle. By Turan's theorem, a K_6-free graph on n vertices has no more than $2n^2/5$ edges. Therefore, if t was the number of monochromatic triangles and s was the number of remaining edges, then since $s + 3t = \binom{n}{2}$, it follows that $\mathrm{cp}(G) + \mathrm{cp}(\overline{G}) \leq s + t \leq \frac{2n^2}{5} + \frac{n^2}{6} - \frac{2n^2}{15} = \frac{13n^2}{30}$.

As was remarked in [18], one can improve on the above argument by using known bounds on $R(k,k)$ and including larger cliques in the cover – however, such improvements quickly become negligible. Indeed, if one begins with a maximal collection of monochromatic copies of K_r, bounds the number of remaining edges by $\alpha_r n^2$, where $\alpha_r := \frac{1}{2} - \frac{1}{2R(r,r)-2}$, and then iterates on the uncovered edges with cliques of size K_{r-1}

down to K_2, it is not hard to see that the bound one obtains is

$$\operatorname{cp}(G) + \operatorname{cp}(\overline{G}) \leq \left(\alpha_3 + \frac{\alpha_4 - \alpha_3}{3} + \frac{\alpha_5 - \alpha_4}{6} + \cdots + \frac{\alpha_r - \alpha_{r-1}}{\binom{r-1}{2}} + \frac{\frac{1}{2} - \alpha_r}{\binom{r}{2}} \right) n^2 \quad (4.4)$$

Even using the most optimistic (i.e. smallest) of the possible values for $R(k,k)$ for $k \geq 5$, this approach will not yield an upper bound better than $0.41n^2$.

One reason why this bound is not tight is because the greedy packing – just throwing in monochromatic triangles, say, until there are no more remaining – can be a rather inefficient packing. In fact, while the above argument shows that a greedy packing of disjoint triangles into G and $\overline{G}$ always finds $\frac{n^2}{30}$ triangles, a subsequent work of Erdős et al. [23] shows by a different method that one can always find $\frac{3n^2}{55}$ such triangles. Picking $G = K_{\frac{n}{2},\frac{n}{2}}$ shows that the number of edge disjoint monochromatic triangles can be as few as $\frac{n^2}{12}$, and in [23] it is conjectured this is optimal. Building on the approach of Erdős et al. while making use of a *fractional* version of the packing problem, Keevash and Sudakov [47] give a computer-aided argument showing that the number of edge disjoint monochromatic triangles is always at least

$$\left(\frac{13}{196} + \frac{1}{84} - \frac{1}{1568} \right) n^2 + o(n^2) \approx \frac{n^2}{12.88} + o(n^2) \quad (4.5)$$

As noticed by Bujtas et al. [12], using this collection of triangles instead of one chosen greedily yields clique partitions for G and $\overline{G}$ with at most $0.34481n^2 + o(n^2)$ cliques, thereby improving the upper bound in (4.2) significantly.

4.1.1 New results

In this chapter, we shrink the gap in (4.2) slightly on both ends, showing that $\max_{G\in\mathcal{G}_n} \operatorname{cp}(G) + \operatorname{cp}(\overline{G}) \sim \alpha \cdot n^2$ for some $\alpha \in (0.28048, 0.3186)$, which improves upon the previously best known range of $[0.28, 0.34481)$. In particular, in Section 4.2 we modify the C_5 blow-up construction from [18] to yield an infinite family of graphs $G \in \mathcal{G}_n$ with

$$\operatorname{cp}(G) + \operatorname{cp}(\overline{G}) \geq \frac{23}{82}n^2 + O(n) = \left(\frac{7}{25} + \frac{1}{2050}\right) n^2 + O(n). \tag{4.6}$$

This implies that the conjectured asymptotic (4.3) is false, at least by a hair. On the other hand, in Section 4.3, we adapt the methods of Keevash and Sudakov [47] to directly attack the problem of upper bounding $\operatorname{cp}(G) + \operatorname{cp}(\overline{G})$ for all $G \in \mathcal{G}_n$. Running their algorithm on a 16-core compute grid, we bound the value of a certain linear program on all graphs with $n \leq 20$ vertices, and from this we extract the upper bound $\alpha < 0.3186$.

4.2 Improving the lower bound

The section closely follows [18] (with only one significant difference), but we include a complete argument for the reader's convenience. Before proceeding with the construction, we need the following lemma, which has appeared in many places but perhaps first in Pullman and Donald [66]. Recall that the edge chromatic number $\chi'(G)$ of a graph G is the minimum number of colors needed to color the edges of G so that no two edges of the same color are incident to the same vertex. We use the notation $G \equiv H$ to denote the graph on vertices $V(G) \sqcup V(H)$ formed by adding all edges between $V(G)$ and $V(H)$.

Lemma 4.2.1. *Let G be any graph with n vertices and e edges. Then $\operatorname{cp}(G \equiv \overline{K_\ell}) \geq n\ell - e$. If $\chi'(G) \leq \ell$, then $\operatorname{cp}(G \equiv \overline{K_\ell}) = n\ell - e$.*

Proof. Let $H = \overline{K_\ell}$ and let E_{G-H} be the set of all $n\ell$ edges between $V(G)$ and $V(H)$. Suppose $C_1, \ldots, C_r$ is a clique partition of $G \equiv H$. Since C_i can have at most one

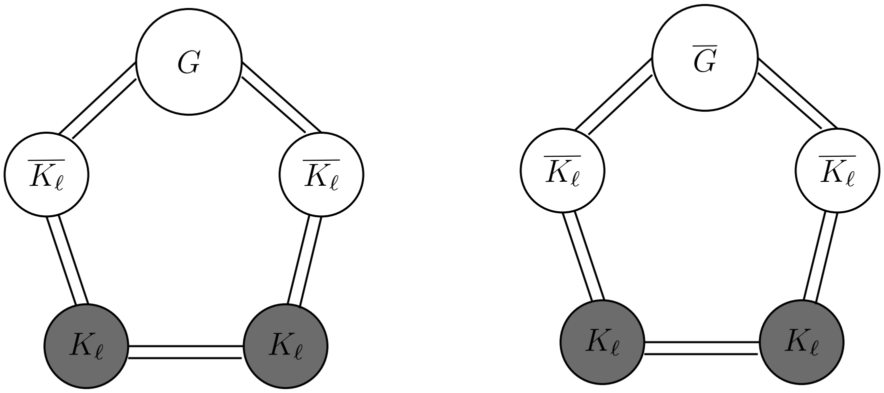

Figure 4-1: The graph $H_\ell(G)$ and its complement.

