Lower Bounds on Pure Dynamic Programming for Connectivity Problems on Graphs of Bounded Path-Width

Kacper Kluk
Institute of Informatics, University of Warsaw, Poland

Jesper Nederlof
Department of Information and Computing Sciences, Utrecht University, The Netherlands

Abstract

We give unconditional parameterized complexity lower bounds on pure dynamic programming algorithms – as modeled by *tropical circuits* – for connectivity problems such as the Traveling Salesperson Problem. Our lower bounds are higher than the currently fastest algorithms that rely on algebra and give evidence that these algebraic aspects are unavoidable for competitive worst case running times.

Specifically, we study input graphs with a small width parameter such as treewidth and pathwidth and show that for any k there exists a graph G of pathwidth at most k and $k^{\mathcal{O}(1)}$ vertices such that any tropical circuit calculating the optimal value of a Traveling Salesperson round tour uses at least $2^{\Omega(k \log \log k)}$ gates. We establish this result by linking tropical circuit complexity to the nondeterministic communication complexity of specific compatibility matrices. These matrices encode whether two partial solutions combine into a full solution, and Raz and Spieker [Combinatorica 1995] previously proved a lower bound for this complexity measure.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Parameterized Complexity, treewidth, tropical circuits, connectivity problems

Digital Object Identifier 10.4230/LIPIcs.ICALP.2026.130

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2512.23121`

Funding *Kacper Kluk*: Supported by Polish National Science Centre SONATA BIS-12 grant number 2022/46/E/ST6/00143.
Jesper Nederlof: supported by the project COALESCE that has received funding from the European Research Council (ERC), grant agreement No 853234.

Acknowledgements We thank Wim van den Broeck for making us aware of relations between our work and [13], and Daniel Dadush for making us aware of direction connections of tropical circuits and extension complexity as outlined in [20].

1 Introduction

A common paradigm in theoretical computer science is to pin down which algorithmic techniques can or cannot achieve certain central goals by formalizing such a technique within a precise algorithmic model and then proving sharp limitations for that model. Classic examples include unconditional lower bounds for Linear Programming-based algorithms (e.g. extension complexity [14, 37, 40]), conditional lower bounds for preprocessing algorithms in parameterized complexity (conditioned on the non-collapse of the polynomial hierarchy [6, 17]), and unconditional lower bounds for resolution-based algorithms for refuting the Strong Exponential Time Hypothesis [2]. This paradigm reveals which algorithmic nuances are fundamentally unavoidable and shows us where genuine breakthroughs must come from.

53rd International Colloquium on Automata, Languages, and Programming (ICALP 2026).
Editors: Sayan Bhattacharya, Danupon Nanongkai, Michael Benedikt, and Gabriele Puppis;
Article No. 130; pp. 130:1–130:23
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

One notable example of such algorithmic nuance is that of algebraic cancellation: for several combinatorial computational problems, the fastest known algorithms crucially exploit certain algebraic cancellation in rather counter-intuitive ways, and it is a well-studied open question whether these algorithms can be matched with (more natural, and in some sense robust) "combinatorial" algorithms. This question arises for instance in the settings of Boolean matrix multiplication (e.g. [1, 38]), various variants of the problem of finding a perfect matching solvable by reduction to determinant computation [24, 34], and fast exponential time algorithms for Hamiltonicity [4]. This very question of whether the use of algebraic cancellation is needed does not only occur in algorithm design, but also in various other disciplines. For example, in extremal combinatorics, it is an open question in various settings whether combinatorial proofs exist that match algebraic proofs (for example, a combinatorial proof of the skew two families theorem remains illusive [7]), and the log rank conjecture in communication complexity asks whether algebraic low rank decompositions of matrices in general imply low rank combinatorial decompositions (or more specifically, whether the rank and partioning number of a Boolean matrix are quasi-polynomially related), see e.g. [30, 31].

One such setting in which we would like to replace algebraic arguments with combinatorial arguments within the area of parameterized complexity is that of *connectivity problems parameterized by treewidth*: In 2011 [12] it was shown that many connectivity problems parameterized by treewidth can be solved faster than what was deemed possible at the time. In particular, problems like Hamiltonian Cycle and Steiner Tree were solved with a randomized algorithm in $2^{\mathcal{O}(k)}N^{\mathcal{O}(1)}$ time, when given a tree decomposition of width k of an N-vertex input graph. Before the work [12], the fastest known algorithm for problems parameterized by treewidth were only based on straightforward combinatorial dynamic programming techniques that naturally lead to running times like $k^{\mathcal{O}(k)}N^{\mathcal{O}(1)}$ and seemed hard to improve. Later work that followed up on [12] and provided deterministic algorithms for weighted extensions such as the Traveling Salesperson Problem (TSP) on bounded treewidth graphs [5] and an alternative version connected the approach with a matroid-based extension of the aforementioned two-family theorem [16]. But these newer algorithms are slower and typically do not lead to conditionally optimal running times due to the *overhead* implied by the need for computing a row-basis of an exponentially-sized matrix.

All these works [12, 5, 16] rely on algebraic cancellation by exploiting that the rank of certain *compatibility* matrices is small, a particularly counterintuitive approach in comparison to the previous simple and clean combinatorial $k^{\mathcal{O}(k)}N^{\mathcal{O}(1)}$ time algorithms. It is a notable open question to remove the mentioned overhead in the runtime for the deterministic algorithm and weighted extensions. For example, a positive resolution of this open question would bring us closer to solving the TSP in time 1.9999^N [35], where N is the number of cities.

But, to do so it seems crucial to get a better combinatorial understanding of the involved compatibility matrices, simultaneously bypassing the undesired algebraic cancellation arguments. In this work we give a strong indication that more than just elementary combinatorial techniques are needed, via the algorithmic model of *tropical circuits*.

Tropical Circuits. A tropical circuit is an arithmetic circuit in which the inputs are labeled with variables that take an integer as value and the two arithmetic gates correspond to the max and sum, or alternatively, min and sum, operations. See Section 2 for a precise definition. The motivation for studying the expressiveness of tropical circuits is that it models a broad class of dynamic programming algorithms. In particular, if there exists for a maximization problem (for which its instances are defined by a set of input weights) a dynamic programming algorithm that is "pure" in the sense that the associated recurrence only features the operations max and sum, then there is a tropical circuit that outputs the optimal objective value with circuit size being proportional to the running time of the dynamic programming algorithm.

For example, the classic Bellman-Held-Karp [3, 19] $\mathcal{O}(N^2 2^N)$ time algorithm for TSP is naturally converted into a tropical circuit with $\mathcal{O}(N^2 2^N)$ gates and the $\mathcal{O}(N^3)$ time Floyd-Warshall [15, 39] algorithm that computes the shortest path lengths between each pair of vertices is naturally converted into a tropical circuit with $\mathcal{O}(N^3)$ gates. Both tropical circuits cannot be substantially improved [21, 25] (see also [22, Corollary 2.2]).

In the realm of parameterized complexity, it is easy to see that canonical applications of dynamic programming can be modeled in an efficient way as tropical circuits. Examples are the dynamic programming algorithms for Steiner Tree and Set Cover (see e.g. [10, Section 6.1]) with few number of terminals and elements and, especially relevant for this paper, the $\mathcal{O}(2^k N)$ time algorithm for the maximum independent set problem on graphs with a given tree decomposition of width k and N vertices.

For much more details on tropical circuits, we refer to the excellent textbook by Jukna [22].

Problems parameterized by treewidth. A very popular research line that started in [28, 29] is that of investigating the fine-grained complexity of various NP-hard problems parameterized by width measures such as the treewidth of the input graph. In particular, for many NP-hard problems we are now able to

- design algorithms with a running time of the type $f(k)N^{\mathcal{O}(1)}$ for some width measure k and N denoting the number of vertices of the input graph, and
- prove that any improvement of this running time to $f(k)^{1-\Omega(1)}N^{\mathcal{O}(1)}$ or even $f(k)^{o(1)}N^{\mathcal{O}(1)}$ violates the Exponential Time Hypothesis (ETH) or the Strong Exponential Time Hypothesis (SETH).

Such algorithms are often called *(S)ETH-tight algorithms*. Such tight algorithms provide insight on how amenable the problem at hand is for divide and conquer techniques: conditioned on standard hypotheses, they essentially reveal how much information of partial solutions exactly is relevant in order to decide whether the partial solutions combine or not.

The aforementioned type of connectivity problems (such as TSP and Steiner Tree) forms an important class of problems for which we do not generally have SETH-tight algorithms, because of the aforementioned overhead implied by the algebraic methods. A natural question is hence, whether this algebraic bottleneck can be avoided and whether we can replace the algebraic arguments with combinatorial ones.

Our results

In this paper we provide evidence that direct combinatorial techniques on their own are insufficient for designing faster algorithms for connectivity problems, by giving unconditional lower bounds for tropical circuits. We state our lower bounds in terms of the *pathwidth* of the input graph. This is similar to the treewidth of a graph, except that we require more specifically to decompose the graph in a path-like manner instead of a tree-like manner. Hence, the pathwidth of a graph is always at least the treewidth of a graph and hence our lower bounds also imply lower bounds parameterized by treewidth. See Section 2 for definitions. Before we study connectivity problems, we first study the complexity of a more basic problem:

Maximum Weight Independent Set. In the Maximum Weight Independent Set problem one is given a graph $G = (V, E)$ along with a vertex weight x_v for every $v \in V$, and is asked for the value $IS_G := \max_{I \in \mathcal{I}(G)} \sum_{v \in I} x_v$ where $\mathcal{I}(G)$ denotes the family of all independent sets of G. Our lower bound for this problem reads as follows (full definitions are postponed to Section 2):

▶ **Theorem 1.1.** *For any $k \geq 1$, there exists a graph G of pathwidth at most k on $k^{\mathcal{O}(1)}$ vertices such that any tropical circuit calculating IS_G uses at least $\Omega(2^k)$ gates.*

Note that pathwidth is a width measure that is always at least the treewidth of the graph. Since there is a simple $\mathcal{O}(2^k N)$-sized tropical circuit that calculates IS_G for an N-vertex graph of pathwidth k, this result is optimal in a tight sense.[1] A previous result by Korhonen [27] showed that for *every* graph G of treewidth k and maximum degree d any tropical circuit calculating IS_G must be of size at least $2^{\Omega(k/d)}$. This result is less tight than our new result and also seems to crucially rely on some properties of the Maximum Weight Independent Set problem. While the type of universal lower bound from [27] is quite interesting, it does not directly have added value in our context of worst case complexity analysis and hence we do not pursue it further in this work.

Theorem 1.1 is obtained using the following two ingredients. The graphs for which we show the bounds are constructed based on ideas from the classic reduction from CNF-Sat to Maximum Weight Independent Set. The tropical circuit size bound itself is shown by analyzing the structure of the so-called rectangles, a combinatorial notion that typically arises in the study of tropical circuits that captures the way in which the partial solution calculated at some node of the tropical circuit can combine with the computations done by the rest of the circuit, see e.g. [22].

