Hard Problems on Random Graphs

Jan Dreier
Department of Computer Science, RWTH Aachen University, Germany
https://tcs.rwth-aachen.de/~dreier
dreier@cs.rwth-aachen.de

Henri Lotze
Department of Computer Science, RWTH Aachen University, Germany
https://tcs.rwth-aachen.de
lotze@cs.rwth-aachen.de

Peter Rossmanith
Department of Computer Science, RWTH Aachen University, Germany
https://tcs.rwth-aachen.de
rossmani@cs.rwth-aachen.de

Abstract

Many graph properties are expressible in first order logic. Whether a graph contains a clique or a dominating set of size k are two examples. For the solution size as its parameter the first one is W[1]-complete and the second one W[2]-complete meaning that both of them are hard problems in the worst-case. If we look at both problem from the aspect of average-case complexity, the picture changes. Clique can be solved in expected FPT time on uniformly distributed graphs of size n, while this is not clear for Dominating Set. We show that it is indeed unlikely that Dominating Set can be solved efficiently on random graphs: If yes, then every first-order expressible graph property can be solved in expected FPT time, too. Furthermore, this remains true when we consider random graphs with an arbitrary constant edge probability. We identify a very simple problem on random matrices that is equally hard to solve on average: Given a square boolean matrix, are there k rows whose logical AND is the zero vector? The related Even Set problem on the other hand turns out to be efficiently solvable on random instances, while it is known to be hard in the worst-case.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Theory of computation → Complexity theory and logic

Keywords and phrases random graphs, average-case complexity, first-order model checking

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.40

Category Track A: Algorithms, Complexity and Games

Funding Grant DFG-927/15-1 from the German Science Foundation (DFG).

Acknowledgements We thank the refeeres of ICALP for their very helpful comments that helped to improve the readability of the paper.

1 Introduction

The worst-case analysis of problems has a long tradition and has led to a complexity theory that allows us easily to classify many problems. Such complexity theories do not only exist for the time complexity, but also for other resources such as space. There are complexity classes for approximations, parallel computations, randomization, parameterized algorithms, and many more. Usually they come with complete problems under certain reductions.

The average-case analysis of problems is less developed, but Levin showed quite early that there exist problems that are hard to solve even on random inputs [19]. He considered problems together with input distributions and defined reductions that take the probability distribution into account. This also led to complete problems for an analogue of NP in the average-case

47th International Colloquium on Automata, Languages, and Programming (ICALP 2020).
Editors: Artur Czumaj, Anuj Dawar, and Emanuela Merelli; Article No. 40; pp. 40:1–40:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

world. This theory has been constantly refined. For example, Gurevich [17] showed some inherent limitations of these techniques. Ben-David et al. showed a rare connection to the worst-case world: If all problems in NP with a "simple" probability distribution on the inputs can be solved in average polynomial time, then NEXPTIME = EXPTIME [1]. Such connections are extremely rare and the latter one together with all others suffer from a severe problem: They rely on quite unnatural probability distributions. Up to today no connection seems to exist that shows, e.g., that if some natural problem in NP is hard to solve on average under a uniform distribution, then P $\neq$ NP. Such a result would be a big breakthrough.

It has to be noted, however, that it is in general not easy even to find problems that are hard on average (with a uniform distribution). Having such problems is crucial in cryptography. The RSA system is based on the assumption that factoring the product of two primes is hard on average, but we cannot prove today that the existence of an algorithm that can factor in polynomial time on average would imply some unexpected collapse between worst-case complexity classes.

Many problems that are hard in the worst-case become easy on average. Take for example the three-coloring problem on graphs. While NP-complete this problem can be solved in constant time on average when drawing the graph from a uniform distribution of all graphs of size n: It is easy to see that you can find a triangle in expected constant time by just looking for one among the first three vertices, then the next three and so on. Each time you find a triangle with probability $\frac{1}{8}$ and you find a triangle on average with only eight tries. See for example [4] for a similar but more complicated example.

The same holds for finding any fixed size subgraph or induced subgraph. This means that p-Clique, the problem of finding a clique of size k, can be solved in expected $f(k)\mathit{poly}(n)$ time for some function f if the input is a uniformly distributed graph of size n. Parameterized complexity shows that it is unlikely to solve the same problem in $f(k)\mathit{poly}(n)$ time in the worst-case [9]. Fountoulakis, Friedrich, and Hermelin showed that finding cliques is in FPT if the probability in the random graph is an almost arbitrary function of its size [14].

In this paper we look at first-order model checking on uniformly distributed random graphs and more generally on Erdős–Rényi graphs with a constant edge probability. In this model we assume that each possible edge in a graph with n vertices exists independently with a probability of p. While in the worst-case the FO model checking problem seems to become harder the more quantifier alternations we have, this hierarchy collapses when looking at the average time complexity. We will show that the dominating set problem is as hard as the whole model checking problem. We also identify a very natural problem on boolean matrices that has the same complexity: Does a random boolean matrix have k rows whose logical AND is the zero vector? We conclude that this matrix problem and the dominating set problem are hard on *average* (unless the very general model checking problem is easy, which would be unexpected). Finally we consider also the Even Set problem, which has been finally shown to be W[1]-hard in the worst-case [2] and is similar to the above mentioned matrix problem. Nevertheless it turns out that Even Set can be efficiently solved on random instances.

Among the techniques "half"-reductions play an important role. While in a reduction f from A to B you require that $w \in A$ iff $f(w) \in B$, we often need only one direction: Showing "if $w \in A$ then $f(w) \in B$" is sufficient if $f(w) \in B$ holds with a very small probability. Having an algorithm for B we can then solve A as follows. Compute $f(w)$ and find out whether $f(w) \in B$ holds. If not, conclude that $w \notin A$. If yes, then solve $w \in A$ with a very slow, but simple algorithm. As this happens with a small probability it does not spoil the expected running time.

2 Preliminaries

Parameterized complexity

Parameterized complexity was introduced by Downey and Fellows in a series of papers to investigate further what makes problems hard to solve (see, e.g., [5, 6, 7, 8, 9, 13, 22]). Instead of measuring the run time solely as a function on the input length, it may also depend on other parameters of the input. A parameterized problem has therefore a parameter k and the input length n and we classify a problem as *fixed parameter tractable* if it can be solved in time $f(k)\mathit{poly}(n)$ for some computable function f. If an NP-hard problem is fixed parameter tractable, then it runs in polynomial time for every fixed value of k and the degree of the polynomial does not depend on k. In particular this means that there exist efficient algorithms for scenarios where the parameter is small.