vertex in H, it follows that $|E(C_i) \cap E(G)| \geq \binom{|E(C_i)\cap E_{G-H}|-1}{2} \geq |E(C_i) \cap E_{G-H}| - 1$. Letting $S = \{i : E(C_i) \cap E_{G-H} \neq \emptyset\}$ and summing this inequality over S, we obtain

$$e \geq \sum_{i \in S} |E(C_i) \cap E(G)| \geq \sum_{i \in S} |E(C_i) \cap E_{G-H}| - |S| \geq n\ell - r \tag{4.7}$$

which implies $\mathrm{cp}(G \equiv H) \geq n\ell - e$. When $\chi'(G) \leq \ell$, we can assign each of the ℓ nodes in H to one of the ℓ color classes of a valid edge coloring in G, and obtain a collection of triangles of the form $\{v, x, y\}$, for $v \in H$ and $(x, y) \in E(G)$ which has been given color v in the edge coloring. No edge in E_{G-H} will be used twice precisely because no vertex in G is incident to two edges of the same color. This gives a collection of e edge-disjoint triangles which cover all the edges in G and leaves at most $n\ell - 2e$ edges left to cover, and adding in those edges yields a clique partition of size at most $n\ell - e$. □

We now describe a construction which generalizes the construction in [18]. Let ℓ and m be any positive integers, and let G be any graph on m vertices. We define $H_\ell = H_\ell(G)$ to be the graph in Figure 4-1, where the double lines are to be interpreted in the same way as the $\equiv$ symbol, i.e. including all possible edges between the vertices on either end. Observe that $\overline{H_\ell(G)} \cong H_\ell(\overline{G})$, and that the edges of $H_\ell(G)$ can be split into $X_\ell(G) := G \equiv \overline{K_{2\ell}}$ and $Y_\ell = \overline{K_\ell} \equiv K_\ell \equiv K_\ell \equiv \overline{K_\ell}$, as depicted in Figure 4-2. Clearly $\chi'(G) \leq \chi'(K_m)$, which is at most m, since we can assign the numbers $0, 1, \ldots, m-1$ to each vertex and color the edge (i, j) by $i - j \mod m$. So if $m \leq 2\ell$,

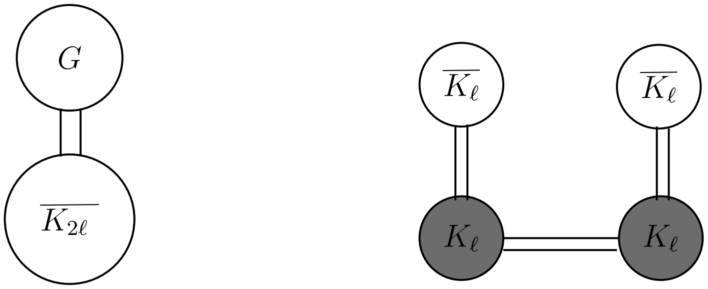

Figure 4-2: Decomposing H_ℓ into the edge-disjoint union of the two graphs $X_\ell(G) = G \equiv \overline{K_{2\ell}}$ (left) and $Y_\ell = \overline{K_\ell} \equiv K_\ell \equiv K_\ell \equiv \overline{K_\ell}$ (right).

Lemma 4.2.1 implies that $\text{cp}(X_\ell(G)) = m\ell - e(G)$. Therefore

$$\begin{aligned}\text{cp}(H_\ell(G)) + \text{cp}(\overline{H_\ell(G)}) &= \text{cp}(H_\ell(G)) + \text{cp}(H_\ell(\overline{G}))\\ &= \text{cp}(X_\ell(G)) + \text{cp}(X_\ell(\overline{G})) + 2\text{cp}(Y_\ell)\\ &= 2m\ell - \binom{m}{2} + 2\text{cp}(Y_\ell)\end{aligned}$$

for *any* graph G on $m \leq 2\ell$ vertices. (In fact, this still gives a lower bound on $\text{cp}(H_\ell(G)) + \text{cp}(\overline{H_\ell(G)})$ for any G and any m.) The term $\text{cp}(Y_\ell)$ was computed in [18], and we include the argument in the appendix.

Lemma 4.2.2 (Lem. 2 and 3 in [18]). *For any ℓ, $\text{cp}(Y_\ell) \geq \frac{7}{4}\ell^2 + O(\ell)$, and this is tight infinitely often.*

So for any G on m vertices, we have

$$\text{cp}(H_\ell(G)) + \text{cp}(\overline{H_\ell(G)}) \geq 2m\ell - \binom{m}{2} + \frac{7}{2}\ell^2 \tag{4.8}$$

Note that $H_\ell(G)$ has $n := m + 4\ell$ vertices, so maximizing (4.8) in m while keeping n fixed, we find that the optimum occurs at $m = \frac{9}{8}\ell$. At this value of m, the lower bound is $(8 - \frac{81}{128})\ell^2 + O(\ell)$ for a graph on $\frac{41}{8}\ell$ vertices, implying that, for infinitely many n,

$$\max_{G \in \mathcal{G}_n} \text{cp}(G) + \text{cp}(\overline{G}) \geq \frac{(8 - \frac{81}{128})}{(\frac{41}{8})^2} n^2 + O(n) = \frac{23}{82} n^2 + O(n).$$

4.3 Improving the upper bound

The problem of partitioning a graph G into as few cliques as possible is equivalent to the problem of *packing* disjoint copies of $K_3, K_4, \dots, K_n$ inside of G in such a way as to maximize a certain linear objective function. Indeed, given a clique partition C of G, let C_i denote the number of cliques of size i in C, for $i = 2, \dots, n$. Then

$$|C| = \sum_{i=2}^{n} C_i\,, \quad \text{and} \quad \sum_{i=2}^{n} \binom{i}{2} C_i = |E(G)| \tag{4.9}$$

and so

$$\begin{aligned} \mathrm{cp}(G) &= \min_C |C| \\ &= E(G) - \underbrace{\max_C \sum_{i \geq 3} \left(\binom{i}{2} - 1 \right) C_i}_{=:v(G)}\,. \end{aligned} \tag{4.10}$$

We will also consider *r-restricted clique packings/partitions*, in which the largest clique can have size at most r. We define $\mathrm{cp}(G, r)$ to be the minimum number of cliques of size at most r needed to partition the edges of G. Equivalently, $\mathrm{cp}(G, r) = E(G) - v_r(G)$, where

$$v_r(G) := \max_C \sum_{i=3}^{r} \left(\binom{i}{2} - 1 \right) C_i. \tag{4.11}$$

Clearly $\mathrm{cp}(G, r) \geq \mathrm{cp}(G)$, and one would expect the numbers $\mathrm{cp}(G, r)$ and $\mathrm{cp}(G)$ to be relatively close for large r. This is indeed the case, as we show in the following lemma.

Lemma 4.3.1. *For any $\epsilon > 0$, there exists an integer $r_0 = r_0(\epsilon)$ such that for any $r \geq r_0$ and any graph G on n vertices,*

$$\mathrm{cp}(G, r) \leq \mathrm{cp}(G) + \epsilon \cdot n^2.$$

Proof. We make use of the following fact, which is a straightforward consequence of Wilson's theorem [77]: for any fixed $t \geq 2$ and $\epsilon > 0$, there is an integer $m_0 = m_0(t, \epsilon)$

such that for all $m \geq m_0$, there is a partition of K_m into edge-disjoint copies of K_t and at most ϵm^2 left-over edges. Set $t = \frac{1}{2\epsilon}$ and $r_0 = m_0(t, \epsilon/5)$.