Connectivity Problems. We show lower bounds on the tropical circuit complexity of the following graph connectivity problems. Let G be an N-vertex graph with an edge weight $x_{u,v} \in \mathbb{N}$ for every pair of distinct vertices u and v. Let $\mathcal{H}(G)$ denote the family of all sequences of $N+1$ vertices $(u_i)_{i=0}^N$ such that $u_0 = u_N$ and $u_0 \to u_1 \to \cdots \to u_{N-1} \to u_N$ is a directed Hamiltonian cycle in G. We define $DTSP_G$ to be the minimum $\sum_{i \in [N]} x_{u_{i-1},u_i}$ taken over all $(u_0, \ldots, u_N) \in \mathcal{H}(G)$. For an undirected graph, we define the undirected variant TSP_G similarly while identifying variables $x_{u,v}$ and $x_{v,u}$ for all $uv \in E(G)$.

With these definitions in place, our main results can be stated as follows:

▶ **Theorem 1.2.** *For any $k \geq 1$, there exists a graph G of pathwidth at most k on $k^{\mathcal{O}(1)}$ vertices such that any tropical circuit calculating $DTSP_G$ uses at least $2^{\Omega(k \log \log k)}$ gates.*

▶ **Theorem 1.3.** *For any $k \geq 1$, there exists a graph G of pathwidth at most k on $k^{\mathcal{O}(1)}$ vertices such that any tropical circuit calculating TSP_G uses at least $2^{\Omega(k \log \log k)}$ gates.*

We also provide a similar lower bound for the Directed Spanning Tree problem. We let $\mathcal{T}(G)$ denote the family of all functions $p : V(G) \to V(G)$ such that for exactly one vertex $v \in V(G)$, $p(v) = v$, and edges $(p(u), u)$ over all other vertices u form an out-tree rooted at v (i.e. a tree oriented away from the root). Furthermore, we define DST_G as the minimum $\sum_{v \in V(G): p(v) \neq v} x_{p(v),v}$ taken over all $p \in \mathcal{T}(G)$ (i.e., DST_G is the minimum weight spanning tree of G).

[1] This can be shown by translating the standard $\mathcal{O}(2^k N)$ time algorithm for IS_G (see e.g. [10, Section 7.3.1]) in the natural way into a tropical circuit.

▶ **Theorem 1.4.** *For any $k \geq 1$, there exists a graph G of pathwidth at most k on $k^{\mathcal{O}(1)}$ vertices such that any tropical circuit calculating DST_G uses at least $2^{\Omega(k \log\log k)}$ gates.*

By adapting textbook dynamic programming algorithms, one can directly obtain tropical circuits for $DTSP_G, TSP_G$ and DST_G of size $k^{\mathcal{O}(k)}N$ if G has N vertices and pathwidth/treewidth k. Thus, there is still a gap between the lower bound and upper bound, and in fact this seems closely related to a similar gap in the area of communication complexity (see Section 7 for more details on this). Nevertheless, our results show that the currently fastest $2^{\mathcal{O}(k)}N$ time algorithms for TSP and Directed Steiner Tree (which generalizes Directed Spanning Tree) cannot be matched with merely pure dynamic programming.

The proofs of these three results combine the basic ingredients of the proof of Theorem 1.1 with the properties of the *Matchings Compatibility Matrix*, defined as follows. The rows and columns of this binary matrix are both indexed by perfect matchings of a bipartite graph, and an entry in the matrix indicates whether the union of these two perfect matchings forms a Hamiltonian cycle. This matrix has already been studied in [36] in the context of the log-rank conjecture in communication complexity, where a lower bound on the nondeterministic communication complexity and an upper bound of its rank was given. The rank of (a slight variant of) this matrix also turned out to be important for studying the complexity of the Hamiltonian Cycle and TSP problem, both parameterized by the path/treewidth of the input graph and parameterized by the number of vertices [9, 11, 35].

Our main technical contribution is that the lower bound on the nondeterministic communication complexity of this matrix from [36] can be used to obtain the above lower bounds on the size of tropical circuits.

Extension Complexity. A notion closely related to minimum tropical circuit size of a polynomial is that of the *extension complexity* of a polytope.[2] The extension complexity $\mathrm{xc}(P)$ of a polytope P is the minimum number of facets of a (possibly, higher-dimensional) polytope that can be projected to P. The notion has been introduced by Yannakakis [40] and culminated in celebrated exponential lower bounds of the polytope associated with TSP_G [14] and the non-bipartite matching polytope [37]. These two (Gödel-prize winning) papers also achieve their lower bound by lower bounding the nondeterministic communication complexity of a certain matrix, but we use a completely different matrix.

While it is known that dynamic programming algorithms can usually be turned into extension complexity upper bounds [8, 33], of course many problems amenable to linear programming cannot be solved directly with dynamic programming. Analogously, the spanning tree polynomial (denoted with DST_G in this paper) cannot be computed with tropical circuits of size $2^{o(\sqrt{n})}$ (see [23]), but has polynomial extension complexity (see e.g. [32]). In the other direction, it is known that the existence of small *monotone neural networks* implies small extension complexity (see [20]). In particular, since tropical circuits are a special case of such networks, for any multilinear polynomial f and its corresponding polytope P, every tropical circuit calculating f must have size at least $\frac{1}{2}\mathrm{xc}(P)$.

The extension complexity of polytopes in terms of treewidth of the input graph was already upper bounded in previous works [26, 13]. In [13] the authors presented lower bounds, but those are conditioned on the Exponential Time Hypothesis. The currently (to our best knowledge) highest relevant unconditional lower bounds are a $2^{\Omega(n/\log n)}$ lower bound on the

[2] To compare the two notions, we focus on multilinear polynomials, i.e. polynomials of maximum individual degree 1. With such a polynomial we can associate a polytope which is the convex hull of all points $(a_1, \ldots, a_n) \in \{0,1\}^n$ with the property that $x^{a_1} \ldots x^{a_n}$ is a monomial in the polynomial.

extension complexity of the independent set polytope (corresponding to IS_G) from [18], and a $2^{\Omega(n)}$ lower bound on the extension complexity of the TSP polytope (corresponding to TSP_G) from [37].

These lower bounds are incomparable to our lower bounds in Theorem 1.1 and Theorem 1.2. We hope that our techniques can be used to prove stronger lower bounds on the extension complexity of polytopes (as a function of the treewidth of the input graph, or in general).

Organization. This paper is organized as follows: In Section 2 we provide the necessary definitions and preliminary tools used in the remainder of the paper. Section 3 presents the proof of Theorem 1.1. Section 4 presents the aforementioned matchings compatibility matrix and the required results about its structure. In Section 5 we prove Theorem 1.2 and Theorem 1.3. In Section 6 we prove Theorem 1.4, and we provide some concluding remarks in Section 7.

2 Preliminaries

We use the following basic notation. For a positive integer ℓ, we use $[\ell]$ to denote the set $\{1, 2, \ldots, \ell\}$. For a graph G, we use $V(G)$ to denote the set of vertices of G, and $E(G)$ to denote the set of edges of G. For a graph G and a subset $X \subseteq V(G)$, we write $G[X]$ to denote the subgraph of G induced by X, i.e., a graph with vertices restricted to X and all edges between vertices of X preserved.

By $\mathcal{S}_k$ we denote the group of all permutations on the set $[k]$. By $\bar{\mathcal{S}}_k$ we denote the subset of $\mathcal{S}_k$ of permutations that contain exactly one cycle. By $\mathcal{S}^2_{2k}$ we denote the subset of $\mathcal{S}_{2k}$ of permutations that contain exactly k cycles, each of size 2 and by $\mathcal{S}^k_{2k}$ we denote the subset of $\mathcal{S}_{2k}$ of permutations that contain exactly 2 cycles, each of size k.

A *cycle type* of a permutation is defined as the multiset of the sizes of all its cycles. A *conjugation* of a permutation $\rho \in \mathcal{S}_k$ by a permutation $\pi \in \mathcal{S}_k$ is defined as $\pi^{-1}\rho\pi$, i.e. the composition of the inverse of π, ρ and π. It is well known that permutation conjugation preserves its cycle type, and hence $\bar{\mathcal{S}}_k, \mathcal{S}^2_{2k}, \mathcal{S}^k_{2k}$ are closed under taking conjugations. We will need the following group-theoretic properties of permutations and their conjugations. All of those are either well known or easy to obtain with some basic calculations done on subgroups.

▶ **Proposition 2.1.** *The following equalities hold for any $k \geq 1$ and for any maximal set $\mathcal{Z}_k \subseteq \mathcal{S}_k$ of permutations of the same cycle type.*

1. $|\bar{\mathcal{S}}_k| = (k-1)!$, $|\mathcal{S}^k_{2k}| = (2k-1)!/k$, $|\mathcal{S}^2_{2k}| = (2k-1)!! = (2k-1)(2k-3)\cdots 3 \cdot 1$,
2. $\{\pi^{-1}\rho\pi \mid \pi \in \mathcal{S}_k\} = \mathcal{Z}_k$ *for every* $\rho \in \mathcal{Z}_k$,
3. $|\{\pi \in \mathcal{S}_k \mid \pi^{-1}\rho_1\pi = \rho_2\}| = |\mathcal{S}_k|/|\mathcal{Z}_k|$ *for every* $\rho_1, \rho_2 \in \mathcal{Z}_k$,
4. $|\{\rho_2 \in \mathcal{S}^2_{2k} \mid \rho_2\rho_1 \in \mathcal{S}^k_{2k}\}| = (2k-2)!! = (2k-2)(2k-4)\cdots 4 \cdot 2$ *for every* $\rho_1 \in \mathcal{S}^2_{2k}$.

Pathwidth. Let G be a graph (undirected or directed). A *path decomposition* of G is a sequence of subsets $\beta_1, \beta_2, \ldots, \beta_\ell \subseteq V(G)$ called *bags* such that:

- for every edge $(u, v) \in E(G)$, there is some $i \in [\ell]$ such that $u, v \in \beta_i$, and
- for every vertex $v \in V(G)$, there exist $1 \leq i_1 \leq i_2 \leq \ell$ such that $\{i \mid v \in \beta_i\} = \{i_1, i_1+1, \ldots, i_2-1, i_2\}$, i.e., the bags containing v form a connected subinterval of the sequence of all bags.

The *width* of a path decomposition $(\beta_i)_{i=1}^{\ell}$ is defined as $\max_{i\in[\ell]} |\beta_i| - 1$. The *pathwidth* of G is defined as the minimum possible width of a path decomposition of G.

2.1 Tropical polynomials

We treat all polynomials as defined over the tropical $(\max, +)$ semiring, i.e., $(f \cdot g)(x) = f(x) + g(x)$ and $(f + g)(x) = \max(f(x), g(x))$ for any two polynomials f, g.

Independent Set polynomial. If we treat the weights x_v as indeterminates we can view IS_G as a polynomial in the $(\max, +)$ semiring by replacing the max operation by addition and the $+$ operation by multiplication:

$$IS_G := \sum_{I \in \mathcal{I}(G)} \left(\prod_{v \in I} x_v \right),$$

where we remind the reader that $\mathcal{I}(G)$ denotes the family of all independent sets of G.