In this paper we will look at distributional problems on random graphs and boolean matrices. Here a distributional problem will be a parameterized problem together with a probability distribution of the inputs. Usually we will denote such a distributional problem by stating the problem and the probability distribution separately.

We use the notation of Flum and Grohe [13] for parameterized problems. The first two important problems we consider are the dominating set problem on undirected graphs and a simple problem on boolean matrices:

▶ **Definition 1.**

p-Dominating Set

Input:	A graph G and $k \in \mathbf{N}$.
Parameter:	k
Problem:	Is there a dominating set of size $\leq k$ for G?

p-Matrix$(\wedge)$

Input:	A boolean matrix $M \in \{0,1\}^{n\times n}$ and $k \in \mathbf{N}$.
Parameter:	k
Problem:	Are there k rows in M whose logical AND is the zero vector?

Logic on graphs and the zero-one law

We use graphs as a structure (V, E) where V is the vertex set and E the binary edge relation. Instead of Euv we will write $u \sim v$, which expresses that there is an edge between u and v in an undirected graph. First-order (FO) formulas on graphs are atomic formulas of the form $x = y$ or $x \sim y$ or one of the following: $\phi \wedge \psi$, $\phi \vee \psi$, $\neg\phi$, $\forall x\phi$, $\exists x\phi$, where ϕ and ψ are already FO-formulas. The semantics are as expected. A sentence is a formula without free variables.

▶ **Definition 2.** We define the first-order model checking problems on graphs. The more general problem on relational structures can be reduced to this more special problem [13].

p-MC(FO)

Input:	A first-order sentence ϕ and a graph G
Parameter:	$\lvert\phi\rvert$, the length of ϕ
Problem:	Does ϕ hold in G, i.e., $G \models \phi$?

For example, the formula $\exists x_1 \exists x_2 \ldots \exists x_k \forall y \bigvee_i (x_i = y \vee x_i \sim y)$ expresses that a graph has a dominating set of size at most k. If a formula ϕ holds for a graph G we write $G \models \phi$. If a formula ϕ follows from a set of formulas Φ we write $\Phi \models \phi$. This is the case iff there is a formal derivation of ϕ from Φ, which we write as $\Phi \vdash \phi$. Sometimes we will use colored graphs, which we represent by a graph G and a coloring function χ mapping vertices to a set of colors. Formulas can speak about colors via atomic formulas of the form $\chi(x) = \mathit{red}$ and we write $(G, \chi) \models \phi$ if the formula ϕ is true for G with colors χ.

By $G(n,p)$ we denote an Erdős–Rényi-graph with n vertices and edge probability p, where edges exist independently from each other with a probability of exactly $p \in [0,1]$. Fagin [11] proved the zero-one law for first-order sentences, which states that for every sentence ϕ either $\lim_{n\to\infty} \Pr[G(n,1/2) \models \phi] = 0$ or $\lim_{n\to\infty} \Pr[G(n,1/2) \models \phi] = 1$. With other words, a graph property that is expressible in first-order logic either holds asymptotically almost surely or almost never. Given ϕ as an input, it can be decided whether the limit is 0 or 1 and Grandjean showed that it turns out to be a PSPACE-complete problem [15]. An important role in the proof of the zero-one law play the so-called *extension axioms* (not to be confused with the axiom of extension in Zermelo–Fraenkel set theory). They state that every constant-size set of vertices is connected in every possible way to other vertices. For a set or vector of variables $x_1, \ldots, x_k$ we will often write $\bar{x}$. With this notation an extension axiom can be written as

$$\forall \bar{x} \forall \bar{y} \exists z \Big(\bigwedge_{i,j} x_i \neq y_j \to \bigwedge_i (x_i \sim z \wedge y_i \not\sim z) \Big).$$

For an extension axiom it is easy to see that it holds almost surely, but if we look at the whole set Φ of all extension axioms it turns out that there is only one countable model up to isomorphisms, the so-called *Rado graph*, which contains every finite and countable infinite graph as an induced subgraph. Hence by the Łoś–Vaught Theorem [20, 23], $\Phi \models \phi$ or $\Phi \models \neg\phi$ for every first-order sentence ϕ. This means also that either $\Phi \vdash \phi$ or $\Phi \vdash \neg\phi$. To find out out which one is true we can just enumerate all proofs. Note that in a proof only a finite number of formulas from Φ are used and the proof itself is of course also finite. The result by Grandjean states that this can be done in polynomial space.

All these observations suggest that FO-model checking should be easy on random graphs: Just find out from ϕ alone whether it holds almost surely or almost never. Then verify that this is indeed the case for the given G. The strategy of using an abundance of witnesses suggests itself, just as the triangle finding described in the introduction. While this intuition is correct for purely existential formulas, life becomes much harder when considering formulas with quantifier alternations.

In worst-case complexity FO-model checking with a fixed number of quantifier alternations form the complete problems for the A-hierarchy [12]. Among known results about the relationship to other complexity classes are W[t] $\subseteq$ A[t] and W[1] = A[1]. To today's knowledge this hierarchy appears to be proper, see e.g. [3]. A collapse of the A-hierarchy implies a collapse of both the W-hierarchy and of the polynomial hierarchy.

In this paper we investigate how hard FO-model checking is on average, which could be interpreted as looking at the average-case analogue to the A-hierarchy. In the worst-case model checking purely existential formulas is already W[1] = A[1]-complete, while it is easy to see that you can achieve expected FPT time (because of abundance of witnesses). If we turn to edge probabilities apart from $\frac{1}{2}$ and look at Erdős–Rényi graphs $G(n,p)$ with a constant p the problem stays in expected FPT time. Grohe showed that for sparse Erdős–Rényi graphs $G(n, d(n)/n)$ with $d(n) = n^{o(1)}$ the whole p-MC(FO) can be solved in expected FPT time [16]. The latter result also holds for graphs with vertex colors and it turns out that for colored random graphs Grohe's result is optimal with regard to the edge density: For no $\epsilon > 0$ and $G(n, n^\epsilon/n)$ is it possible to solve colored-p-MC(FO) in expected FPT time unless AW[$*$] $\subseteq$ FPT/poly [10].