Let C be a clique partition with $|C| = \mathrm{cp}(G)$. For any $r \geq r_0$, we can obtain an r-restricted clique partition $\widetilde{C}$ from C as follows: keep each clique of size at most r, and for each clique K_m with $m > r$, decompose it into at most $\binom{m}{2}/\binom{t}{2}$ copies of K_t and cover the remaining edges (of which there are at most $\frac{\epsilon}{5} \cdot m^2$) with K_2's. This gives a clique partition $\widetilde{C}$ of size

$$\begin{aligned} |\widetilde{C}| &\leq \sum_{i=2}^{r} C_i + \sum_{i>r} \left(\frac{\binom{i}{2}}{\binom{t}{2}} + \frac{\epsilon}{5} i^2 \right) C_i \\ &\leq \sum_{i=2}^{n} C_i + \epsilon \cdot \sum_{i=2}^{n} \binom{i}{2} C_i \\ &= |C| + \epsilon \cdot |E(G)| \end{aligned}$$

from which the lemma follows. □

4.3.1 Fractional clique packings

For a fixed family $\mathcal{F}$ of graphs and any graph G, let $\binom{G}{\mathcal{F}}$ denote the set of (unlabeled, non-induced) subgraphs of G which are isomorphic to some $F \in \mathcal{F}$. Following Keevash and Sudakov [47], and Yuster [80], we say a function $\psi : \binom{G}{\mathcal{F}} \to [0, 1]$ is a *fractional $\mathcal{F}$-packing of G* if for every edge $e \in E(G)$, we have

$$\sum_{e \in H \in \binom{G}{\mathcal{F}}} \psi(H) \leq 1.$$

We denote by $G_{\mathcal{F}}$ the polyhedron of all fractional $\mathcal{F}$-packings of G. As we are interested in the fractional analogue of clique packings, we will only be concerned with families of the form

$$\mathcal{F}_r := \{K_3, K_4, \ldots, K_r\}.$$

Let $\nu_r(G)$ be the value of the linear program

$$\max_{\psi \in G_{\mathcal{F}_r}} \sum_{H \in \binom{G}{\mathcal{F}_r}} \left(\binom{|H|}{2} - 1 \right) \psi(H) \tag{4.12}$$

When the objective function is simply $\sum_{H \in \binom{G}{\mathcal{F}}} \psi(H)$, and the family $\mathcal{F} = \{F\}$ is just a single graph, a theorem of Haxell and Rödl [36] implies that relaxing the domain of maximization from (integer) packings to fractional packings can only change the value of the optimum by $o(n^2)$. Subsequently, Yuster [80] extended this result to arbitrary families of graphs. For finite families (such as $\mathcal{F}_r$), Yuster's proof easily extends to arbitrary linear objective functions [79]. Therefore:

Theorem 4.3.1. *For any $r \geq 3$ and $G \in \mathcal{G}_n$,*

$$\upsilon_r(G) - \nu_r(G) = o(n^2).$$

The advantages of studying fractional clique packings rather than clique partitions are twofold. Firstly, solving the linear program (4.12) is computationally feasible, unlike the corresponding integer program. Secondly, they can be *averaged*, which not only enables one to turn finite computations into asymptotic bounds, but also allows one to leverage the results of a search on n vertices to reduce the search space when looking for a minimizer on $n+1$ vertices. This is the approach used by Keevash and Sudakov in [47], and the following averaging lemma (for a different LP) appears as their Lemma 2.1, with the same proof.

For each r, define

$$f_r(n) := \min_{G \in \mathcal{G}_n} \nu_r(G) + \nu_r(\overline{G}).$$

Lemma 4.3.2. *For any $r \geq 3$, the sequence $\frac{f_r(n)}{n(n-1)}$ is increasing in n.*

Proof. Let $G \in \mathcal{G}_{n+1}$, and let $G_1, \ldots, G_{n+1}$ be the induced subgraphs on the vertex subsets of size n. Let $\psi_i, \overline{\psi_i}$ be optimal fractional packings on G_i and $\overline{G_i}$. Since each edge of G (and $\overline{G}$) occurs in $n-1$ of the G_i, we have that

$$\psi := \frac{1}{n-1}\sum_{i=1}^{n+1}\psi_i, \qquad \overline{\psi} := \frac{1}{n-1}\sum_{i=1}^{n+1}\overline{\psi_i}$$

are fractional packings on G and $\overline{G}$ with combined objective value at least $\frac{n+1}{n-1}f_r(n)$, and hence $\frac{f_r(n+1)}{(n+1)n} \geq \frac{(n+1)f_r(n)}{n(n-1)(n+1)} = \frac{f_r(n)}{n(n-1)}$, as claimed. □

Since the sequence $\frac{f_r(n)}{n(n-1)}$ is obviously bounded above by $1/2$, it follows that is converges to a limit $c_r \in (0, 1/2)$. Since c_r is increasing in r, the sequence $\{c_r\}$ also converges to a limit we'll call c_∞.

Theorem 4.3.2.

$$\max_{G\in\mathcal{G}_n} \operatorname{cp}(G) + \operatorname{cp}(\overline{G}) \sim \left(\frac{1}{2} - c_\infty\right) n^2$$

Proof. This essentially follows from Lemma 4.3.1 and Theorem 4.3.1. More explicitly, for any $\epsilon > 0$, we can pick r large enough so that $|\operatorname{cp}(G) - \operatorname{cp}(G, r)| < \epsilon n^2$ for any $G \in \mathcal{G}_n$, and $|c_r - c_\infty| < \epsilon$. Now pick n large enough so that $|v_r(G) - \nu_r(G)| < \epsilon n^2$ for any $G \in \mathcal{G}_n$ and $|f_r(n) - c_r n^2| < \epsilon n^2$. It follows that

$$\max_{G\in\mathcal{G}_n} \operatorname{cp}(G) + \operatorname{cp}(\overline{G}) \in \left(\frac{1}{2} - c_\infty \pm 8\epsilon\right) n^2$$

for n sufficiently large. □

The same argument shows that $\max_{G\in\mathcal{G}_n} \operatorname{cp}(G, r) + \operatorname{cp}(\overline{G}, r) \sim \left(\frac{1}{2} - c_r\right) n^2$. Let us define $\alpha_r := \frac{1}{2} - c_r$, and $\alpha_\infty = \frac{1}{2} - c_\infty$. We seek an upper bound on α_∞, and since $\alpha_\infty \leq \alpha_r = \frac{1}{2} - c_r \leq \frac{1}{2} - \frac{f_r(n)}{n(n-1)}$ for any n, it suffices for our purposes to compute a lower bound on the value of $\frac{f_r(n)}{n(n-1)}$ for any particular pair of positive integers (r, n). For example, it takes only a few minutes on a modern computer to compute $f_4(8) = 6$ numerically by solving the LP (4.12) on every non-isomorphic graph on 8 vertices. This shows that $\alpha_\infty \leq \alpha_4 \leq \frac{1}{2} - \frac{6}{8\cdot 7} = \frac{11}{28} \approx 0.3928$. This already beats the best bound one can get from purely Ramsey-based arguments, although it does not beat the Keevash-Sudakov triangle packing bound. In the remainder of this chapter, we improve this bound in two ways: first, we show in Section 4.3.2, we can *combine*

Ramsey-type arguments with estimates on $f_r(n_0)$ to yield better estimates on $f_r(n)$ for n much larger than n_0; second, in Section 4.3.3 we compute the exact value of $f_4(n)$ up to $n = 19$, using an algorithm of Keevash and Sudakov which is significantly more efficient than brute force search.