Traveling Salesperson Problem polynomial. Similarly to the Independent Set polynomial IS_G, we define the directed TSP polynomial of G as

$$DTSP_G := \sum_{(u_0,\ldots,u_N) \in \mathcal{H}(G)} \left(\prod_{i \in [N]} x_{u_{i-1},u_i} \right),$$

where we remind the reader that $\mathcal{H}(G)$ denotes the family of all sequences of $N+1$ vertices $(u_i)_{i=0}^N$ such that $u_0 = u_N$ and $u_0 \to u_1 \to \cdots \to u_{N-1} \to u_N$ is a directed Hamiltonian cycle in G. Naturally, we also view TSP_G similarly as a polynomial by identifying variables $x_{u,v}$ and $x_{v,u}$ for all $uv \in E(G)$.

Directed Spanning Tree polynomial. We also define a directed spanning tree polynomial of G as

$$DST_G := \sum_{p \in \mathcal{T}(G)} \left(\prod_{v \in V(G): p(v) \neq v} x_{p(v),v} \right)$$

where we remind the reader that $\mathcal{T}(G)$ denotes the family of all functions $p : V(G) \to V(G)$ such that for exactly one vertex $v \in V(G)$, $p(v) = v$, and edges $(p(u), u)$ over all other vertices u form an out-tree rooted at v.

The *support* of a monomial $m = x_{i_1}^{a_1} \ldots x_{i_\ell}^{a_\ell}$ is the set of variables $\{x_{i_1}, \ldots, x_{i_\ell}\}$ and is denoted as $\sup(m)$. The *support of a polynomial* $p = m_1 + \cdots + m_\ell$ is the union of supports $\sup(m_i)$ over all $i \in [\ell]$. We let $\sup(p)$ denote the support of polynomial p. For a monomial m and a polynomial p, we write $m \in p$ if p treated as a formal expression is of the form $m + q$ for some polynomial q. We say that a polynomial p is *homogeneous* if and only if for some $d \in \mathbb{N}$, the degree of all monomials $m \in p$ is d. For a polynomial p, we write $|p|$ to denote the number of monomials m such that $m \in p$.

A *valuation* is a function which maps variables to $\mathbb{R}$. A *characteristic valuation* χ_m of a monomial m is a valuation such that $\chi_m(x) = 1$ if $x \in \sup(m)$ and $\chi_m(x) = -1$ otherwise. As stated before, we evaluate polynomials in a $(\max, +)$ semiring, i.e., given a valuation v, the monomial x evaluates to $v(x)$, $(f + g)(v)$ evaluates to $\max(f(v), g(v))$ and $(f \cdot g)(v)$ evaluates to $f(v) + g(v)$.

For two polynomials p, q, we write $p \subseteq q$ if for every monomial $m \in p$, we have $m \in q$. We write $p \simeq q$ iff $p \subseteq q$ and $q \subseteq p$. In particular, $p \simeq q$ does not imply that $p = q$ (take, e.g. $p = x$ and $q = x + x$). The definition of $\simeq$ is motivated by the following observation.

▶ **Proposition 2.2.** *For any two multilinear polynomials f, g, we have $f \simeq g$ if and only if for any valuation v, we have $f(v) = g(v)$.*

Proof. The implication from $f \simeq g$ to $f(v) = g(v)$ is immediate. In the other direction, assume w.l.o.g. that $f \subseteq g$ does not hold, and hence there is some monomial $m \in f$ such that $m \notin g$. It is easy to see that $f(\chi_m) = |\sup(m)|$ and $g(\chi_m) < |\sup(m)|$, which is a contradiction. ◀

2.2 Tropical circuits

A *tropical circuit* is a directed acyclic graph in which every vertex (called *node*) is of in-degree 0 or 2. Vertices with in-degree 0 are labeled with either a variable, in which case they are called *input nodes*, or with a constant 0, in which case they are called *constant nodes*. The nodes with in-degree 2 are called *operation nodes* and are labeled with a binary operator, either plus or max. There is one node, designated as the *output node*.

Evaluation of a tropical circuit given a valuation v is defined the following way. The nodes are processed according to the topological order of the graph. Constant nodes evaluate to value 0. Input nodes labeled x evaluate to $v(x)$. Operation nodes labeled with, respectively, plus and max evaluate to, respectively, sum and maximum of the values of its two predecessors. The output of the evaluation of the whole circuit is the value obtained at the output node.

We say that a tropical circuit Γ *calculates some polynomial* p, if the evaluation of Γ on v is equal to $p(v)$ for every valuation of variables v. It is easy to see that semantically, evaluating a tropical circuit computes exactly some polynomial: the input node labeled x computes x, the max nodes compute the sum of two polynomials, and the plus nodes compute the product of two polynomials.

By Proposition 2.2 and the definitions, we immediately get the following.

▶ **Proposition 2.3.** *Let f, g be two multilinear polynomials and let Γ be a tropical circuit calculating f. Then Γ calculates g iff $f \simeq g$.*

Intuitively, this claim states that a tropical circuit calculates some polynomial iff the set of its monomials is exactly the set of all monomials which appear during the evaluation of said circuit.

The main combinatorial ingredient regarding tropical circuits we will use is the following decomposition lemma. For the proof, see, e.g., [22, Lemma 3.4].

▶ **Lemma 2.4.** *Let f be a homogeneous polynomial calculated by a tropical circuit of size τ and let X denote an arbitrary subset of $\sup(f)$. Then f can be written as*

$$f \simeq \sum_{i \in [\tau]} g_i \cdot h_i,$$

where $|\sup(g_i) \cap X|, |\sup(h_i) \cap X| \leq \frac{2}{3}|X|$ for each $i \in [\tau]$.

Note that the general definition of tropical circuits allows for constants other than 0 to appear in the circuit. We can omit them as such constants do not appear in the polynomials we consider. In fact, we only need constant 0 to deal with the empty independent set in IS_G, which can be thought of as a monomial of size 0.

3 Independent Set

In this section, we show the bound of Theorem 1.1. The class of graphs we use is inspired by the classic reduction from CNF-Sat to Maximum Weight Independent Set. Fix $k > 0$ and let $q = 4\binom{k}{2}$. Consider all possible 2-CNF clauses on k variables: there are exactly q of those.

We can think of each clause as a tuple (a, b, n^a, n^b) where $a, b \in [k]$ represents an unordered pair of variables and $n^a, n^b \in \{0, 1\}$ represents whether a-th and b-th variable is negated in the clause. We define the undirected graph G_k in the following way. We put

$$V(G_k) = \{v_{i,j} \mid i \in [k], j \in \{0, \dots, 2q-1\}\} \cup \{w_j \mid j \in [q]\}.$$

The vertices $v_{i,j}$ represent the literals of the CNF formula: i denotes the variable, and the parity of j determines whether it is negated or not. Every literal has q copies, one for each clause. The vertices w_j represent all possible 2-CNF clauses. For the sake of convenience, we will use w_{a,b,n^a,n^b} to denote the vertex corresponding to the clause represented by a, b, n^a, n^b as described above. We put $V_{i,r} = \{v_{i,2j+r} \mid j \in \{0, \dots q-1\}\}$ for $i \in [k], r \in \{0, 1\}$, that is, even and odd vertices representing the i-th variable. We also put $V_i = V_{i,0} \cup V_{i,1}$.

We put the edges accordingly:

$$E(G_k) = \{(v_{i,j-1}, v_{i,j}) \mid i \in [k], j \in [2q-1]\} \cup \{(w_i, l_{i,j}) \mid i \in [q], j \in \{1, 2\}\}$$

where $l_{i,1} = v_{a_i,2(i-1)+n_i^a}$ and $l_{i,2} = v_{b_i,2(i-1)+n_i^b}$ where $w_i = w_{a_i,b_i,n_i^a,n_i^b}$. That is, $l_{i,j}$ correspond to the literals of the clause represented by w_i. Because there is a copy $v_{i,j}$ of each literal for every clause, every $v_{i,j}$ is connected at most to one vertex $w_{1+\lfloor j/2 \rfloor}$. The degree of each w_i is exactly 2.

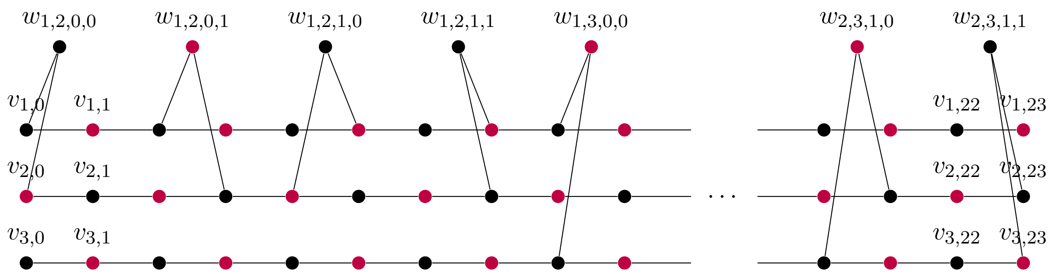


Figure 1 The graph G_3. Red vertices belong to the canonical solution for valuation $\rho(1) = 1, \rho(2) = 0, \rho(3) = 1$.

It is easy to see that the graph G_k has pathwidth at most $k+1$: We start with a natural path decomposition of width k of a $k \times 2q$ grid of which $G_k[V_1 \cup \dots \cup V_k]$ is a subgraph. The bags containing the neighbours of the vertices w_i form a pairwise disjoint subintervals of the decomposition, so w_i can be added to these bags naturally, resulting in a decomposition of G_k of width $k+1$. Proving the following lemma will immediately show Theorem 1.1.

▶ **Lemma 3.1.** *For any $k \geq 1$, any tropical circuit calculating IS_{G_k} uses at least $2^k/3$ gates.*

The rest of this section is dedicated to proving this lemma. Fix $k > 0$. The proof will be done in two steps. First, we will show a combinatorial structure of so-called rectangles in our described graph. Then, we will relate this notion to the polynomials calculated by tropical circuits and show the desired lower bound as a consequence.

The *canonical solution I_ρ given by the assignment $\rho : [k] \to \{0, 1\}$* is defined as

$$I_\rho = \{v_{i,2j+\rho(i)} \mid i \in [k], j \in \{0, \dots, q-1\}\} \cup \{w_i \mid \rho(a_i) \neq n_i^a \text{ and } \rho(b_i) \neq n_i^b\}.$$

That is, we pick vertices $v_{i,j}$ for all j even or all j odd depending on $\rho(i)$ and include all vertices w_i corresponding to clauses not satisfied by assignment ρ (see Figure 1 for an example). Clearly, there are 2^k canonical solutions, one per each assignment ρ, and every canonical solution is an independent set of G_k.

We will say that a pair $\mathcal{A}, \mathcal{B} \subseteq 2^{V(G)}$ of families of independent sets of G forms a *rectangle* in G (denoted $\mathcal{A} \cdot \mathcal{B}$) if and only if every pair of sets $A \in \mathcal{A}, B \in \mathcal{B}$ are disjoint and non-adjacent (in particular, $A \cup B$ forms an independent set in G). We refer to the families $\mathcal{A}, \mathcal{B}$ as *sides* of the rectangle. We will say that a rectangle $\mathcal{A} \cdot \mathcal{B}$ *contains* an independent set I if $I = A \cup B$ for some $A \in \mathcal{A}, B \in \mathcal{B}$. We will say that a set $A \in \mathcal{A}$ is *useful* if there exists $B \in \mathcal{B}$ such that $A \cup B$ is a canonical solution. We define a set $B \in \mathcal{B}$ being useful in an analogous way.