Universal sets, bisectors, and colorings

▶ **Definition 3.** Let $n \in \mathbf{N}$ and $k \in \mathbf{N}$ with $n \geq 2k$.

1. A family $\mathcal{U}$ of functions $[n] \mapsto \{0,1\}$ is called an (n,k)-*universal set* if for every subset $M \subseteq [n]$ of size $|M| = k$ and every $M' \subseteq M$ there is a function $f \in \mathcal{U}$ such that $f(t) = 0$ for every $t \in M'$ and $f(t) = 1$ for every $t \in M - M'$.
2. A family $\mathcal{B}$ of functions $[n] \mapsto \{0,1\}$ is called a k-*universal bisector family* if for every subset $M \subseteq [n]$ of size $|M| = k$ there is a function $f \in \mathcal{B}$ such that $f(t) = 0$ for every $t \in M$ and every function f bisects $[n]$ in two sets of almost the same size: $|f^{-1}(0)| = \lceil n/2 \rceil$.
3. A family $\mathcal{C}$ of functions $[n] \mapsto \{\mathit{black}, \mathit{white}, \mathit{gray}\}$ is called a k-*universal coloring* if for every subset $M \subseteq [n]$ of size $|M| = k$ and every $M' \subseteq M$ there is a function $f \in \mathcal{C}$ such that $f(t) = \mathit{black}$ for every $t \in M'$ and $f(t) = \mathit{white}$ for every $t \in M - M'$. Moreover, $|f^{-1}|(\mathit{gray}) = \lceil n/2 \rceil$ for every $f \in \mathcal{C}$.

(n,k)-universal sets are a well-known concept that has been used, e.g., in the derandomization of algorithms. Naor, Schulman, and Srinivasan designed such a family that can be constructed in linear time and has size $2^k k^{O(\log k)} \log n$ [21].

The concept of a universal bisector is somehow orthogonal to (n,k)-universal sets. Combining both concepts leads to (n,k)-universal colorings that will always color half of the nodes gray.

A simple idea to build a k-universal bisector family of small size is this: For every size k subset M of $[2k]$ we define the function

$$b_M\colon [n] \to \{0,1\},\; b_M(t) = \begin{cases} 0 & \text{if } t \bmod 2k \in M \\ 1 & \text{otherwise,} \end{cases} \tag{1}$$

where $a \bmod b$ is the remainder when dividing a by b.

Assume that $S \subseteq [n]$ with $|S| \leq k$. If we choose M such that it contains $t \bmod 2k$ for every $t \in S$ (and some more elements), then clearly $b_M(i) = 0$ for every $i \in S$. Hence, we have a k-universal bisector family. If n is a multiple of $2k$ then $|b_M^{-1}(0)| = n/2$ because every group of $2k$ elements is split equally. The last group, however, can be split unevenly leading to an error of $O(k)$. Algorithmically such a family of functions that bisects with a small error would be sufficient for our purposes. It is, however, possible to achieve perfect balance at a small cost in the size of the family, which makes their application slightly easier. The next lemma shows that universal bisector families exist.

We leave the smallest possible size of such families as an open question as it is not a critical issue for the results of this paper, but give two comments on the issue: If $n = 2k$ you have to use all $\binom{2k}{k} = \Theta(k^{-1/2}4^k)$ possible balanced bipartitions and it seems that a size of $O^*(4^k)$ is already optimal. If n is much bigger, however, a random perfect bipartition works with a relatively high probability of $\Omega^*(2^{-k})$ and suggests that families of size $O^*(2^k)$ exist, although it is not immediately clear how to construct families of that size.

▶ **Lemma 4.** *For every $k \in \mathbf{N}$ and $n \geq 2k(2k+1)$ there is a k-universal bisector family of size $O(4^k k)$. A table of all functions in the family can be computed in time $O(4^k kn)$, which is linear in the size of the table.*

Proof. We use a slight modification of the construction in (1). Let $n = 2kd + r$ with $r < 2k$. Let us call the last r elements of $[n]$ the *jokers*. We define

$$b_M(t) = \begin{cases} 0 & \text{if } t < 2kd \text{ and } t \bmod 2k \in M, \\ 1 & \text{if } t < 2kd \text{ and } t \bmod 2k \notin M, \\ t \bmod 2 & \text{if } t \geq 2kd. \end{cases}$$

Such a function $b_M(t)$ maps exactly $\lceil n/2 \rceil$ elements to 0 and is therefore perfectly balanced, but those function do not give us a universal family of bisectors. If S contains odd jokers, not all of S is mapped to 0. We can overcome this problem by using $2k + 1$ families build in this way, but where each family uses a different interval in $[n]$ to place the jokers. In that way for each $S \subseteq [n]$ of size up to $2k$ there will be one family where S does not contain a joker (by the pigeon-hole principle). There is enough space for these intervals if $n \geq 2k(2k + 1)$ and the resulting universal family of bisectors has size $(2k + 1)4^k$. ◀

Combining k-universal bisectors and (n, k)-universal sets leads easily to (n, k)-universal colorings.

▶ **Lemma 5.** *For every $k \in \mathbf{N}$ and $n \geq 2k(2k+1)$ there is an (n, k)-universal coloring family of size $8^k k^{O(\log k)} \log n$. A table of all functions in the family can be computed in linear time.*

Proof. We use a k-universal bisector family $\mathcal{B}$ of size $O(4^k k)$ from Lemma 4 and an (n, k)-universal set $\mathcal{U}$ of size $2^k k^{O(\log k)} \log n$ according to [21]. For $f \in \mathcal{U}$ and $g \in \mathcal{B}$ we define a function $h \colon [n] \to \{\mathit{black}, \mathit{white}, \mathit{gray}\}$ as follows:

$$h(v) = \begin{cases} \mathit{gray} & \text{if } g(v) = 1 \\ \mathit{black} & \text{if } g(v) = 0 \text{ and } f(v) = 0 \\ \mathit{white} & \text{if } g(v) = 0 \text{ and } f(v) = 1 \end{cases}$$

It is easy to see that the set of all such h's forms an (n, k)-universal family of colorings. ◀