4.3.2 Ramsey-type improvements

In [47], it was observed that the averaging argument in Lemma 4.3.2 can be improved, in a sense, by using a different decomposition of G into smaller subgraphs based on a greedy packing as described in the introduction. In particular, given any bicoloring of K_{3n}, greedily select vertex-disjoint monochromatic triangles $T_1, \ldots, T_i$, where the fact that $R(3,3) = 6$ guarantees that we can do this until 3 vertices remain – giving us $n-1$ triangles $T_1, \ldots, T_{n-1}$, and one set of 3 vertices denoted T_n. Consider the 3^n colorings c of K_n obtained by picking one vertex in each T_i and the edges between them. Each coloring has some fractional packing ψ_c of weight at least $f_3(n)$, and since each edge between T_i and T_j for $i \neq j$ occurs in exactly 3^{n-2} of these, the average $3^{-(n-2)} \sum_c \psi_c$ is a valid fractional packing in K_{3n} of weight at least $9f_3(n)$. Since each of the monochromatic triangles $T_1, \ldots, T_{n-1}$ are edge disjoint from this packing, they can be included as well, yielding a lower bound

$$f_3(3n) \geq 9f_3(n) + 2(n-1). \tag{4.13}$$

Since $R(4,4) = 18$, we can greedily find vertex disjoint monochromatic copies of K_4, $H_1, \ldots, H_{n-4}$, with 16 vertices remaining. From the remaining vertices, we can find edge disjoint monochromatic triangles $T_{n-3}, T_{n-2}, T_{n-1}, T_n$, which we join with the remaining four vertices to form $H_{n-3}, \ldots, H_n$, each of size four. Repeating the same process as above, we see that

$$f_4(4n) \geq 16f_4(n) + 5(n-4) + 8. \tag{4.14}$$

For $r = 5$, we can use the bound $R(5,5) \leq 48$ to find $n - 9$ vertex-disjoint copies of K_5, with 45 vertices left over. We can then find $\lceil (45 - 18)/4 \rceil = 7$ copies of K_4, with 17 vertices left over, in which we can find 2 monochromatic triangles, and distribute the remaining vertices so that each of these 11 parts has size 5. Arguing as above, this then implies

$$f_5(5n) \geq 25 f_5(n) + 9(n - 9) + 37. \tag{4.15}$$

We omit the details, but using similar arguments and the Ramsey number bounds $R(6,6) \leq 165$ and $R(7,7) \leq 540$ yields the inequalities

$$f_6(6n) \geq 36 f_6(n) + 14n - 151 \tag{4.16}$$

$$f_7(7n) \geq 49 f_7(n) + 20n - 532. \tag{4.17}$$

According to András Gyárfás [33], Paul Erdős, sitting in the Atlanta Airport in 1995, asked his companions whether every bicoloring of the edges of $K_{R(k,k)}$ contains *two edge-disjoint monochromatic copies of* K_k. Ralph Faudree pointed out that this is not true, at which point Erdős asked for the smallest number $n(k)$ for which any bicoloring of $K_{n(k)}$ *does* contain two edge-disjoint monochromatic K_k's. The next day, Faudree showed $n(3) = 7$, and some time later, Gyárfás showed $n(4) = 19$. For our purposes, however, we require *vertex-disjoint* monochromatic copies of K_r. In the appendix we give an argument, inspired by the proof of $n(4) = 19$ by Gyárfás, showing that $n = 20$ is sufficient to find two vertex-disjoint monochromatic K_4's, provided there is also a monochromatic K_5:

Lemma 4.3.3. *Any bicoloring of the edges of K_{20} with a monochromatic copy of K_5 contains two vertex-disjoint monochromatic copies of K_4.*

With this fact in hand, we can obtain a slight improvement over (4.14):

Lemma 4.3.4. *For any $n \geq 12$, $f_4(4n) \geq 16 f_4(n) + 5n - 9$.*

Proof. Consider any bicoloring of K_{4n}. Since $4n \geq 48 \geq R(5,5)$, there is some monochromatic copy of K_5 – call this subgraph N. While there are at least $R(4,4) =$

18 vertices in $K_{4n} \setminus N$, we can greedily select vertex-disjoint monochromatic copies of K_4 in $K_{4n} \setminus N$, $H_1, \ldots, H_{n-5}$. This leaves a set S of 15 remaining vertices. By Lemma 4.3.3, the coloring induced on $S \cup N \cong K_{20}$ has two vertex disjoint copies of K_4, which we call H_{n-4} and H_{n-3}. Removing the vertices in $H_{n-4} \cup H_{n-3}$ from $N \cup S$, we are left with twelve vertices, which must contain 3 vertex-disjoint monochromatic triangles T_1, T_2 and T_3. This leaves behind a set of three vertices $\{v_1, v_2, v_3\}$. Decomposing K_{4n} into the n blocks of size 4

$$H_1, \ldots, H_{n-3}, T_1 \cup \{v_1\}, T_2 \cup \{v_2\}, T_3 \cup \{v_3\}$$

we consider the 4^n edge-colorings c of K_n obtained by picking one vertex from each part. Each of these has a fractional clique packing ψ_c of size at least $f_4(n)$, and since each edge is used in $4^{-(n-2)}$ such ψ_c, we know that $4^{-(n-2)} \sum_c \psi_c$ is a valid packing in K_{4n}. Adding in the copies of K_4 and K_3 inside the parts, we see that $f_4(4n) \geq 16 f_4(n) + 5(n-3) + 6$. □

4.3.3 Computer-aided calculations

We next describe (a generalization of) the algorithm used by Keevash and Sudakov in the case of triangle packings [47], which we call the KS bootstrap method. For any finite family of graphs $\mathcal{F} = \{H_1, \ldots, H_r\}$, any graph $G \in \mathcal{G}_n$, and any vector $\Gamma \in \mathbb{R}^{\mathcal{F}}$, we let $\nu_{\mathcal{F},\Gamma}(G)$ be the value of the linear program

$$\max_{\psi \in G_{\mathcal{F}}} \sum_{H \in \binom{G}{\mathcal{F}}} \Gamma(H)\psi(H) \tag{4.18}$$