For simplicity, the notion of a rectangle introduced here is defined in terms of families of independent sets of G. This concept, however, will be crucial in showing all of the subsequent bounds, hence, in later sections we will redefine rectangles in terms of tropical polynomials, as in the statement of Lemma 2.4, in order for the definition to be more general.

3.1 Thin rectangles

The key combinatorial property of the circuits calculating IS_{G_k} is expressed through the following lemma.

▶ **Lemma 3.2.** *Let $\mathcal{A} \cdot \mathcal{B}$ be a rectangle in G_k that contains at least one canonical solution I_ρ. Then, either $\mathcal{A}$ or $\mathcal{B}$ contain at most one useful set.*

For the rest of this subsection, we fix a rectangle $\mathcal{A} \cdot \mathcal{B}$, and focus on proving the lemma via the following series of claims.

▷ Claim 3.3. Let I_1, I_2 be two different useful sets belonging to the same side of $\mathcal{A} \cdot \mathcal{B}$. Then, there exists $i \in [k], j \in \{0, \dots, 2q-1\}$ such that $v_{i,j}$ belongs to exactly one of I_1, I_2.

Proof. W.l.o.g. we assume that $I_1, I_2 \in \mathcal{A}$. If both sets have different intersection with some V_i, then the claim trivially follows, hence w.l.o.g. we assume that for some $l \in [q]$, we have $w_l \in I_1$ and $w_l \notin I_2$.

Since I_2 is useful, we have $J_2 \in \mathcal{B}$ such that $I_2 \cup J_2$ is a canonical solution. Since I_1 and J_2 are disjoint, we have $w_l \notin I_2 \cup J_2$, hence $v_{i,j} \in I_2 \cup J_2$ for some neighbor $v_{i,j}$ of w_l. As I_1 and J_2 are non-adjacent, we have $v_{i,j} \notin J_2$, hence $v_{i,j} \in I_2$. As I_1 is an independent set of G_k, we have $v_{i,j} \notin I_1$, which finishes the proof of the claim. ◁

▷ Claim 3.4. Let I_1, I_2 be two different useful sets belonging to the same side of $\mathcal{A} \cdot \mathcal{B}$. Then, there exists $i \in [k], r \in \{0, 1\}$ such that $V_{i,r} \subseteq I_1$ and $V_{i,1-r} \subseteq I_2$.

Proof. Assume w.l.o.g. that $I_1, I_2 \in \mathcal{A}$. By the previous claim, we have some $i \in [k], j \in \{0, \dots, 2q-1\}$ such that w.l.o.g. $v_{i,j} \in I_1$ and $v_{i,j} \notin I_2$. Put $r = j \bmod 2$. Since I_1, I_2 are useful, we have the corresponding $J_1, J_2 \in \mathcal{B}$. By contradiction, let $j' \in \{0, \dots, 2q-1\}$ be an index minimizing $|j - j'|$ for which $v_{i,j'}$ does not belong to the expected set, i.e., such that either $j' \bmod 2 = r$ and $v_{i,j'} \notin I_1$ or $j' \bmod 2 \neq r$ and $v_{i,j'} \notin I_2$.

Consider the case $j' \bmod 2 = r$, the other one will be symmetric. As $v_{i,j} \in I_1 \cup J_1$, we also have $v_{i,j'} \in I_1 \cup J_1$. By minimality, we have that either $v_{i,j'+1}$ or $v_{i,j'-1}$ is in I_2. If $v_{i,j'} \notin I_1$ then $v_{i,j'} \in J_1$, but this would imply that $I_2 \cup J_1$ is not an independent set, which is a contradiction. ◁

▷ Claim 3.5. Let $I_1, I_2 \in \mathcal{A}$ be two different useful sets and let $J_1 \in \mathcal{B}$ be such that $I_1 \cup J_1$ is a canonical solution. Then J_1 is the only useful set in $\mathcal{B}$.

Proof. Assume by contradiction that we have useful $J_2 \in \mathcal{B}$ different to J_1. Applying the previous claim to both sides, we obtain $i_A, i_B \in [k]$ and $r_A, r_B \in \{0, 1\}$ such that: $V_{i_A, r_A} \subseteq I_1, V_{i_A, 1-r_A} \subseteq I_2, V_{i_B, r_B} \subseteq J_1$ and $V_{i_B, 1-r_B} \subseteq J_2$. Naturally, we have $i_A \neq i_B$.

Pick $w = w_{i_A,i_B,1-r_A,1-r_B}$. That is, for some $j \in \{0,\dots,q-1\}$, the neighbors of w are exactly $v_A = v_{i_A,2j+1-r_A}$ and $v_B = v_{i_B,2j+1-r_B}$. We have $v_A \in I_2$ and $v_B \in J_2$. Since $V_{i_A,r_A} \subseteq I_1$ and I_1 is independent in G_k, we have $v_A \notin I_1$. Similarly, $v_B \notin J_1$. Since I_2 and J_1 are disjoint, we have $v_A \notin J_1$, and similarly $v_B \notin I_1$. Since I_2 and J_1 are not adjacent, we have $w \notin J_1$, and similarly $w \notin I_1$. However, this means that $I_1 \cup J_1$ is not a canonical solution since neither w nor its two neighbors belong to $I_1 \cup J_1$. This is a contradiction, hence the claim is proven. $\lhd$

Lemma 3.2 follows immediately from the last claim.

3.2 Lower bound on thin rectangle circuits

In this subsection we prove Theorem 1.1. At this point we can abstract away from the exact structure of G_k. The proof will depend only on the thin rectangle property proven in the previous subsection. The proof will closely follow the proof of [22, Lemma 2.18].

Let Γ be any tropical circuit that calculates IS_{G_k} and put $G := G_k$. For any node w of Γ, we define its *below* B_w as the polynomial calculated by Γ if we designate w to be its output node. Intuitively, B_w captures the contribution of the subcircuit rooted at w to the output of the whole calculation.

In a similar spirit, we would like to define the *above* of a node w as the contribution of the remainder of the circuit. Note that B_w does not need to be contained in IS_G, however, assuming the output node is reachable from w, for at least one monomial m, we have $m \cdot B_w \subseteq IS_G$. We will define A_w as the sum of all such monomials, that is

$$A_w = \sum_{m : m \cdot B_w \subseteq IS_G} m.$$

Naturally, $A_w \cdot B_w \subseteq IS_G$ for every node w from which the output node is reachable.

Every monomial of each A_w, B_w represents an independent set in G. Let A^*_w denote the useful monomials of A_w, that is, monomials $m \in A^*_w$ such that there exist $m' \in B_w$ for which $m \cdot m'$ represents a canonical solution. We define B^*_w analogously. As proven by Lemma 3.2, for every node w, one of A^*_w, B^*_w must be of size at most 1. If $|A^*_w| \leq 1$, we will say that $A_w \cdot B_w$ is *A-thin*, otherwise, we will say that it is *B-thin*.

If w is an output node, then we have $A_w = \{1\}, B_w = IS_G$, so $A_w \cdot B_w$ is A-thin. If w is an input node labeled x, we have $B_w = \{x\}$, therefore $A_w \cdot B_w$ is B-thin. Let $w_\ell, \dots, w_1$ be any path in Γ where w_1 is an output node and w_ℓ is some input node. Based on our observations, there is some edge on this path (w_{i+1}, w_i) such that w_i is A-thin and w_{i+1} is B-thin, and so $|A^*_{w_i} \cdot B^*_{w_{i+1}}| \leq 1$.

Let IS^*_G be the polynomial containing all monomials of IS_G representing a canonical solution. The following claim shows that all canonical solutions belonging to the rectangles described above can be propagated along such input to output paths of Γ. We will use the term predecessor to refer to in-neighbors of a node of the circuit.

$\rhd$ Claim 3.6. Let w be a node of Γ and let $c \in IS^*_G$ be such that $c = a \cdot b$ for some $a \in A^*_w, b \in B^*_w$. Then, for at least one predecessor u of w, we have $c \in A^*_u \cdot B^*_u$. Moreover, if w is a plus gate, then the above holds for both predecessors of w.

Proof. Let u, v be the predecessors of w. First, consider the case where w is a max gate, that is, $B_w = B_u + B_v$. Then, either $b \in B_u$ or $b \in B_v$. W.l.o.g. assume the former. We have $a \cdot B_u \subseteq a \cdot B_w \subseteq IS_G$, hence $a \in A_u$. Therefore, $b \in B^*_u$, $a \in A^*_u$, and hence $c \in A^*_u \cdot B^*_u$ as desired.

Second, consider the case where w is a plus gate, that is, $B_w = B_u \cdot B_v$. Then, we have $b = b_v \cdot b_u$ for some $b_v \in B_v$ and $b_u \in B_u$. We have $a \cdot b_v \cdot B_u \subseteq a \cdot B_v \cdot B_u \subseteq IS_G$, hence $a \cdot b_v \in A_u$. Therefore, $b_u \in B_u^*$, $a \cdot b_v \in A_u^*$, and hence $c \in A_u^* \cdot B_u^*$. By a symmetric argument, $c \in A_v^* \cdot B_v^*$. ◁

Now, consider the following process. We fix a canonical solution $c \in IS_G^*$ and start at output node w_1. If we are currently in a max node w_i, we move towards the predecessor w_{i+1} given by the claim. If we are currently in a plus node, we move towards the predecessor w_{i+1} for which $|B_{w_{i+1}}^*|$ is larger. Such process terminates at an input node and produces a path $w_\ell, \dots, w_1$. As argued before, for some $i \in [\ell - 1]$, we have $|A_{w_i}^* \cdot B_{w_{i+1}}^*| \leq 1$.

We have $c \in A_{w_{i+1}}^* \cdot B_{w_{i+1}}^*$. If w_i is a max node, we have $B_{w_{i+1}} \subseteq B_{w_i}$, hence $A_{w_{i+1}}^* \subseteq A_{w_i}^*$, hence $c \in A_{w_i}^* \cdot B_{w_{i+1}}^*$, and hence $A_{w_i}^* \cdot B_{w_{i+1}}^* = \{c\}$.

If w_i is a plus node, then we additionally have $c \in A_v^* \cdot B_v^*$ where v is the predecessor of w_i other than w_{i+1}. Given the way we chose w_{i+1}, we have $|B_v^*| \leq |B_{w_{i+1}}^*| \leq 1$. Since $B_{w_i}^* \subseteq B_{w_{i+1}}^* \cdot B_v^*$, we have $|B_{w_i}^*| \leq 1$. Thus, $A_{w_i}^* \cdot B_{w_i}^* = \{c\}$.