3 Results

We define the following three formulas:

$$\phi \equiv \forall \bar{x} \forall \bar{y} \exists z \Big(\bigwedge_{i,j=1}^{k} x_i \neq y_j \to \bigwedge_{i=1}^{k} (x_i \sim z \wedge y_i \not\sim z) \Big)$$

$$\phi' \equiv \forall \bar{x} \forall \bar{y} \exists z \Big(\bigwedge_{i=1}^{k} \big(\chi(x_i) = \mathit{black} \wedge \chi(y_i) = \mathit{white} \big) \to \chi(z) = \mathit{gray} \wedge \bigwedge_{i=1}^{k} (x_i \sim z \wedge y_i \not\sim z) \Big)$$

$$\phi'' \equiv \forall \bar{x} \forall \bar{y} \exists z \Big(\bigwedge_{i=1}^{k} \big(\chi(x_i) = \mathit{black} \wedge \chi(y_i) = \mathit{white} \big) \to \chi(z) = \mathit{gray} \wedge \bigwedge_{i=1}^{k} (x_i \not\sim z \wedge y_i \not\sim z) \Big)$$

The next lemma uses k in the formulas ϕ and ϕ' as the parameter.

▶ **Lemma 6.** *If there is an algorithm that, given a graph $G = G(n, 1/2)$ and a coloring $\chi \colon \mathcal{G} \to \{\mathit{gray}, \mathit{black}, \mathit{white}\}$, can decide in expected FPT time whether $(G, \chi) \models \phi'$, then there is an algorithm that can decide for $G = G(n, 1/2)$ in expected FPT time whether $G \models \phi$.*

Proof. Let X be a $2k$-universal family of colorings. Given a graph G first solve $(G,\chi) \models \phi'$ for every $\chi \in X$ in expected FPT time. If the answer is *yes* for every χ, then we can conclude $G \models \phi$, as by using a universal family of colorings, we have covered all distinguishable ways to color the nodes of G: Assume that $(G,\chi) \models \phi'$ for every $\chi \in X$, but $G \not\models \phi$. Then there exist $\bar{x}, \bar{y}$ where $\bar{x}$ and $\bar{y}$ are distinct such that for all z, $\bigwedge_{i=1}^{k}(x_i \sim z \wedge y_i \not\sim z)$ is unsatisfied. However, as $|\bar{x}| = |\bar{y}| = k$ and X is a $2k$-universal family of colorings, there is in particular a coloring $\chi \in X$ such that $\chi(\bar{x}) = \mathit{black}$ and $\chi(\bar{y}) = \mathit{white}$. If we assume that $G \not\models \phi$ for these particular $\bar{x}, \bar{y}$, then clearly also $(G,\chi) \not\models \phi'$ for these choices of χ and $\bar{x}, \bar{y}$.

We cannot tell whether $G \models \phi$ holds or not if $(G,\chi) \not\models \phi'$ for at least one coloring $\chi \in X$. The probability that this happens is exponentially small in n as we will show in the following. The negation of ϕ' reads

$$\exists\bar{x}\exists\bar{y}\forall z\Big(\bigwedge_{i=1}^{k}\big(\chi(x_i) = \mathit{black} \wedge \chi(y_i) = \mathit{white}\big) \wedge \Big(\chi(z) = \mathit{gray} \to \bigvee_{i=1}^{k}(x_i \not\sim z \vee y_i \sim z)\Big)\Big).$$

Once the coloring and $\bar{x}, \bar{y}$ are fixed, the probability that a gray vertex z is not connected to some x_i or connected to some y_i, is exactly $1 - 2^{-2k}$. This happens with all $\frac{n}{2} + O(k)$ many gray vertices with a probability of $(1-2^{-2k})^{\frac{n}{2}+O(k)}$. Altogether the probability of $G \not\models \phi'$ is

$$\sum_{\chi \in X} P[(G,\chi) \not\models \phi'] \le |X|\binom{n}{k}\binom{n-k}{k}\Big(1 - \frac{1}{2^{2k}}\Big)^{\frac{n}{2}+O(k)} =$$

$$= 8^k n^{O(k)}\Big(1 - \frac{1}{2^{2k}}\Big)^{\frac{n}{2}+O(k)} = e^{-n2^{-2k-1}+O(k\log n)},$$

because there are $|X| \le 8^k \mathit{poly}(n)$ (Lemma 5) many colorings and at most $\binom{n}{k}\binom{n-k}{k} = n^{O(k)}$ ways to choose $\bar{x}$ and $\bar{y}$. In that case, we can solve $G \models \phi$ by brute force in $n^{O(k)}$ time. For big enough n, the expected running time of the complete algorithm then remains in expected FPT time as long as $(G,\chi) \models \phi'$ is decidable in expected FPT time. The tradeoff works as long as n^k is subexponential and for large k the problem is automatically in FPT even in the worst-case. ◀

Let $\bar{G}_\chi = (V, E')$ be defined as follows: If $\chi(u) = \mathit{gray}$ and $\chi(v) = \mathit{black}$ or $\chi(u) = \mathit{black}$ and $\chi(v) = \mathit{gray}$ then $uv \in E'$ iff $uv \notin E$. Otherwise $uv \in E'$ iff $uv \in E$.

Informally speaking, $\bar{G}_\chi$ is the same graph as G, but edges are replaced by non-edges and vice versa between a vertex pair that is colored black and gray. It is important to note that if G is random then $\bar{G}_\chi$ is also random, although the edge probability of the flipped edges of $\bar{G}_\chi$ are inverted. This poses no problem if the edge probability is 1/2, but wrecks havoc when it is not. Edge probabilities different from 1/2 are discussed in Section 4.

▶ **Lemma 7.** *Let G be a graph and χ a coloring. Then $(G,\chi) \models \phi'$ iff $(\bar{G}_\chi,\chi) \models \phi''$.*

Proof. The only difference between ϕ' and ϕ'' is $x_i \sim z$ versus $x_i \not\sim z$. This subformula is only relevant when x_i is black and z is gray, as it is guarded by that condition. ◀

A coloring $\chi\colon [n] \to \{\mathit{black}, \mathit{white}, \mathit{gray}\}$ is *balanced* if $\lceil n/2 \rceil$ numbers are mapped to *gray*.