For any $\ell \in \mathbb{R}$, and any set L of graphs, define

$$\mathcal{L}(L, \ell) := \{G \in L \, : \, \nu_{\mathcal{F},\Gamma}(G) + \nu_{\mathcal{F},\Gamma}(\overline{G}) \leq \ell\},$$

and let $\Lambda_{\mathcal{F},\Gamma}(L) = \min_{G \in L} \nu_{\mathcal{F},\Gamma}(G') + \nu_{\mathcal{F},\Gamma}(\overline{G'})$. We also define $\mathsf{ext}_1(L)$ be the set of *one-vertex extensions* of the graphs in L. The KS bootstrap method is based on the following observation: by Lemma 4.3.2, any graph $G' \in \mathcal{G}_{n+1}$ with $\nu_{\mathcal{F},\Gamma}(G') + \nu_{\mathcal{F},\Gamma}(\overline{G'}) \leq \frac{n+1}{n-1} \cdot \ell$ must be a one-vertex extension of some graph in $\mathcal{L}(\mathcal{G}_n, \ell)$. In other words, if $\{\ell_n\}_{n\in\mathbb{N}}$ is any sequence of numbers satisfying $\ell_{n+1} \geq \frac{n+1}{n-1}\ell_n$, then

$$\mathcal{L}(\mathcal{G}_{n+1}, \ell_{n+1}) \subseteq \mathsf{ext}_1(\mathcal{L}(\mathcal{G}_n, \ell_n)).$$

Let us refer to such sequences ℓ_n as *level sequences*.

Algorithm 6: KS Bootstrap Method

```
n ← $n_0$
compute $L = \mathcal{L}(\mathcal{G}_n, \ell_n)$ (e.g. via exhaustive search)
while $L \neq \emptyset$ do
    $\Lambda[n] = \Lambda_{\mathcal{F},\Gamma}(L)$
    $S \leftarrow \mathsf{ext}_1(L)$
    $L \leftarrow \mathcal{L}(S, \ell_{n+1})$
    $\Lambda[n+1] = \ell_{n+1}$
    $n \leftarrow n + 1$
end
return $\Lambda$
```

Note that the sequence ℓ_n used by Algorithm 6 does not have to be determined before runtime – as long as it is guaranteed to be a level sequence, this guarantees the loop invariant $\mathcal{L}(\mathcal{G}_n, \ell_n) \subseteq L$, and hence $\Lambda[n] \leq \Lambda_{\mathcal{F},\Gamma}(\mathcal{G}_n)$. In [47], they choose a parameter d (called the "search depth"), and define ℓ_n recursively by taking $\ell_{n_0} = +\infty$ and ℓ_{n+1} to be $\frac{n+1}{n-1} \cdot \alpha_n$, where α_n is either (a) the dth smallest value in the set $\{\nu_{\mathcal{F},\Gamma}(G') + \nu_{\mathcal{F},\Gamma}(\overline{G'}) : G \in \mathcal{L}(\mathcal{G}_n, \ell_n)\}$, if this set has at least d elements, or (b) ℓ_n, if the set has fewer than d elements. The role of d is to limit the number of graphs stored in the set L – if $d = \infty$, then Algorithm 6 has to solve the LP (4.18) on every graph up to size n in order to compute $\Lambda_{\mathcal{F},\Gamma}(\mathcal{G}_n)$, while if d is too small, then the while loop will terminate after a small number of iterations. We ran an implementation[1] of this method on a 16-core computing grid with $d = 11$, starting with an exhaustive

[1] There are other implementation details omitted from our pseudocode description of Algorithm 6 which also have significant impact its runtime and memory usage, such as how and when to check and prune isomorphisms, which LP solver to use, which value of n_0 to exhaust from, and how to split work among multiple processors, if run in parallel. Our implementation is very similar to the one used in [47] – we recommend reading their magma code which can be found online at https://people.math.ethz.ch/~sudakovb/triangles-program.

n i	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	2	4	6	8	11	15	19	23	27	33	39	45	51	57	>64.725
2	4	5	7	9	12	16	20	24	28	34	40	46	52	58	*
3	5	6	8	10	12.5	16.5	20.5	24.5	28.5	34.5	40.5	*	*	*	*
4	6	7	9	11	13	17	21	25	29	34.75	40.75	*	*	*	*
5	7	8	9.5	12	14	17.5	21.25	25.25	29.25	35	*	*	*	*	*
6	8	$8.\overline{3}$	10	$12.\overline{3}$	14.5	18	21.5	25.5	29.5	35.25	*	*	*	*	*
7	$8.\overline{3}$	8.5	$10.\overline{3}$	12.5	$14.\overline{6}$	18.25	22	26	30	35.5	*	*	*	*	*
8	9	9	10.5	12.6	14.8	$18.\overline{3}$	22.25	26.25	30.25	*	*	*	*	*	*
9	$10.\overline{6}$	9.5	10.6	$12.\overline{6}$	15	18.5	$22.\overline{3}$	$26.\overline{3}$	30.5	*	*	*	*	*	*
10	12.5	10	$10.\overline{6}$	12.8	15.5	$18.\overline{6}$	22.5	26.5	30.75	*	*	*	*	*	*
11	*	$10.\overline{3}$	10.8	13	15.6	18.75	*	*	31	*	*	*	*	*	*

Table 4.1: The lowest values of $\nu_4(G) + \nu_4(\overline{G})$ for $G \in \mathcal{G}_n$, $n = 6, \ldots, 19$, as found by the KS bootstrap method. The level ℓ_{20} was 64.72527+ when the algorithm terminated, which implies that $f_4(20) > 64.725$.

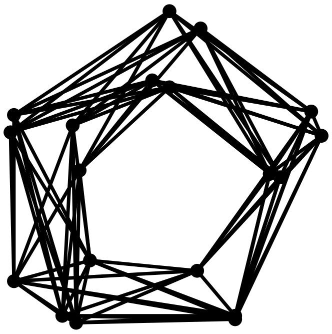

Figure 4-3: One of the graphs with the smallest value of $\nu_4(G)+\nu_4(\overline{G})$ on 19 vertices.

search on $n_0 = 6$ vertices, and obtained the results summarized in Table 4.1. The last column in particular implies $f_4(20) > 64.725$, which implies $c_4 > 0.1703$. Using Lemma 4.3.4, and inequalities (4.15), (4.16), and (4.17) (in that order), we can obtain the bound $c_7 \geq 0.1814$, which implies

$$\max_{G \in \mathcal{G}_n} \operatorname{cp}(G) + \operatorname{cp}(\overline{G}) < 0.3186n^2 + o(n^2). \tag{4.19}$$

$$\begin{matrix}
0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&1&1&1&1\\
0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&1&1&1&1\\
0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&1&1&1&1\\
0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&1&1&1&1\\
1&1&1&1&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0\\
1&1&1&1&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0\\
1&1&1&1&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0\\
1&1&1&1&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1\\
0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1\\
0&0&0&0&0&0&0&0&1&1&0&1&1&1&1&1&1&1&1\\
0&0&0&0&1&1&1&1&1&1&1&0&1&1&1&0&0&0&0\\
0&0&0&0&1&1&1&1&1&1&1&1&0&1&1&0&0&0&0\\
0&0&0&0&1&1&1&1&1&1&1&1&1&0&1&0&0&0&0\\
0&0&0&0&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0\\
1&1&1&1&0&0&0&0&1&1&1&0&0&0&0&0&1&1&1\\
1&1&1&1&0&0&0&0&1&1&1&0&0&0&0&1&0&1&1\\
1&1&1&1&0&0&0&0&1&1&1&0&0&0&0&1&1&0&1\\
1&1&1&1&0&0&0&0&1&1&1&0&0&0&0&1&1&1&0
\end{matrix}$$

Figure 4-4: The corresponding adjacency matrix.