Repeating this process for each canonical c creates a mapping from IS_G^* to the set $V(\Gamma) \cup E(\Gamma)$. If we map c to a vertex w, then we have a guarantee that $A_w^* \cdot B_w^* = \{c\}$. If we map c to an edge (u, w), we have a guarantee that $A_w^* \cdot B_u^* = \{c\}$. This implies that the mapping is injective, and hence $2^k = |IS_G^*| \leq |V(\Gamma)| + |E(\Gamma)| \leq 3|V(\Gamma)|$. This finishes the proof of Lemma 3.1, which consequently proves Theorem 1.1

4 Matching compatibility matrix

In the following section, a matrix is a function from $I \times J$ to an arbitrary value set, where I and J are some sets of indices of, respectively, rows and columns of the matrix. We do not require I, J to be a set of form $[n]$ for some $n \in \mathbb{N}$. All matrices considered in this section have values in the set $\{0, 1\}$.

▶ **Definition 4.1.** *For a 0-1 matrix M with row indices I and column indices J, we say that a pair $I' \subseteq I, J' \subseteq J$ forms a* rectangle *of M if and only if it induces an all-ones submatrix of M, i.e., if and only if $M_{i,j} = 1$ for each $i \in I', j \in J'$. The* size *of a rectangle R is defined as $|I'| \cdot |J'|$ and denoted as $|R|$.*

▶ **Definition 4.2.** *A* rectangle cover *of a 0-1 matrix M is a set of rectangles $(I_1, J_1), \dots, (I_s, J_s)$ which cover all ones of M, i.e., such that for each i, j with $M_{i,j} = 1$, there exists $p \in [s]$ such that $(i, j) \in I_p \times J_p$. The* size *of the cover is the number of rectangles s.*

To place Definition 4.2 into context, let us stress that the nondeterministic communication complexity of a matrix M is defined as $\log_2 s$, where s is the minimum size of a rectangle cover of M.

4.1 Complete bipartite graphs

▶ **Definition 4.3.** *A matching compatibility matrix $\mathcal{M}_k$ of order k is a binary matrix of size $k! \times k!$ with rows and columns indexed by permutations $\mathcal{S}_k$ which satisfies*

$$\mathcal{M}_k(\rho_1, \rho_2) = 1 \quad \text{if and only if} \quad \rho_2\rho_1 \in \bar{\mathcal{S}}_k.$$

By C_k we denote the size of the smallest rectangle cover of $\mathcal{M}_k$.

Our bounds on sizes of tropical circuits will be based on the following bound on C_k due to Raz and Spieker.

▶ **Lemma 4.4** ([36]). $C_k = 2^{\Omega(k \log \log k)}$.

It is worth noting that C_k is believed to be bounded by $2^{\Omega(k \log k)}$. Showing this would give us asymptotically tight bounds on the value of C_k as the upper bound of $k! = 2^{\mathcal{O}(k \log k)}$ is trivial. This, however, remains an open problem.

A small rectangle cover of a matrix implies the existence of large rectangles in it, but the converse does not need to hold in the general case. The following claim and lemma shows, that in the case of $\mathcal{M}_k$, the size of a minimal rectangle cover and maximal rectangle are in fact related, and within a poly-logarithmic factor of what one can expect.

▷ Claim 4.5. Let R be any rectangle in $\mathcal{M}_k$. Then, there exists a rectangle cover of $\mathcal{M}_k$ of size $\ell = \frac{k!(k-1)!}{|R|} \cdot 2k \ln k$.

Proof. Let $Q \subseteq \mathcal{S}_k \times \mathcal{S}_k$ denote the set of pairs (ρ_1, ρ_2) such that $\rho_2\rho_1 \in \bar{\mathcal{S}}_k$, i.e., the set of 1-entries of $\mathcal{M}_k$. Let P_1 and P_2 denote the sets of permutations which are indices of, respectively, rows and columns of R. Thus, for every $\rho_1 \in P_1, \rho_2 \in P_2$, we have $(\rho_1, \rho_2) \in Q$. Consider a map $\mu_{\alpha,\beta} : \mathcal{S}_k \times \mathcal{S}_k \to \mathcal{S}_k \times \mathcal{S}_k$ parameterized by permutations $\alpha, \beta \in \mathcal{S}_k$, defined as

$$\mu_{\alpha,\beta}(\rho_1, \rho_2) = (\alpha\rho_1\beta, \beta^{-1}\rho_2\alpha^{-1}).$$

First, note that $\rho_2\rho_1 \in \bar{\mathcal{S}}_k$ if and only if $(\beta^{-1}\rho_2\alpha^{-1})(\alpha\rho_1\beta) = \beta^{-1}\rho_2\rho_1\beta \in \bar{\mathcal{S}}_k$, as permutation conjugation preserves cycle type, hence $(\rho_1, \rho_2) \in Q$ if and only if $\mu_{\alpha,\beta}(\rho_1, \rho_2) \in Q$.

Let α, β be two random permutations sampled independently from a uniform distribution on $\mathcal{S}_k$. First, we show that for any two pairs $(\rho_1, \rho_2), (\sigma_1, \sigma_2) \in Q$, we have

$$\Pr[\mu_{\alpha,\beta}(\rho_1, \rho_2) = (\sigma_1, \sigma_2)] = \frac{1}{k!(k-1)!}.$$

To do this, we show that there are k different pairs α, β for which $\alpha\rho_1\beta = \sigma_1$ and $\beta^{-1}\rho_2\alpha^{-1} = \sigma_2$. By Proposition 2.1, we have exactly k permutations β which satisfy $\sigma_2\sigma_1 = \beta^{-1}\rho_2\rho_1\beta$. For each of those, we must put $\alpha = \sigma_2^{-1}\beta^{-1}\rho_2$ to obtain a pair satisfying the conditions.

For any pair $(\rho_1, \rho_2) \in Q$, we have $(\rho_1, \rho_2) \in \mu_{\alpha,\beta}(R)$ if and only if $\mu_{\alpha^{-1},\beta^{-1}}(\rho_1, \rho_2) \in R$, and therefore

$$\Pr[(\rho_1, \rho_2) \in \mu_{\alpha,\beta}(R)] = \frac{|R|}{k!(k-1)!}.$$

Now, sample ℓ pairs α_i, β_i independently uniformly from $\mathcal{S}_k \times \mathcal{S}_k$ and let $R_i = \mu_{\alpha_i,\beta_i}(R)$. Clearly $R_i \subseteq Q$ for each $i \in [\ell]$. Let $\bar{R} = \bigcup_{i\in[\ell]} R_i$. For any $(\rho_1, \rho_2) \in Q$, we have

$$\Pr\left[(\rho_1, \rho_2) \notin \bar{R}\right] \leq \left(1 - \frac{|R|}{k!(k-1)!}\right)^{\ell} < e^{-2k \ln k} = k^{-2k} < (k!(k-1)!)^{-1}.$$

By union bound, we have $\Pr\left[\bar{R} \neq Q\right] < 1$, hence there exist a choice of α_i, β_i for which $\bar{R} = Q$. For this choice of α_i, β_i we have that $R_1, \ldots, R_\ell$ is a rectangle cover, which finishes the proof. ◁

Immediately, we get the following lemma as a corollary.

▶ **Lemma 4.6.** *Every rectangle in $\mathcal{M}_k$ has size at most $k!(k-1)! \cdot C_k^{-1} \cdot 2k \ln k$.*

There is a natural correspondence between permutations in $\mathcal{S}_k$ and perfect matchings of a complete bipartite graph $K_{k,k}$. Let us label the vertices of both sides of a bipartition of $K_{k,k}$ as respectively $v_1, \ldots, v_k$ and $u_1 \ldots, u_k$. For any perfect matching $M \subseteq \{v_1, \ldots, v_k\} \times \{u_1, \ldots, u_k\}$, its corresponding permutation is $\rho \in \mathcal{S}_k$ defined as

$$\rho(s) = t \quad \text{if and only if} \quad (v_s, u_t) \in M.$$

It is easy to see that for any two perfect matchings M_1, M_2 in $K_{k,k}$, their union $M_1 \cup M_2$ forms a Hamiltonian cycle iff $\rho_2^{-1}\rho_1 \in \bar{\mathcal{S}}_k$ iff $\rho_1^{-1}\rho_2 \in \bar{\mathcal{S}}_k$, where ρ_i denotes a permutation corresponding to M_i.

4.2 Complete graphs

Similarly to bipartite cliques, every perfect matching in a complete graph K_{2k} can be represented by a permutation in the set $\mathcal{S}^2_{2k}$. It is then easy to see that the union of two perfect matchings M_1, M_2 in K_{2k} forms a Hamiltonian cycle in K_{2k} iff $\rho_2\rho_1$ is in $\mathcal{S}^k_{2k}$.

▶ **Definition 4.7.** *A clique matching compatibility matrix $\mathcal{M}^*_k$ of order k is a binary matrix of size $(2k-1)!! \times (2k-1)!!$ with rows and columns indexed by permutations from $\mathcal{S}^2_{2k}$ which satisfies*

$$\mathcal{M}^*_k(\rho_1, \rho_2) = 1 \quad \textit{if and only if} \quad \rho_2\rho_1 \in \mathcal{S}^k_{2k}.$$

▶ **Lemma 4.8.** *Every rectangle in $\mathcal{M}^*_k$ has size at most $(2k-1)! \cdot C_k^{-1} \cdot 2k \ln k$.*

Proof. Pick any such rectangle R and let P_1, P_2 denote the sets of permutations which are indices of, respectively, rows and columns of R. For a set $C \subseteq [2k]$ of size k, we define $\mathcal{S}_C \subseteq \mathcal{S}^2_{2k}$ as the permutations with all cycles of size 2 which map all elements of C to $[2k] - C$. Note that $\mathcal{S}_C = \mathcal{S}_{[2k]-C}$. Moreover, for every pair $\rho_1 \in P_1, \rho_2 \in P_2$, we have $\rho_1, \rho_2 \in \mathcal{S}_C$ if we set C to be one of two maximal independent sets of the cycle formed by the union of the edges of ρ_1, ρ_2. Therefore

$$P_1 \times P_2 \subseteq \bigcup_{\substack{C \cup D = [2k],\ 1 \in C \\ |C|=|D|=k}} \mathcal{S}_C \times \mathcal{S}_C,$$

and so

$$|P_1| \cdot |P_2| \leq \sum_{\substack{C \cup D = [2k],\ 1 \in C \\ |C|=|D|=k}} |(P_1 \cap \mathcal{S}_C) \times (P_2 \cap \mathcal{S}_C)|.$$

Fix $C \cup D = [2k]$, $|C| = |D| = k$, and enumerate $C := \{c(1), \ldots, c(k)\}$ and $D := \{d(1), \ldots, d(k)\}$ (we treat c and d as bijective functions from $[k]$ to resp. C and D). Put $P_1^* = \{d^{-1}\rho_1 c : \rho_1 \in P_1 \cap \mathcal{S}_C\}$ and $P_2^* = \{c^{-1}\rho_2 d : \rho_2 \in P_2 \cap \mathcal{S}_C\}$. Note that both sets are well defined and $P_1^*, P_2^* \subseteq \mathcal{S}_k$. Since c, d are bijective, we have $|P_1^*| = |P_1 \cap \mathcal{S}_C|$ and $|P_2^*| = |P_2 \cap \mathcal{S}_C|$.