▶ **Lemma 8.** *If we can solve p-*MATRIX*$(\wedge)$ on random $n \times n$-matrices in expected FPT time, then we can solve $(G,\chi) \models \phi''$ for every balanced coloring χ on $G(n, 1/2)$ in expected FPT time.*

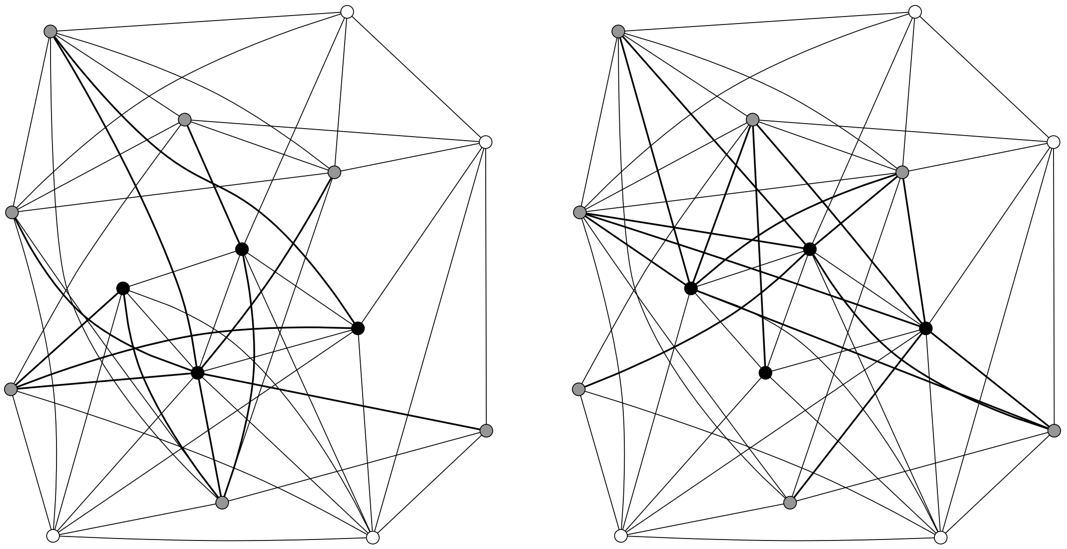

Figure 1 A graph G (left) with a node coloring χ and the corresponding graph $\bar{G}_\chi$ (right) with gray-black edges flipped.

Proof. Let $G = (V, E)$ be a random graph with n vertices and $\chi \colon \mathcal{G} \to \{\mathit{gray}, \mathit{black}, \mathit{white}\}$ be a balanced coloring. Assume first that n is even and that $V = \{v_1, \dots, v_{n/2}, u_1, \dots, u_{n/2}\}$ where v_i are not gray and u_i are gray. We construct a matrix $M \in \mathbf{F}_2^{n/2 \times n/2}$ such that $M_{ij} = 0$ iff $v_i u_j \in E$. This matrix is random and therefore we can find out in expected FPT time whether there are $2k$ rows $i_1, \dots, i_{2k}$ whose logical AND is 0. If this is not the case then $(G, \chi) \models \phi''$: Choosing $\bar{x}$ and $\bar{y}$ corresponds to picking $2k$ rows $i_1, \dots, i_{2k}$ of M. Choosing z corresponds to picking a column j of M. As M is a no-instance regardless what $i_1, \dots, i_{2k}$ are, the AND of the corresponding rows is never 0. So there is a column j with $M_{ij} = 1$ for all $i \in \{i_1, \dots, i_{2k}\}$ and correspondingly in G there is a gray z for every black $\bar{x}$, white $\bar{y}$ that is non-adjacent to all of $\bar{x}$ or $\bar{y}$ and we can conclude $(G, \chi) \models \phi''$.

Otherwise (if there are no such $2k$ rows) we can test in time $n^{O(k)}$ whether $(G, \chi) \models \phi''$. The probability for this to happen is at most $\binom{n}{2k}(1 - \frac{1}{2^{2k}})^{n/2}$ because there are $\binom{n}{2k}$ ways to choose $2k$ rows and for each column the probability is 2^{-2k} that it contains ones in all selected rows.

It remains to consider an odd n. In that case χ colors $\lceil n/2 \rceil$ vertices gray. Using the above construction leads to a matrix M with one more column than rows. The premise of the lemma allows us, however, only to assume that p-Matrix($\wedge$) is solvable on square matrices. If we just remove the last column of M we are left with a random square matrix. The probability is still exponentially small that the truncated matrix is a yes-instance of p-Matrix($\wedge$), but if it is a no-instance we can conclude that M is also a no-instance and then $(G, \chi) \models \phi''$ follows. Otherwise, we can again use a brute-force solution in time $n^{O(k)}$. ◀

▶ **Lemma 9.** *If we can solve p-Dominating Set on $G(n, 1/2)$ in expected FPT time, then we can solve p-Matrix($\wedge$) on random matrices in expected FPT time.*

Proof. (Sketch) Let $M \in \mathbf{F}_2^{n \times n}$ be a random matrix. Let $\bar{M} = M \oplus 1$ (pointwise negation of M). Then M contains k rows whose AND is zero iff the directed graph H with $\bar{M}$ as its adjacency matrix has a dominating set of size k, which corresponds to the logical OR of k rows in $\bar{M}$ to be equal to the one vector.

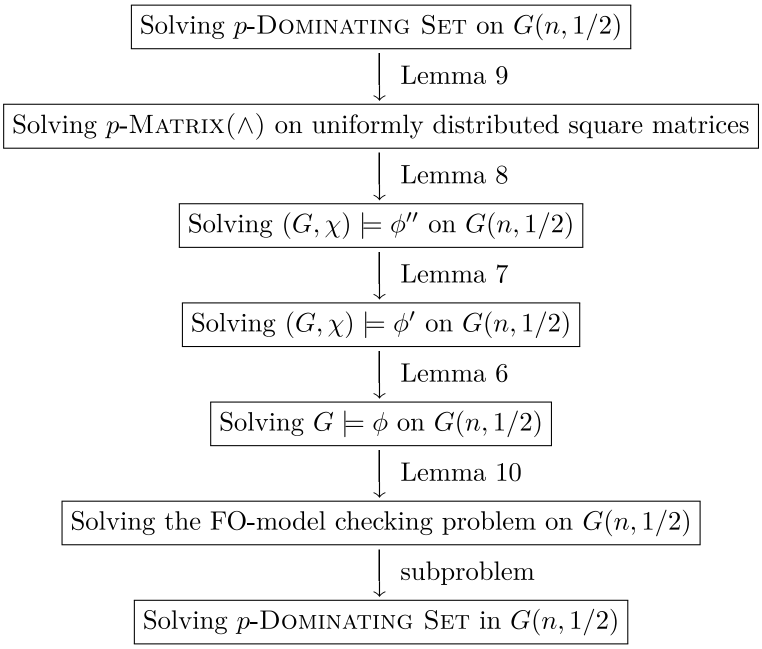
Solving p-Dominating Set on $G(n,1/2)$
Lemma 9
Solving p-Matrix($\wedge$) on uniformly distributed square matrices
Lemma 8
Solving $(G,\chi) \models \phi''$ on $G(n,1/2)$
Lemma 7
Solving $(G,\chi) \models \phi'$ on $G(n,1/2)$
Lemma 6
Solving $G \models \phi$ on $G(n,1/2)$
Lemma 10
Solving the FO-model checking problem on $G(n,1/2)$
subproblem
Solving p-Dominating Set in $G(n,1/2)$