4.4 Related questions

- What is c_4? An upper bound comes from $G = K_{n/2,n/2}$, which has $\mathrm{cp}_4(G) + \mathrm{cp}_4(\overline{G}) = \frac{7n^2}{24}$, implying that $c_4 \leq \frac{5}{24} = 0.208333$ – is this tight?

- Let $\mathcal{G}_n^\alpha = \{G \in \mathcal{G}_n : |E(G)| = \alpha \cdot \binom{n}{2}\}$. Then what is

  $$f(\alpha) := \lim_{n\to\infty} n^{-2} \cdot \max_{G\in\mathcal{G}_n^\alpha} \mathrm{cp}(G) + \mathrm{cp}(\overline{G})?$$

  Is $\mathrm{argmax}_\alpha f(n, \alpha) = \frac{1}{2}$? Is $\max_{G\in\mathcal{G}_n} \mathrm{cp}(G) + \mathrm{cp}(\overline{G})$ always attained by a graph G with $G \cong \overline{G}$? One can ask the same question of the fractional version.

- A related question would be to compute $\max_{G\in\mathcal{G}_n} \min\{\mathrm{cp}(G), \mathrm{cp}(\overline{G})\}$. Using the construction $H_\ell(G)$ from Section 4.3, and taking G to be a self-complementary graph on $\frac{9\ell}{8}$ vertices, we see that

  $$0.1402 \approx \frac{23}{162} \leq \max_{G\in\mathcal{G}_n} \min\{\mathrm{cp}(G), \mathrm{cp}(\overline{G})\} \leq \frac{1}{4}.$$

4.5 Appendix: Missing proofs

Lemma 4.2.2: *For any ℓ, $\mathrm{cp}(Y_\ell) \geq \frac{7}{4}\ell^2 + O(\ell)$, and this is tight infinitely often.*

Proof. Let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be an optimal clique partition of Y_ℓ. Let us denote the left (according to Figure 4-2) copy of K_ℓ in Y_ℓ by A and the right copy by B. Suppose that $\mathcal{C}' = \{C_1, \ldots, C_t\}$, for some $t \leq k$, is the sub-collection of cliques which contain vertices in both A and B. Let E_A and E_B be the edges in $A \cap \mathcal{C}'$ and $B \cap \mathcal{C}'$, so that Y_ℓ is the edge disjoint union of $(A \setminus E_A) \equiv \overline{K}_\ell$ and $(B \setminus E_B) \equiv \overline{K}_\ell$ with $\mathcal{C}'$, and therefore

$$\mathrm{cp}(Y_\ell) \geq 2\ell^2 - 2\binom{\ell}{2} + |E_A| + |E_B| + t \tag{4.20}$$

If clique C_i has a_i vertices in A and b_i vertices in B, then

$$\sum_i^t a_i b_i = \ell^2 \tag{4.21}$$

and

$$|E_A| + |E_B| + t = \sum_{i=1}^t \left(\binom{a_i}{2} + \binom{b_i}{2} + 1\right). \tag{4.22}$$

Minimizing (4.22) over positive integers a_i, b_i subject to the constraint (4.21), we see the minimum occurs when $a_i = b_i = 2$, i.e. each $C_i \in \mathcal{C}'$ is a K_4 with two vertices in each of A and B. □

Lemma 4.3.3: *Any bicoloring of the edges of K_{20} with a monochromatic copy of K_5 contains two vertex-disjoint monochromatic copies of K_4.*

Proof. Suppose that we have a bicoloring of K_{20} with a red copy $N = \{n_1, ..., n_5\}$ of K_5. If there is a blue copy of K_4, then we are finished, because this blue copy and N cannot share an edge, and therefore share at most one vertex. We may now assume that all monochromatic copies of K_4 are red.

We can address the case in which there exists a vertex v such that it is incident to at least nine red and blue edges each relatively quickly. We denote by R and B the cliques on the red and blue neighbors of v, respectively. Because the Ramsey number

$R(3,4) = 9$ and our graph has no blue copy of K_4, R must contain a red copy of K_3 and B must contain a red copy of K_4. Adding v to the red copy of K_3 in R results in two red copies of K_4, one in $R \cup v$ and one in N. We may now assume that all vertices have at least eleven incident edges of the same color.

Consider the case in which some vertex v has two red and two blue edges adjacent to a red copy M of K_4. If v has at least eleven red edges, then it has at least nine red edges connected to $K_{19} \backslash M$, which, by the same argument above, implies $K_{20} \backslash M$ has a red copy of K_4. The same argument holds if v has at least eleven blue edges. We may now assume that no vertex has two red and two blue edges adjacent to a red copy of K_4.

From here, we consider two cases:

Case I: Suppose that there exists five vertices $V = \{v_1, ..., v_5\} \subset K_{20} \backslash N$, each with at least three red edges adjacent to N. Because no vertex has both two red and two blue edges adjacent to a red copy of K_4, each vertex of V has at least four red edges adjacent to N. In addition, because our graph has no blue copy of K_4 every set $V \backslash v_i$ has a red edge. If one vertex, say v_1, has five edges adjacent to N, then, assuming (wlog) that (v_2, v_3) is a red edge in $V \backslash v_1$ and that the at most two blue edges between v_2, v_3 and N are not incident to n_4 or n_5, the subsets $\{v_1, n_1, n_2, n_3\}$ and $\{v_2, v_3, n_4, n_5\}$ are both red copies of K_4. We may now assume that each vertex in V has exactly four red edges adjacent to N. Let $f(v_i)$ denote the unique vertex in N for which $(v_i, f(v_i))$ is blue, and $f(V)$ denote the range of f. We consider multiple cases, depending on the size of $|f(V)|$.

Suppose $|f(V)| > 2$. Let (v_1, v_2) be a red edge in V, and let us denote by n_1, n_2, n_3 three edges in N for which (v_i, n_j), $i = 1, 2$, $j = 1, 2, 3$, are all red edges. By assumption, $|f(V) \cap \{n_1, n_2, n_3\}| > 0$, so (wlog) suppose that $f(v_3) = n_1$. Then $\{v_1, v_2, n_1, n_2\}$ and $\{v_3, n_3, n_4, n_5\}$ are both red copies of K_4.

Suppose $|f(V)| = 2$. Without loss of generality, let $f(V) = \{n_1, n_2\}$ and $|f^{-1}(n_1)| \geq 3$. Because there are no blue copies of K_4 in our graph, $f^{-1}(n_1)$

contains a red edge (v_i, v_j), and the subsets $\{v_i, v_j, n_2, n_3\}$ and $\{v_k, n_1, n_4, n_5\}$ are both red copies of K_4, where $v_k \in f^{-1}(n_2)$.