Finally, for every $\rho_1^* \in P_1^*, \rho_2^* \in P_2^*$, we have $\rho_2^*\rho_1^* = c^{-1}(\rho_2\rho_1)c \in \bar{\mathcal{S}}_k$. Thus, P_1^*, P_2^* form a rectangle in $\mathcal{M}_k$, hence by Lemma 4.6,

$$|(P_1 \cap \mathcal{S}_C) \times (P_2 \cap \mathcal{S}_C)| = |P_1^* \times P_2^*| \leq k!(k-1)! \cdot C_k^{-1} \cdot 2k \ln k.$$

Summing over all C, we get $|P_1| \cdot |P_2| \leq \binom{2k-1}{k-1} \cdot k!(k-1)! \cdot C_k^{-1} \cdot 2k \ln k = (2k-1)! \cdot C_k^{-1} \cdot 2k \ln k$, which finishes the proof of the lemma. ◀

5 Traveling salesperson problem

Directed graphs. Let $G_{n,k}$ denote a directed graph with $V(G_{n,k}) = \{v_{c,r} : c \in [n], r \in [k]\}$ and

$$E(G_{n,k}) = \{(v_{c,r_1}, v_{(c \bmod n)+1,r_2}) \mid c \in [n], r_1, r_2 \in [k]\}.$$

It is easy to see that the pathwidth of $G_{n,k}$ is at most $3k$. Combining the following lemma with Lemma 4.4 immediately gives Theorem 1.2. The rest of this section will be dedicated to proving it.

▶ **Lemma 5.1.** *For every $k \geq 1$ and $n \geq 3k+3$, any tropical circuit calculating $DTSP_{G_{n,k}}$ is of size at least $C_k/(2k \ln k)$.*

Proof. Fix n, k and put $G := G_{n,k}$. Let $V_i = \{v_{i,r} : r \in [k]\}$. Let $G^i = G[V_i \cup V_{(i \bmod n)+1}]$ and let $E_i = E(G^i)$, i.e., E_i contains all edges whose tail belongs to V_i. For every perfect matching M of G^i, we will identify it with a permutation $\rho \in \mathcal{S}_k$ defined as

$$\rho(s) = t \quad \text{iff} \quad (v_{i,s}, v_{(i \bmod n)+1,t}) \in M.$$

For every Hamiltonian cycle H of G, the set $E(H) \cap E_i$ is a perfect matching in G^i. We will say that the sequence of permutations $\rho_1, \ldots, \rho_n \in \mathcal{S}_k$ represents H if $E(H) \cap E_i = \rho_i$ for all $i \in [n]$. Note that such representing set is unique, and moreover, satisfies

$$\rho_i \rho_{i-1} \ldots \rho_2 \rho_1 \rho_n \rho_{n-1} \ldots \rho_{i+2} \rho_{i+1} \in \bar{\mathcal{S}}_k$$

for any $i \in [k]$. Conversely, every sequence of permutations satisfying the above represents a unique Hamiltonian cycle of G. In particular, there are $(k-1)! \cdot (k!)^{n-1}$ such sequences and, hence, Hamiltonian cycles in G. Additionally, for any set of indices $i_1, \ldots, i_\ell \in [n]$ of size at most $n-1$, if we fix $\rho_{i_1}, \ldots, \rho_{i_\ell}$, then there are exactly $(k-1)! \cdot (k!)^{n-1-\ell}$ ways to fix the rest of the permutations for the above inclusion to hold.

Now, look at the polynomial $DTSP_G$. Let $\bar{E} = \{(v_{i,1}, v_{(i \bmod n)+1,1}) \mid i \in [n]\}$ and let $\bar{X} = \{x_{s,t} \mid (s,t) \in \bar{E}\}$. That is, the set $\bar{X}$ contains exactly one variable corresponding to an edge in E_i for each $i \in [n]$. Similarly, let X_i denote the set $\{x_{s,t} \mid (s,t) \in E_i\}$.

We will say that a pair of polynomials g, h forms a rectangle $g \cdot h$ in $DTSP_G$ if $g \cdot h \subseteq DTSP_G$, i.e., for every pair of monomials $g' \in g, h' \in h$, the variables of $\sup(g') \cup \sup(h')$ correspond to edges forming a Hamiltonian cycle of G. Additionally, we will say that such rectangle is balanced if $|\sup(g) \cap \bar{X}|, |\sup(h) \cap \bar{X}| \leq \frac{2}{3}|\bar{X}|$. We would like to prove the following claim.

▷ Claim 5.2. Let $g \cdot h$ be any balanced rectangle in $DTSP_G$. Then

$$|g \cdot h| \leq (k-1)! \cdot (k!)^{n-1} \cdot C_k^{-1} \cdot (2k \ln k).$$

First, we finish the proof of the lemma given the claim. Let τ denote the size of the smallest tropical circuit calculating $DTSP_G$. Lemma 2.4 says that $DTSP_G$ can be covered by a union of τ balanced rectangles. Thus,

$$(k-1)! \cdot (k!)^{n-1} = |DTSP_G| \leq \tau \cdot (k-1)! \cdot (k!)^{n-1} \cdot C_k^{-1} \cdot (2k \ln k),$$

hence $\tau \geq C_k/(2k \ln k)$. The rest of the section is dedicated to proving the claim.

Fix any balanced rectangle $g \cdot h$. Obviously, supports of g and h are disjoint. We will say that E_i is monochromatic w.r.t. $g \cdot h$ if either $X_i \cap \sup(g)$ or $X_i \cap \sup(h)$ is empty. Let $i_1, \ldots, i_\ell$ denote the indices of sets E_i that are not monochromatic. We consider two cases depending on whether $\ell > k$ or not.

Assume $\ell > k$ and fix a non-monochromatic index i_j. Define

$$V^g_{tail} = \{u \in V_{i_j} \mid \exists_{w \in V_{(i_j \bmod n)+1}} x_{u,w} \in \sup(g)\},$$

$$V^g_{head} = \{w \in V_{(i_j \bmod n)+1} \mid \exists_{u \in V_{i_j}} x_{u,w} \in \sup(g)\},$$

i.e., V^g_{tail} (resp. V^g_{head}) denote tails (resp. heads) of all edges in E_{i_j} whose related variables belong to the support of g. We define V^h_{tail}, V^h_{head} analogously. It is easy to see that $V^g_{tail} \cap V^h_{tail} = V^g_{head} \cap V^h_{head} = \emptyset$.

A perfect matching in G^{i_j} induced by any monomial of $g \cdot h$ must match V^g_{head} with V^g_{tail} and V^h_{head} with V^h_{tail}, and the number of possible ways to do that is $|V^g_{head}|! \cdot |V^h_{head}|!$, which is at most $(k-1)!$ as both sets are nonempty and disjoint. Therefore, the number of all cycles corresponding to monomials of $g \cdot h$ is at most

$$((k-1)!)^\ell \cdot (k!)^{n-\ell} = \frac{1}{k^\ell} \cdot (k!)^n \le \frac{1}{k^{k+1}} \cdot (k!)^n = (k-1)! \cdot (k!)^{n-1} \cdot \frac{1}{k^k}$$

and $\frac{1}{k^k} \le C_k^{-1} \le C_k^{-1} \cdot (2k \ln k)$.

Now, assume $\ell \le k$. Thus, at least $n - k \ge \frac{2}{3}n + 1$ of E_i are monochromatic. Since $g \cdot h$ is balanced, we have two indices i_g, i_h such that E_{i_g}, E_{i_h} are both monochromatic, and both intersections $X_{i_g} \cap \sup(h)$ and $X_{i_h} \cap \sup(g)$ are empty. In particular, the support of every monomial in g corresponds to edges whose intersection with E_{i_g} induces a perfect matching in G^{i_g}. The same holds for h and G^{i_h}. W.l.o.g. we can assume that $i_g < i_h$.

For a monomial $g' \in g$ (and analogously for $h' \in h$), we define its type as the set

$$\lambda_{g'} := \left\{ e \in E(G) - (E_{i_g} \cup E_{i_h}) \,:\, x_e \in \sup(g') \right\}.$$

Fix an arbitrary pair of types $\lambda_{g^*}, \lambda_{h^*}$ and let g^* (resp. h^*) denote the set of all monomials of g (resp. h) with that type. The number of such possible pairs is bounded by the number of different projections of a Hamiltonian cycle onto $E(G) - (E_{i_g} \cup E_{i_h})$, hence is at most $(k!)^{n-2}$. By the definition, every cycle H corresponding to some monomial of $g^* \cdot h^*$ has the same intersection with E_i for all $i \notin \{i_g, i_h\}$.

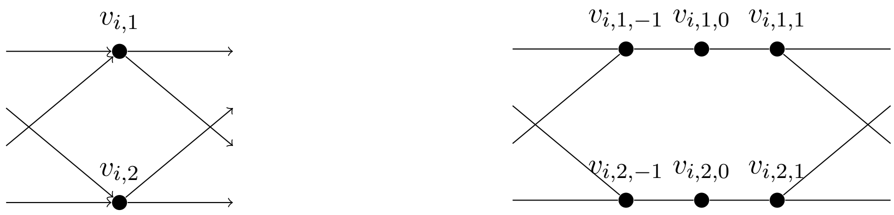
Let ρ_i for $i \in [n] - \{i_g, i_h\}$ represent the set $(\lambda_{g^*} \cup \lambda_{h^*}) \cap E_i$ which is a perfect matching in G^i. Let $\bar{\rho}_l = \rho_{i_h-1}\rho_{i_h-2} \ldots \rho_{i_g+1}$ and $\bar{\rho}_r = \rho_{i_g-1}\rho_{i_g-2} \ldots \rho_1 \rho_n \ldots \rho_{i_h+1}$. Let P_g (resp. P_h) denote the set of perfect matchings of G^{i_g} (resp. G^{i_h}) induced by the monomials in g^* (resp. h^*). Every permutation in the product $(\bar{\rho}_l P_g \bar{\rho}_r) \cdot P_h$ belongs to $\bar{\mathcal{S}}_k$, hence by Lemma 4.6,

$$k(k-1)! \cdot C_k^{-1} \cdot (2k \ln k) \ge |\bar{\rho}_l P_g \bar{\rho}_r| \cdot |P_h| = |P_g| \cdot |P_h| = |g^*| \cdot |h^*| \ge |g^* \cdot h^*|.$$

Therefore $|g \cdot h| \le (k!)^{n-1} \cdot (k-1)! \cdot C_k^{-1} \cdot (2k \ln k)$. ◀

Undirected graphs. Let $\bar{G}_{n,k}$ denote an undirected graph with $V(\bar{G}_{n,k}) = \{v_{c,r,i} \colon c \in [n], r \in [k], i \in \{-1, 0, 1\}\}$ and

$$\begin{aligned} E(\bar{G}_{n,k}) = \{(v_{c,r_1,1}, v_{(c \bmod n)+1,r_2,-1}) \mid c \in [n], r_1, r_2 \in [k]\} \cup \\ \{(v_{c,r,-1}, v_{c,r,0}) \mid c \in [n], r \in [k]\} \cup \\ \{(v_{c,r,0}, v_{c,r,1}) \mid c \in [n], r \in [k]\}. \end{aligned}$$



(a) i-th column of $G_{n,2}$. **(b)** in $\bar{G}_{n,2}$.

Figure 2 The reduction from $G_{n,k}$ to $\bar{G}_{n,k}$.