Figure 2 Structure of the main proof. "Solving" means that the problem can be solved in expected FPT time. $A \longrightarrow B$ means that if A is in expected FPT time then so is B.

By using k-universal bisector families we can assume that a dominating set is among the first third of the vertices and that the k rows of a solution can be found in the upper third of $\bar{M}$. Decompose $\bar{M}$ into nine blocks and rearrange them as follows:

$$\bar{M} = \begin{pmatrix} A & B & C \\ D & E & F \\ G & H & I \end{pmatrix} \quad \to \quad M' = \begin{pmatrix} D' & B & A^T \\ B^T & E' & C \\ A & C^T & F' \end{pmatrix}$$

Here B^T is the transposed matrix B and A' is a symmetric matrix built from the upper triangular part of A. It is easy to see that M' is symmetric and random, if M is random. This means that M' is the adjacency matrix of a random, undirected graph G. It is also clear that if $\bar{M}$ contains k rows whose logical OR is the one vector, then M' contains $3k$ such rows and G has a dominating set of size $3k$. We can construct G and find out whether it has a dominating set of size $3k$. If not, then we know that M does not contain k rows whose logical AND is the zero vector. Otherwise, we can run a brute-force $n^{O(k)}$ algorithm, because the probability that we have to do this is at most $\binom{n}{3k}(1 - \frac{1}{2^{3k}})^{n-3k}$. ◀

▶ **Lemma 10.** *If there is an algorithm that can decide $G \models \phi$ on $G(n,1/2)$ in expected FPT time, then there is an algorithm that solves p-MC(FO) on $G(n,1/2)$ in expected FPT time.*

Proof. Let Φ be the set of extension axioms and ψ be a first-order formula on the logic of graphs. Remember that then either $\Phi \vdash \psi$ or $\Phi \vdash \neg\psi$ [11]. We can enumerate all proofs that use axioms in Φ in ascending order of length. Eventually we will find a proof $\Phi' \vdash \psi$ or $\Phi' \vdash \neg\psi$ for a finite subset $\Phi' \subseteq \Phi$. In our formulation of extension axioms (where $\bar{x}$ and $\bar{y}$ have the same size and only $x_i \neq y_j$, but not $x_i \neq x_j$, $y_i \neq y_j$ is required), a longer axiom implies all shorter ones. Therefore there is a single extension axiom ϕ with $2k+1$ variables such that $\phi \vdash \psi$ or $\phi \vdash \neg\psi$. The length of ϕ and k are bounded by a function of the length of ψ because the whole proof that we found depends only on ψ. We decide in expected FPT time on the parameter $|\phi|$ and therefore also on the parameter $|\psi|$ whether $G \models \phi$. If the answer is *yes* we can conclude whether $G \models \psi$ holds and we are done. The answer is *no* only with an exponentially small probability in the number n of vertices if $|\phi|$ is small (for example if $|\phi| = O(\log n)$): There are $n - 2k$ possibilities to choose z outside of $\bar{x}$ and $\bar{y}$ and then the probability that z is correctly connected to them is exactly 2^{-2k}. Hence the probability of *no* is at most $n^{2k}(1 - 2^{-2k})^{n-2k} = \Omega(e^{-n/2k+k\log n})$ and we can use a brute-force $n^{O(k)}$ algorithm in that case. ◀

▶ **Theorem 11.** *p-Dominating Set can be solved on $G(n, 1/2)$ in expected FPT time iff p-MC(FO) can be solved on $G(n, 1/2)$ in expected FPT time.*

Proof. Assume p-Dominating Set can be solved on $G(n, 1/2)$ in expected FPT time. By Lemmas 9, 8, 7, 6, and 10 we can conclude that p-MC(FO) can be solved on $G(n, 1/2)$ in expected FPT time. The other direction is trivial as p-Dominating Set is a special case of p-MC(FO), where the length of the formula is linear in the size of the sought-after dominating set. See Figure 2 for an overview. ◀

4 Playing with the probability

Up to now we were looking at the uniform distribution of graphs, which corresponds to an edge probability of 1/2. It seems at first glance that all the proof techniques should also work for an arbitrary constant probability of $0 < p < 1$. A close look at the proofs, however, shows that Lemma 7 uses the fact that $p = 1/2$ in a crucial way: We flip gray–black edges, which changes their probability from p to $1 - p$, which is only harmless when $p = 1/2$. The next lemma shows that we can prove a variant for an arbitrary edge probability.

▶ **Lemma 12.** *If we can solve $G \models \phi''$ on $G(n, p)$ in expected FPT time for some rational number $0 < p \leq 1/2$, then we can also solve $G \models \phi'$ on $G(n, p)$ in expected FPT time.*

Proof. Let G' be a graph and χ a coloring and $\bar{G}'_\chi$ be the same graph with edges between a black and gray vertex flipped. Then $(\bar{G}'_\chi, \chi) \models \phi''$ iff $(G', \chi) \models \phi'$. However, if the edge probability in G' is p then the edge probability in $\bar{G}'_\chi$ is different: $1 - p$ for gray–black edges and p for the others. Assume we can change that by turning every gray–black edge that is present in $\bar{G}'_\chi$ into a non-edge with probability $\frac{p}{1-p}$ such that the resulting probability for each edge is exactly p. Let us call the resulting graph $\bar{G}''_\chi$, which is distributed as $G(n, p)$ if G' is distributed as $G(n, p)$.