Suppose $|f(V)| = 1$. Without loss of generality, let $f(V) = n_1$. Then V does not contain a blue copy of K_3, otherwise our graph would contain a blue copy of K_4. If V contains a red copy $\{v_1, v_2, v_3\}$ of K_3, then $\{v_1, v_2, v_3, n_2\}$ and $\{v_4, n_3, n_4, n_5\}$ are two red copies of K_4, and we are done. If V does not contain a red or blue copy of K_3, then the red edges in V forms a cycle of length five, and there are two vertex-disjoint red edges in V, denoted (v_i, v_j) and (v_k, v_l). In this case, the subsets $\{v_i, v_j, n_2, n_3\}$ and $\{v_k, v_l, n_4, n_5\}$ are both red copies of K_4.

Case II: Suppose that there exists at most four vertices in $K_{20} \backslash N$ with at least three red edges adjacent to N. Then there are at least eleven vertices in $K_{20} \backslash N$ with at least three blue edges adjacent to N. Because no vertex has two red and two edges adjacent to a red copy of K_4, these vertices have at least four blue edges adjacent to N, and so there exists a vertex $n_i \in N$ with at least nine blue edges adjacent to $K_{20} \backslash N$. Therefore, $K_{20} \backslash N$ must contain a red copy of K_4.

□

Bibliography

[1] M. Ajtai and N. Linial. The influence of large coalitions. *Combinatorica*, 13(2):129-145, 1993.

[2] N. Alon. Neighborly families of boxes and bipartite coverings. *Algorithms and Combinatorics*, Volume 14, 1997, pages 27–31.

[3] Andris Ambainis, Mohammad Bavarian, Yihan Gao, Jieming Mao, Xiaoming Sun, and Song Zuo. *Tighter relations between sensitivity and other complexity measures.* In Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I, pages 101–113, 2014.

[4] A. Ambainis, K. Prusis, and J. Vihrovs. *Sensitivity versus certificate complexity of Boolean functions.* In Proceedings of the 11th International Computer Science Symposium in Russia, volume 9691 of Lecture Notes in Computer Science, pages 16–28. Springer International Publishing, 2016.

[5] A. Ambainis and X. Sun. New separation between $s(f)$ and $\mathrm{bs}(f)$. Electronic Colloquium on Computational Complexity (ECCC), 18:116, 2011.

[6] S. Ben-David, P. Hatami and A. Tal. Low-sensitivity functions from unambiguous certificates. In Conference on Innovations in Theoretical Computer Science (ITCS 2017), 2017.

[7] Michael Ben-Or and Nathan Linial. Collective coin flipping, robust voting schemes and minima of banzhaf values. In 26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985, pages 408–416, 1985.

[8] M. Blum and R. Impagliazzo, Generic oracles and oracle classes, 28th FOCS, 1987.

[9] Bela Bollobas. Area of the Union of Disks. *Elem. Math* 23, 61–62, 1968.

[10] A. Borodin and J. Hopcroft, Routing and merging on parallel models of computation, in Procedings, 14th ACM Symp. on Theory of Computing, 1982, pp. 338-344

[11] Harry Buhrman and Ronald De Wolf, *Complexity measures and decision tree complexity: a survey*, Theoretical Computer Science 288 (2002), no. 1, 21–43.

[12] C. Bujtas, A. Davoodi, E. Győri, Z. Tuza. Clique Coverings and Claw-free Graphs, 2016.

[13] J. Chiarelli, P. Hatami and M. Saks. An Asymptotically Tight Bound on the Number of Relevant Variables in a Bounded Degree Boolean function. Combinatorica 40, 237–244 (2020).

[14] S.M. Cioabă, M. Tait. Variations on a theme of Graham and Pollak. *Discrete Mathematics*, Volume 313, Issue 5, 2013, pages 665–676.

[15] Henry Cohn, Robin Pemantle, and James G. Propp. Generating a random sink-free orientation in quadratic time. *Electr. J. Comb.*, 9(1), 2002.

[16] S. Cook and C. Dwork, Bounds on the time for parallel RAM's to compute simple functions, in Proceedings, 14th ACM Symp. on Theory of Computing, 1982, pp. 231-233.

[17] B. Csikós, On the volume of the union of balls, Discrete Comput. Geom. 20 (1998), no. 4, 449–461, 2008.

[18] D. de Caen, P. Erdős, N. J. Pullmann, and N. C. Wormald. Extremal clique coverings of complementary graphs. *Combinatorica*, Volume 6 Issue 4, 1986, pages 309–314.

[19] D. de Caen, D.A. Gregory, and D. Pritikin. Minimum biclique partitions of the complete multigraph and related designs. *Graphs, matrices, and designs*, 1993, pages 93–119.

[20] A. Denise, O. Rocques, M. Termier. Random generation of words of context-free languages according to the frequencies of letters. In: Colloquium on Mathematics and Computer Science, pp. 113–125 (2000)

[21] A. Denise, P. Zimmermann. Uniform random generation of decomposable structures using floating-point arithmetic. Theor. Comp. Sci. 218(2), 233–248 (1999)

[22] P. Erdős, A. W. Goodman, L. Pisa. The Representation of a Graph by Set Intersections, Can. J. Math. 18 (1955), 105–112

[23] P. Erdős, R. J. Faudree, R. J. Gould, M. S. Jacobson, and J. Lehel. Edge disjoint monochromatic triangles in 2-colored graphs, *Discrete Math* Volume 231, 2001, pages 135–141.

[24] P. Erdős. Some recent problems and results in graph theory. *Discrete Math*, Volume 164, 1997, pages 81–85.

[25] Yuval Filmus, Hamed Hatami, Steven Heilman, Elchanan Mossel, Ryan O'Donnell, Sushant Sachdeva, Andrew Wan, and Karl Wimmer. *Real Analysis in Computer Science: A collection of open problems,* Simons Institute, 2014

[26] M. Göös, T. Pitassi, and T. Watson. Deterministic communication vs. partition number. Electronic Colloquium on Computational Complexity (ECCC) TR15-050, 2015

[27] J. Gilmer, M. Saks, and S. Srinivasan. Composition limits and separating examples for some Boolean function complexity measures. In Proceedings of 2013 IEEE Conference on Computational Complexity (CCC 2013), pages 185–196, 2013.

[28] C. Gotsman and N. Linial,The equivalence of two problems on the cube. J. Comb. Theory, Ser. A 61, 1 (1992), 142-146

[29] R.L. Graham and H.O. Pollak. On the addressing problem for loop switching. *Bell System Technical J.*, Volume 50, Issue 8, 1971, pages 2495–2519.

[30] Heng Guo and Mark Jerrum. A polynomial-time approximation algorithm for all-terminal network reliability. SIAM J. Comput., 48(3):964–978, 2019.

[31] Heng Guo, Mark Jerrum. Perfect Simulation of the Hard Disks Model by Partial Rejection Sampling. In ICALP, volume 107 of LIPIcs, pages 69:1–69:10, 2018.

[32] Heng Guo, Mark Jerrum, and Jingcheng Liu. Uniform sampling through the Lovasz local lemma. In *STOC*, 342–355, 2017.