The graph $\bar{G}_{n,k}$ is obtained by performing a textbook reduction on $G_{n,k}$ from directed to undirected version of TSP.

Again, the pathwidth of $\bar{G}_{n,k}$ is at most $3k$. Combining the following lemma with Lemma 4.4 immediately gives Theorem 1.3.

▶ **Lemma 5.3.** *For any $k \geq 1$ and $n \geq 3k+3$, any tropical circuit calculating $TSP_{\bar{G}_{n,k}}$ uses at least $C_k/(2k \ln k)$ gates.*

Proof. Let $\bar{\Gamma}$ be an arbitrary tropical circuit of size $\bar{\tau}$ computing $TSP_{\bar{G}_{n,k}}$. We will show that there exists a tropical circuit Γ of size $\tau \leq \bar{\tau}$ computing $DTSP_{G_{n,k}}$. In particular, together with Lemma 5.1 this shows that $\bar{\tau} \geq C_k/(2k \ln k)$.

Let Γ be the tropical circuit obtained by taking $\bar{\Gamma}$ and performing the following substitutions:

- for every input node labeled with a variable x_{u_1,u_2}, where $u_1, u_2 \in V(\bar{G}_{n,k})$, $u_1 = v_{c,r,-1}, u_2 = v_{c,r,0}$, we replace it with a constant 0,
- for every input node labeled with a variable x_{u_1,u_2}, where $u_1, u_2 \in V(\bar{G}_{n,k})$, $u_1 = v_{c,r,0}, u_2 = v_{c,r,1}$, we replace it with a constant 0,
- for every input node labeled with a variable x_{u_1,u_2}, where $u_1, u_2 \in V(\bar{G}_{n,k})$, $u_1 = v_{c,r_1,1}, u_2 = v_{(c \bmod n)+1,r_2,-1}$, we replace it with a variable $x_{u'_1,u'_2}$ where $u'_1, u'_2 \in V(G_{n,k})$, $u'_1 = v_{c,r_1}, u'_2 = v_{(c \bmod n)+1,r_2}$.

Naturally, the size of Γ is at most $\bar{\tau}$. Let f be some polynomial calculated by Γ. The reduction works in such a way that there is a bijection between directed Hamiltonian cycles in $G_{n,k}$ and undirected ones in $\bar{G}_{n,k}$, and this bijection directly follows the substitution of edges described in the definition of Γ. In particular, it can be easily seen that $f \simeq DTSP_{G_{n,k}}$. ◀

6 Spanning tree

Let $H_{n,k}$ denote a directed graph with $V(H_{n,k}) = \{v_{c,r} : c \in [n], r \in [2k]\}$ and

$$E(H_{n,k}) = \{(v_{c_1,r_1}, v_{c_2,r_2}) \mid c_1, c_2 \in [n], r_1, r_2 \in [2k], |c_1 - c_2| = 1\} \cup \{(v_{c,r_1}, v_{c,r_2}) \mid c \in \{1, n\}, r_1, r_2 \in [2k], r_1 \neq r_2\}.$$

It is easy to see that the pathwidth of $H_{n,k}$ is at most $4k$. Combining the following lemma with Lemma 4.4 immediately gives Theorem 1.4.

▶ **Lemma 6.1.** *For any $k \geq 1$ and $n > 12k^2 \ln k + 3$, any tropical circuit calculating $DST_{H_{n,k}}$ uses at least $C_k/(2k^2 \ln k)$ gates.*

The rest of this section is dedicated to proving this lemma. The majority of the proof follows the same line of argumentation used to prove Lemma 5.1. For the rest of the section, we fix suitable n, k and put $G := H_{n,k}$. Let $V_i = \{v_{i,r} : r \in [2k]\}$. For $i \in [n-1]$, let

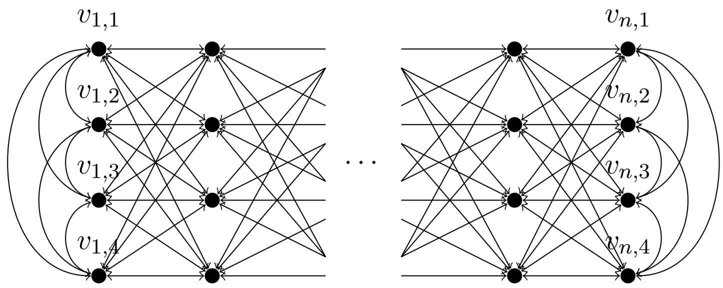


Figure 3 The graph $H_{n,2}$.

$G^i = G[V_i \cup V_{i+1}] - (E(G[V_i]) \cup E(G[V_{i+1}]))$ and let $G^0 = G[V_1], G^n = G[V_n]$. Note that G^i is complete bipartite for $1 \le i \le n-1$ and complete for $i \in \{0, n\}$. We denote $E_i = E(G^i)$ for all $i \in \{0, \dots, n\}$ and put $X_i = \{x_e : e \in E_i\}$.

For every perfect matching M of G^i for $i \in \{1, \dots, n-1\}$, we will identify it with a permutation $\rho \in \mathcal{S}_{2k}$ defined as

$$\rho(s) = t \quad \text{iff} \quad (v_{i,s}, v_{i+1,t}) \in M \text{ or } (v_{i+1,t}, v_{i,s}) \in M.$$

Similarly, for $i \in \{0, n\}$ we identify any perfect matching M of G^i with $\rho \in \mathcal{S}_{2k}^2$ such that

$$\rho(s) = t \quad \text{iff} \quad (v_{i,s}, v_{i,t}) \in M \text{ or } (v_{i,t}, v_{i,s}) \in M.$$

We will say that a Hamiltonian cycle of G is *nice* if its intersection with each E_i is a perfect matching in G^i (ignoring edge directions). We will say that a Hamiltonian path of G is *nice* if it is a subgraph of a nice Hamiltonian cycle and its first and last vertices belong to V_1. Every nice path is a subgraph of exactly one nice cycle, and conversely, every nice cycle contains exactly k nice paths as a subgraph.

We define the content of a nice Hamiltonian cycle H as the unique sequence of permutations $\rho_0, \dots, \rho_n \in \mathcal{S}_{2k}$ such that $E(H) \cap E_i = \rho_i$ for all $0 \le i \le n$. For the sake of clarity, for such a sequence, we define the following notation.

$$\overleftarrow{\rho_i} := \rho_i \rho_{i-1} \dots \rho_2 \rho_1 \qquad \overrightarrow{\rho_i} := \rho_{n-1} \rho_{n-2} \dots \rho_{i+1} \rho_i$$

We assume $\overleftarrow{\rho_0} = \overrightarrow{\rho_n} = \mathbf{id}_{\mathcal{S}_{2k}}$. Note that as described in Section 4.2,

$$\left(\overleftarrow{\rho_{i-1}} \cdot \rho_0 \cdot \overleftarrow{\rho_{i-1}}^{-1}\right) \cdot \left(\overrightarrow{\rho_i}^{-1} \cdot \rho_n \cdot \overrightarrow{\rho_i}\right) \in \mathcal{S}_{2k}^k \tag{1}$$

for any $i \in [n]$. Conversely, for every sequence of permutations satisfying the above (for arbitrarily chosen i), there are exactly two nice cycles H such that their content is $\rho_0, \dots, \rho_n$. Both cycles differ only by their direction. In particular, if Equation (1) holds for one such i, then it holds for every $i \in [n]$.

▷ Claim 6.2. There are $((2k)!)^{n-1} \cdot (2k-1)!$ sequences $(\rho_j)_{j=0}^n$ satisfying (1).

Proof. We fix ρ_j arbitrarily for all $j \ne i$; there are $((2k-1)!!)^2 \cdot ((2k)!)^{n-2}$ ways to do that. Let $\bar{\rho} = \left(\overrightarrow{\rho_i}^{-1} \cdot \rho_n \cdot \overrightarrow{\rho_i}\right)$, hence we have $\bar{\rho} = \rho_i^{-1} \tilde{\rho} \rho_i$, where $\tilde{\rho} \in \mathcal{S}_{2k}^2$ is already fixed. By Proposition 2.1, the number of values $\bar{\rho}$ can take for Equation (1) to hold is $(2k-2)!!$. By the same proposition, the number of $\rho_i \in \mathcal{S}_{2k}^2$ given $\bar{\rho}$ is $(2k)!!$. The total number of ways to fix all ρ_j is therefore $((2k-1)!!)^2 \cdot ((2k)!)^{n-2} \cdot (2k)!! \cdot (2k-2)!! = ((2k)!)^{n-1} \cdot (2k-1)!$. ◁

In particular, the number of nice cycles in G is $2 \cdot ((2k)!)^{n-1} \cdot (2k-1)!$. By a similar argument, for any set of indices $1 \leq i_1, \ldots, i_\ell \leq n-1$ of size at most $n-2$, if we fix $\rho_{i_1}, \ldots, \rho_{i_\ell}$ together with ρ_0 and ρ_n, then there are exactly $((2k)!)^{n-2-\ell} \cdot (2k)!! \cdot (2k-2)!!$ ways to fix the rest of the permutations for (1) to hold.

Proof (Lemma 6.1). We say that a rectangle $g \cdot h \subseteq DST_G$ covers a nice Hamiltonian cycle H if it contains a monomial encoding a nice Hamiltonian path which is a subgraph of H. Since all nice Hamiltonian paths are spanning out-trees as well, any rectangle decomposition of the polynomial calculated by our circuit must cover all nice cycles. We define rectangles being balanced as in the statement of Lemma 2.4 with respect to the set $X = \{x_{(v_{c,1}, v_{c+1,1})} \mid c \in [n-1]\}$.

▷ Claim 6.3. Let $g \cdot h$ be any balanced rectangle in DST_G. Then, the number of nice Hamiltonian cycles covered by $g \cdot h$ is at most

$$((2k)!)^n \cdot C_k^{-1} \cdot 2k \ln k.$$

For any tropical circuit calculating DST_G using τ gates, Lemma 2.4 implies that DST_G can be covered by a union of τ balanced rectangles. Thus, in order to cover every nice cycle, at least one balanced rectangle covers at least $\frac{2((2k)!)^{n-1}(2k-1)!}{\tau}$ of those, hence assuming Claim 6.3,

$$\tau \geq C_k / (2k^2 \ln k).$$

The rest of the section is dedicated to proving Claim 6.3. Fix any balanced rectangle $g \cdot h$. The supports of g and h are disjoint. We will say that E_i is monochromatic w.r.t. $g \cdot h$ if either $X_i \cap \sup(g)$ or $X_i \cap \sup(h)$ is empty.

▷ Claim 6.4. Pick $1 \leq i \leq n-1$. If E_i is not monochromatic w.r.t. $g \cdot h$, then there exists an edge $(u,v) \in E_i$ such that $x_{u,v} \notin \sup(g) \cup \sup(h)$.

Proof. Assume by contradiction that $x_{u,v} \in \sup(g)$ or $x_{u,v} \in \sup(h)$ for every $(u,v) \in E_i$. Since all monomials in $g \cdot h$ represent out-trees, for any pair of edges of the form $(u,v),(w,v) \in E_i$, either both $x_{u,v}, x_{w,v}$ belong to $\sup(g)$, or both belong to $\sup(h)$. Similarly, for any $(u,v) \in E_i$ both $x_{u,v}, x_{v,u}$ must also belong to the same support.