It is easy to see that $(G', \chi) \models \phi'$ follows from $(\bar{G}''_\chi) \models \phi''$ because ϕ'' remains true when removing edges. Nevertheless, the probability of $(\bar{G}''_\chi) \models \phi''$ is exponentially close to one. Therefore we can solve $(G', \chi) \models \phi'$ as follows: First find out whether $(\bar{G}''_\chi, \chi) \models \phi''$ in expected FPT time. If the answer is yes, we can conclude that $(G', \chi) \models \phi'$. Otherwise, we solve $(G', \chi) \models \phi'$ in $n^{O(k)}$ time.

We have, however, assumed that we can delete an edge with probability $p/(1 - p)$, which would be true for a randomized algorithm. Using bisector families iteratively by applying the next family on the vertices that were mapped to 0 we can assume that the $\bar{x}$, $\bar{y}$ that are witnesses for $(\bar{G}''_\chi, \chi) \not\models \phi''$ are among the first ϵn vertices for any $\epsilon > 0$. Instead of testing whether ϕ'' holds on the whole graph we can now test only the subgraph induced by the first ϵn black and white vertices and all gray vertices. There are at most ϵn^2 edges between a relevant black and some gray vertex. Hence, we need only to simulate ϵn^2 coin tosses with a heads probability of $p(1 - p)$, which is a rational number. Using von Neumann's trick we can simulate such a coin toss using expected $O(1)$ random bits [24]. Chernoff bounds show that the total number of such bits needed remains $O(\epsilon n^2)$ with a probability exponentially close to one. We can thus use the edges between gray vertices, there are $\binom{n/2}{2}$ many, as a coin toss with probability p. If we run out of simulated random bits, we can use a brute-force algorithm. ◀

Finally, we show that changing the probability does not make the problems harder or easier. This shows a certain robustness of the average-case complexity of p-MC(FO), p-Matrix($\wedge$), and p-Dominating Set.

▶ **Theorem 13.** *Let $0 < p, q < 1$, $p, q \in \mathbf{Q}$. p-MC(FO) can be solved on $G(n,p)$ in expected FPT time iff dominating set can be solved on $G(n,q)$ in expected FPT time.*

Proof. You can check that all steps in Figure 2 work for a probability other than $1/2$ except Lemma 7 (Lemmas 8 and 9 change p to $1-p$ and together leave it untouched). Plugging in Lemma 12 instead makes the whole chain work for any $0 < p \leq 1/2$. If we can solve p-MC(FO) on $G(n,p)$ we can solve it on $G(n,1-p)$, too, by complementing the graph and interchanging $\sim$ and $\not\sim$ in the formula. So far this shows that all problems in Figure 2 are in expected FPT or none of them on inputs distributed with an edge probability p for the graph problems and a probability of p that an entry is 1 for p-Matrix($\wedge$), but we still have to show that we can change the probability to q without making the problem harder or easier.

For this purpose we use p-Matrix($\wedge$). Let M be a matrix with probabilty p and we assume that $p < q$, otherwise we can invert the matrix and use $1-p$ and $1-q$. Similar to the proof of Lemma 12 we use iterated bisectors to reduce the problem of finding k rows to finding them in the first ϵn rows. We can then take a square submatrix consisting of the first ϵn rows and columns. Using the entries outside of this submatrix as coin tosses with heads probability p, we simulate coin tosses with a head probability of $1-q/p$. As before the probability is exponentially small that we do not succeed in the simulation (and have to use a brute-force algorithm). We use the $1-q/p$-coins to change a 1 in the submatrix to a 0 with probability $1-q/p$. Then the probability of a 1 in the submatrix becomes q. Now we can use the postulated algorithm to solve p-Matrix($\wedge$) on the transformed submatrix. If the answer is no for all bisectors then we can answer no. If the answer is at least once yes, which happens with an exponentially small probability, we can use a brute-force algorithm. ◀

The technique used above relies on p and q being rational numbers as we need to simulate a coin toss with head probability $1-q/p$, which is not necessarily possible for uncomputable probabilies (or even for very inefficiently computable ones).

5 Average Case Complexity of Even Set

The problem p-Matrix($\wedge$) turned out to be as hard as p-MC(FO) on random graphs. Instead of looking for k rows whose logical AND is the zero vector, we can also consider the related problem where we look for k rows whose exclusive or is zero. This variant is actually a problem well-known under the name Even Set and has many other equivalent definitions.

▶ **Definition 14.**

p-Even Set

Input: A matrix $A \in \mathbf{F}_2^{n \times n}$ and a number $k > 0$
Parameter: k
Problem: Are there k rows in A whose exclusive or is $\mathbf{0}$?

This problem was one of the original open problems in Downey and Fellows' book on parameterized complexity [8] and one of the few that has not been solved for a long time. Bhattacharyya et al. recently succeeded to classify the problem as W[1]-hard [2] under randomized fpt-reductions. As the problem is very similar to p-Matrix($\wedge$) we could expect that it is similarly hard on random matrices. It turns out, however, that we can solve it on uniformly distributed boolean square matrices in expected FPT time.

▶ **Theorem 15.** *p-Even Set can be solved in expected FPT time on uniformly distributed random square matrices.*

Proof (Sketch). Assume that M is a boolean $n/2 \times n$ matrix for an even n and $\Pr[M_{ij}] = 1/2$ independently for every $1 \leq i \leq n/2$, $1 \leq j \leq n$. It is easy to see and well-known (see, e.g., [18]) that the probability that M has not full rank is exponentially small in n.

Hence, we can proceed as follows to solve p-Even Set: First use a universal bisector family on a square matrix to select half the rows to form an $n/2 \times n$ matrix. If the original matrix was a yes-instance of p-Even Set, then it contained k rows whose exclusive or is the zero vector. Then the same holds for at least one of the transformed $n/2 \times n$ matrices. We check all of them for full rank. If all have full rank, then we conclude that the original matrix was a no-instance. Otherwise we use a brute-force $n^{O(k)}$ algorithm. This happens with small probability: As the $n/2 \times n$-matrices are random, all of them have full rank with high probability. The proof can easily be adjusted for odd n. ◀

It has to be noted that p-Even Set might behave quite differently on rectangular matrices. If a matrix has n columns, but n^2 rows it seems impossible to reduce the problem back to a square matrix and it is quite possible that the problem is then hard on average.