[33] A. Gyárfás. Fruit Salad. Electronic Journal of Combinatorics, Volume 4, 1997.

[34] Bernhard Haeupler, Barna Saha, and Aravind Srinivasan. New constructive aspects of the Lovasz Local Lemma. *J. ACM*, 58(6):28:1, 2011

[35] Pooya Hatami, Raghav Kulkarni, and Denis Pankratov, *Variations on the sensitivity conjecture*, Theory of Computing Library, Graduate Surveys 4 (2011), 1–27.

[36] P. Haxell and V. Rődl. Integer and fractional packings in dense graphs, *Combinatorica* Volume 21, 2001, pages 13–38.

[37] Kun He, Qian Li, Xiaoming Sun. *A Tighter Relation between Sensitivity and Certificate Complexity* (2016)

[38] T. Helmuth, W. Perkins, S. Petti, Correlation decay for hard spheres via Markov chains, (2020) preprint arXiv:2001.05323

[39] H. Huang and B. Sudakov. A counterexample to the Alon-Saks-Seymour conjecture and related problems, *Combinatorica*, Volume 32, 2012, pages 205–219.

[40] H. Huang, Induced subgraphs of hypercubes and a proof of the sensitivity conjecture, Annals of Mathematics 190 (2019), 949-955.

[41] T. Hickey, J. Cohen, Uniform random generation of strings in a context-free language, SIAMJ. Comput. 12(4)(1983)645–655.

[42] M. Jenssen, F. Joos, and W. Perkins, On the hard sphere model and sphere packings in high dimensions, Forum Math. Sigma 7 (2019)

[43] S. Jukna, A.S. Kulikov. On covering graphs by complete bipartite subgraphs. *Discrete Mathematics*, Volume 309, Issue 10, 2009, pages 3399–3403.

[44] J. Kahn, G. Kalai, and Nati Linial. The influence of variables on boolean functions. In *Proc. 29th Ann. IEEE Symp. on Foundations of Comp. Sci.,* 68–80, 1988.

[45] D. M. Kane. A Low-Depth Monotone Function Given by a Low-Depth Decision Tree that is not an Approximate Junta, Theory of Computing Vol. 9 pp. 587-592 (2013).

[46] S. Kannan, Z. Sweedyk, S. Mahaney. Counting and random generation of strings in regular languages. In: Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 551–557 (1995)

[47] P. Keevash and B. Sudakov. Packing triangles in a graph and its complement, *Journal of Graph Theory*, Volume 47, Issue 3, 2004, pages 203–216.

[48] C. Kenyon and S. Kutin. *Sensitivity, block sensitivity, and l-block sensitivity of Boolean functions.* Inf. Comput., 189(1):43–53, 2004

[49] K. Kolipaka and M. Szegedy. Moser and Tardos meet Lovasz. In STOC, pages 235–244, 2011.

[50] Chengyu Lin, Jingcheng Liu, Pinyan Lu. A Simple FPTAS for Counting Edge Covers. In *SODA*, 922–940, 2012.

[51] Jincheng Liu, Pinyan Lu. FPTAS for Counting Monotone CNF. In *SODA* 1531–1548. 2015.

[52] V.A. Markoff, Uber Polynome, die in einem gegebenen Intervall moglichst wenig von Null abweichen, Math. Ann. 77 (1916) 213-258 (translation by J. Grossman of original Russian article in Acad. Sc. St. Petersburg, 1892).

[53] R. Meka. Explicit resilient functions matching Ajtai-Linial. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms.* SIAM, 1132-1148, 2017.

[54] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. J. Chem. Phys., 21(6):1087–1092, 1953

[55] G. Midrijanis, *Exact Quantum Query Complexity for Total Boolean Functions*, preprint, https://arxiv.org/abs/quant-ph/0403168, 2004

[56] M. Minsky and S. Papert, *Perceptrons.* MIT Press, Cambridge, MA., (1969).

[57] Robin A. Moser and Gabor Tardos. A constructive proof of the general Lovasz Local Lemma. In *J. ACM*, 57(2), 2010.

[58] Noam Nisan and Mario Szegedy, *On the degree of boolean functions as real polynomials*, Computational complexity (1994), no. 4, 301–313.

[59] N. Nisan and A. Wigderson. On rank vs. communication complexity. Combinatorica, 15(4):557–565, 1995.

[60] N. Nisan. CREW PRAMs and decision trees. SIAM Journal of Computing, 20(6):999-1007, 1991.

[61] R. O'Donnell, *Analysis of boolean functions.* (2013).

[62] O.Bernardi, O.Giménez, A linear algorithm for the random sampling from regular languages, Algorithmica (2010) 1–16.

[63] J. Orlin. Contentment in graph theory : Covering graphs with cliques. Indagationes Math. 39, pages 406–424, 1977.

[64] R. O'Donnell and R. Servedio. Learning monotone decision trees in polynomial time, in Proceedings of the 21st Annual IEEE Conference on Computational Complexity, pages 213–225, 2006.

[65] James G. Propp and David B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Struct. Algorithms,* 9(1-2):223–252, 1996.

[66] N. J. Pullman and A. Donald. Clique Coverings of Graphs II – Complements of Cliques, Utilitas Math. 19 (1981). 207–213.

[67] L. Pyber. Clique covering of graphs. Combinatorica 6, 393–398 (1986).

[68] R. Reischuk, A Lower Time Bound for Parallel RAM's without Simultaneous Writes, IBM Research Report, RJ343 I. 1982.

[69] D. Rohatgi, J. Urschel, J. Wellens. Progress on two conjectures in clique and biclique partitions, *in preparation.*

[70] J. Shearer. On a problem of Spencer. Combinatorica, 5(3):241–245, 1985.

[71] Alexander Sherstov. *Making polynomials robust to noise.* In Proceedings of the 44th Annual ACM Symposium on Theory of Computing, pages 747-758, 2012.

[72] Hans-Ulrich Simon, *A tight* $\Omega(\log\log n)$*-bound on the time for parallel RAMs to compute non-degenerate Boolean functions*, Proc. 4th Int. Symp. Fundam. Comput. Theory (FCT), volume 158 of LNCS (1983), 439 – 444.

[73] Avishay Tal, *Properties and applications of boolean function composition*, ITCS (2013), 441–454.

[74] A. Tal. On the sensitivity conjecture. In 43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016.

[75] I. Wegener, The critical complexity of all (monotone) Boolean functions and monotone graph properties. Information and Control, 67:212-222, 1985.

[76] D. Weitz. Counting independent sets up to the tree threshold, STOC 2006: 140-149.

[77] R. M. Wilson. Decompositions of complete graphs into subgraphs isomorphic to a given graph. In *Proc. 5th British Combinatorial Conference, Congr. Numer.* XV, 1976, pages 647–659.

[78] D. Wilson. Generating random spanning trees more quickly than the cover time. In STOC, pages 296–303, 1996.

[79] R. Yuster, personal communication. 2018.

[80] R. Yuster. Integer and fractional packing of families of graphs. *Random Structures & Algorithms*, Vol. 26, 2005, pages 110–118.