6 Conclusion

We have shown that the dominating set problem, which is with one quantifier alternation fairly low in the hierarchy of first-order definable properties, is nevertheless as hard to solve on average as any other problem that is first-order expressible. Stronger evidence for hardness under random instances would be a theorem that bridges the worlds of average-case and worst-case complexities. For example, some collapse in the W- or A-hierarchies implied by an efficient algorithm for dominating set on average instances would be a breakthrough.

A detail that is missing in this paper is the generalization of Theorem 13 to non-rational probability. We believe that more complicated techniques beyond the scope of this paper prove a generalization to even non-computable numbers, but we leave it as an open question to find a simple argument, which probably exists.

References

1 Shai Ben-David, Benny Chor, Oded Goldreich, and Michael Luby. On the theory of average case complexity. *J. Comput. Syst. Sci.*, 44(2):193–219, 1992. doi:10.1016/0022-0000(92)90019-F.

2 Arnab Bhattacharyya, Édouard Bonnet, László Egri, Suprovat Ghoshal, Karthik C. S., Bingkai Lin, Pasin Manurangsi, and Dániel Marx. Parameterized intractability of even set and shortest vector problem. *CoRR*, abs/1909.01986, 2019. arXiv:1909.01986.

3 Ralph Christian Bottesch. On W[1]-hardness as evidence for intractability. In Igor Potapov, Paul G. Spirakis, and James Worrell, editors, *43rd International Symposium on Mathematical Foundations of Computer Science, MFCS 2018, August 27-31, 2018, Liverpool, UK*, volume 117 of *LIPIcs*, pages 73:1–73:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.MFCS.2018.73.

4 Amin Coja-Oghlan and Anusch Taraz. Colouring random graphs in expected polynomial time. In Helmut Alt and Michel Habib, editors, *STACS 2003, 20th Annual Symposium on Theoretical Aspects of Computer Science, Berlin, Germany, February 27 - March 1, 2003, Proceedings*, volume 2607 of *Lecture Notes in Computer Science*, pages 487–498. Springer, 2003. doi:10.1007/3-540-36494-3_43.

5 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

6 Rodney G. Downey and Michael R. Fellows. Fixed-parameter intractability. In *Proceedings of the Seventh Annual Structure in Complexity Theory Conference, Boston, Massachusetts, USA, June 22-25, 1992*, pages 36–49. IEEE Computer Society, 1992. doi:10.1109/SCT.1992.215379.

7 Rodney G. Downey and Michael R. Fellows. Fixed parameter tractability and completeness. In Klaus Ambos-Spies, Steven Homer, and Uwe Schöning, editors, *Complexity Theory: Current Research, Dagstuhl Workshop, February 2-8, 1992*, pages 191–225. Cambridge University Press, 1992.

8 Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999. doi:10.1007/978-1-4612-0515-9.

9 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

10 Jan Dreier and Peter Rossmanith. Hardness of FO model-checking on random graphs. In Bart M. P. Jansen and Jan Arne Telle, editors, *14th International Symposium on Parameterized and Exact Computation, IPEC 2019, September 11-13, 2019, Munich, Germany*, volume 148 of *LIPIcs*, pages 11:1–11:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.IPEC.2019.11.

11 Ronald Fagin. Probabilities on finite models. *J. Symb. Log.*, 41(1):50–58, 1976. doi:10.1017/S0022481200051756.

12 Jörg Flum and Martin Grohe. Model-checking problems as a basis for parameterized intractability. *Logical Methods in Computer Science*, 1(1), 2005. doi:10.2168/LMCS-1(1:2)2005.

13 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006. doi:10.1007/3-540-29953-X.

14 Nikolaos Fountoulakis, Tobias Friedrich, and Danny Hermelin. On the average-case complexity of parameterized clique. *Theor. Comput. Sci.*, 576:18–29, 2015. doi:10.1016/j.tcs.2015.01.042.

15 Etienne Grandjean. Complexity of the first-order theory of almost all finite structures. *Information and Control*, 57(2/3):180–204, 1983. doi:10.1016/S0019-9958(83)80043-6.

16 Martin Grohe. Generalized model-checking problems for first-order logic. In Afonso Ferreira and Horst Reichel, editors, *STACS 2001, 18th Annual Symposium on Theoretical Aspects of Computer Science, Dresden, Germany, February 15-17, 2001, Proceedings*, volume 2010 of *Lecture Notes in Computer Science*, pages 12–26. Springer, 2001. doi:10.1007/3-540-44693-1_2.

17 Yuri Gurevich. Complete and incomplete randomized NP problems. In *28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987*, pages 111–117. IEEE Computer Society, 1987. doi:10.1109/SFCS.1987.14.

18 Valentin F. Kolchin. *Random graphs*, volume 53 of *Encyclopedia of mathematics and its applications*. Cambridge University Press, 1999.

19 Leonid A. Levin. Problems, complete in “average” instance. In Richard A. DeMillo, editor, *Proceedings of the 16th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1984, Washington, DC, USA*, page 465. ACM, 1984. doi:10.1145/800057.808713.

20 Jerzy Łoś. On the categoricity in power of elementary deductive systems and some related problems. *Colloq. Math.*, 3:58–62, 1954.

21 Moni Naor, Leonard J. Schulman, and Aravind Srinivasan. Splitters and near-optimal derandomization. In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, USA, 23-25 October 1995*, pages 182–191. IEEE Computer Society, 1995. doi:10.1109/SFCS.1995.492475.

22 Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford University Press, 2006. doi:10.1093/ACPROF:OSO/9780198566076.001.0001.

23 Robert L. Vaught. Applications to the Löwenheim–Skolem–Tarski theorem to problems of completeness and decidability. *Indagationes Mathematicae*, 16:467–472, 1954.

24 John von Neumann. Various techniques used in connection with random digits. In Alston S. Householder, George E. Forsythe, and Hallett-Hunt Germond, editors, *Monte Carlo method. Proceedings of a Symposium Held June 29, 30 and July 1, 1949 in Los Angeles, California*, volume 12, pages 36–38, 1951. URL: https://dornsifecms.usc.edu/assets/sites/520/docs/VonNeumann-ams12p36-38.pdf.