By our assumption, we have two edges $(p,q),(s,t) \in E_i$ such that $x_{p,q} \in \sup(g)$ and $x_{s,t} \in \sup(h)$ (and hence $x_{t,s} \in \sup(h)$ as well). Either (q,s) or (q,t) belong to E_i. In the former case, we have $x_{q,s} \in \sup(h)$ (as $x_{t,s} \in \sup(h)$) and $x_{s,q} \in \sup(g)$ (as $x_{p,q} \in \sup(g)$). In the latter, $x_{q,t} \in \sup(h)$ and $x_{t,q} \in \sup(g)$. Both cases arrive at a contradiction, which proves the claim. ◁

Let $i_1 < \cdots < i_\ell$ denote the indices in range $[n-1]$ of sets E_i which are not monochromatic. We consider two cases depending on whether $\ell > 4k^2 \ln k$ or not.

Assume $\ell > 4k^2 \ln k$. By sacrificing at most half of indices i_j, we can assume that they are non-adjacent. That is, we assume that $\ell > 2k^2 \ln k$ and that $i_{j+1} - i_j > 1$ for all $j \in [\ell - 1]$. For each $i \in \{i_j\}_{j=1}^{\ell}$, let (p_i, q_i) denote the edge of E_i given by Claim 6.4. Consider the following randomized procedure to generate a nice directed Hamiltonian cycle H of G.

- Select $\rho_0 \in \mathcal{S}_{2k}^2$ uniformly randomly.
- Select a subset of edges $F_0 \subseteq E_0$ by matching vertices of V_1 according to ρ_0 and picking the edge directions uniformly and independently. Let V_1^{tail} and V_1^{head} denote the vertices of V_1 composed of respectively tails and heads of the edges of F_0.

- For $i \in \{1, \ldots, n-1\}$ do the following. Pick $\rho_i \in \mathcal{S}_{2k}$ uniformly randomly and select $F_i \subseteq E_i$ by matching vertices of $V(G^i)$ according to ρ_i and picking the edge directions so that they agree with edges of F_{i-1} (i.e., if we match v_{i,r_1} with v_{i+1,r_2} and $v_{i,r_1} \in V_i^{tail}$, then we add (v_{i+1,r_2}, v_{i,r_1}) to F_i, and (v_{i,r_1}, v_{i+1,r_2}) otherwise). Define $V_{i+1}^{tail}, V_{i+1}^{head}$ analogously.
- The union $F_0 \cup \cdots \cup F_{n-1}$ at this point forms k disjoint directed paths whose endpoints are exactly V_n. There are exactly $(k-1)!$ ways to pick F_n in a way that connects these paths into a Hamiltonian cycle. We pick one of such ways uniformly randomly.

It is easy to see that this procedure generates all nice Hamiltonian cycles of G with uniform probability distribution. Now, we can bound the probability that a cycle generated this way does not contain any of the forbidden edges (p_i, q_i) given by Claim 6.4. Note that only such cycles can be covered by $g \cdot h$.

The steps during which the number of feasible choices could get restricted is when selecting F_{i_j} for some $j \in [\ell]$. Assume w.l.o.g. $p_{i_j} \in V_{i_j}, q_{i_j} \in V_{i_j+1}$, the other case is symmetric. If $p_{i_j} \in V_{i_j}^{tail}$, then every choice of F_{i_j} will be feasible, as an edge between p_{i_j}, q_{i_j} potentially added will always be directed from q_{i_j} to p_{i_j}. In case $p_{i_j} \in V_{i_j}^{head}$, there is exactly $\frac{1}{k}$ chance of adding the forbidden edge to F_{i_j}, as ρ_{i_j} is selected uniformly from whole $\mathcal{S}_{2k}$.

The crucial observation is that the probability that p_{i_j} is in $V_{i_j}^{head}$ at this point, assuming so far we did not pick any of the forbidden edges, is exactly $1/2$, independent of previous choices. This follows from the fact that ρ_{i_j-1} was chosen uniformly from $\mathcal{S}_{2k}$ and due to our assumption that $i_j > i_{j-1} + 1$. The partition of V_{i_j-1} into $V_{i_j-1}^{head}$ and $V_{i_j-1}^{tail}$ can have arbitrary distribution, however, both sets are always of size k, hence the probability of p_{i_j} getting matched by ρ_{i_j-1} with a vertex from $V_{i_j-1}^{tail}$ is exactly $\frac{1}{2}$.

This means, that the probability of a nice Hamiltonian cycle containing none of the forbidden edges is at most

$$\left(1 - \frac{1}{2k}\right)^{\ell} < e^{-k \ln k} \le C_k^{-1} \le C_k^{-1} \cdot 2k^2 \ln k,$$

hence the number of such cycles is at most

$$\left(2((2k)!)^{n-1}(2k-1)!\right) \cdot \left(C_k^{-1} \cdot 2k^2 \ln k\right) = ((2k)!)^n \cdot C_k^{-1} \cdot 2k \ln k,$$

which finishes the proof in case $\ell > 4k^2 \ln k$.

Now, assume $\ell \le 4k^2 \ln k$. Thus, at least $n - 4k^2 \ln k \ge \frac{2n}{3} + 1$ of E_i are monochromatic. Since $g \cdot h$ is balanced, we have two indices $i_g, i_h \in [n-1]$ s.t. E_{i_g}, E_{i_h} are both monochromatic, and both $X_{i_g} \cap \sup(h)$ and $X_{i_h} \cap \sup(g)$ are empty. W.l.o.g. we can assume that $i_g < i_h$.

For a monomial $g' \in g$ (and analogously for $h' \in h$), we define its type as the set

$$\lambda_{g'} := \{e \in E(G) - (E_{i_g} \cup E_{i_h}) : x_e \in \sup(g')\}.$$

Fix an arbitrary pair of types $\lambda_{g^*}, \lambda_{h^*}$ and let g^* (resp. h^*) denote the set of all monomials of g (resp. h) with that type. By the definition, every spanning out-tree corresponding to some monomial of $g^* \cdot h^*$ has the same intersection with E_i for all $i \notin \{i_g, i_h\}$. If $g^* \cdot h^*$ covers at least one nice cycle, this intersection must induce a perfect matching in G^i for each $i \in [n] - \{i_g, i_h\}$ and a matching of size $k-1$ in G^0 (two unmatched vertices of G^0 correspond to the endpoints of all nice paths induced by $g^* \cdot h^*$). Let $(\rho_i)_{i \in [n]-\{i_g, i_h\}}$ denote the set $(\lambda_{g^*} \cup \lambda_{h^*}) \cap E_i$. Let ρ_0 denote the unique permutation of $\mathcal{S}_{2k}^2$ containing $(\lambda_{g^*} \cup \lambda_{h^*}) \cap E_0$ as a subgraph.

Let $\bar{\rho}_l = \overleftarrow{\rho_{i_g-1}}$, $\bar{\rho}_r = \overrightarrow{\rho_{i_h+1}}$ and $\bar{\rho}_m = \rho_{i_h-1}\rho_{i_h-2}\dots\rho_{i_g+2}\rho_{i_g+1}$. Let P_g (resp. P_h) denote the set of perfect matchings of G^{i_g} (resp. G^{i_h}) induced by the monomials in g^* (resp. h^*). Finally, put

$$P_g^* = \left\{\rho \cdot \bar{\rho}_l \cdot \rho_0 \cdot \bar{\rho}_l^{-1} \cdot \rho^{-1} \mid \rho \in P_g\right\}, \text{ and } P_h^* = \left\{\bar{\rho}_m^{-1} \cdot \rho^{-1} \cdot \bar{\rho}_r^{-1} \cdot \rho_n \cdot \bar{\rho}_r \cdot \rho \cdot \bar{\rho}_m \mid \rho \in P_h\right\}.$$

Note that P_g^* is a set of conjugations of ρ_0 by permutations of the form $(\rho \cdot \bar{\rho}_l)^{-1}$. By Proposition 2.1, we have $|\{\pi \in \mathcal{S}_{2k} \mid \pi^{-1}\rho_0\pi = \pi^*\}| = (2k)!!$ for any fixed $\pi^* \in \mathcal{S}_{2k}^2$, hence $|P_g| \le (2k)!!\cdot|P_g^*|$. Similarly $|P_h| \le (2k)!!\cdot|P_h^*|$. By Equation (1), every pair $\rho_g \in P_g^*, \rho_h \in P_h^*$ satisfies $\rho_h\rho_g \in \mathcal{S}_{2k}^k$, hence applying Lemma 4.8 gives us

$$2 \cdot |P_g| \cdot |P_h| \le 2((2k)!!)^2 \cdot |P_g^*| \cdot |P_h^*| \le 2((2k)!!)^2(2k-1)! \cdot C_k^{-1} \cdot 2k \ln k.$$

Finally, note that the quantity $2 \cdot |P_g| \cdot |P_h|$ bounds from above the number of nice cycles covered by $g^* \cdot h^*$. The number of pairs of types $\lambda_{g^*}, \lambda_{h^*}$ covering at least one nice cycle is bounded by the number of projections of a nice path onto $E(G) - (E_{i_g} \cup E_{i_h})$, hence by

$$k((2k-1)!!)^2((2k)!)^{n-3}.$$

Therefore, the total number of nice cycles covered by $g \cdot h$ is at most

$$\left(k((2k-1)!!)^2((2k)!)^{n-3}\right) \cdot \left(2((2k)!!)^2(2k-1)! \cdot C_k^{-1} \cdot 2k \ln k\right) = ((2k)!)^n \cdot C_k^{-1} \cdot 2k \ln k,$$

which finishes the proof of Lemma 6.1 ◀

7 Conclusion

As noted earlier, the exact nondeterministic communication complexity of the matchings compatibility matrix is still open: While $C_k = k^{\mathcal{O}(k)}$ holds trivially, the currently best known lower bound is $C_k = 2^{\Omega(k \log\log k)}$ [36]. Improving this lower bound to $k^{\mathcal{O}(k)}$ would automatically imply the same bound for Theorem 1.2, Theorem 1.3, and Theorem 1.4. In particular, this complexity would be asymptotically tight (up to factors polynomial in N), as tropical circuits of size $2^{\mathcal{O}(k \log k)} \cdot N^{\mathcal{O}(1)}$ can be easily constructed for all these problems by following algorithms based on naive dynamic programming on path decompositions. In the reverse direction, better upper bounds on C_k seem also useful for obtaining better tropical circuits for $DTSP_G, TSP_G$ and DST_G, and possibly even for obtaining faster algorithm for TSP parameterized by pathwidth. It seems however, as also expressed by the authors of [36], that C_k is closer to $2^{\Omega(k \log k)}$ than it is to $2^{\Theta(k \log\log k)}$.

Another natural opportunity for further work would be to find lower bounds for other computational problems, for example it seems plausible that with tools from Theorem 1.1 and the reduction ideas from [28] one can also obtain a $\Omega(3^k)$ lower bound for tropical circuits calculating the minimum weight dominating set of a graph with pathwidth k.

More ambitiously, a natural open question is whether the lower bounds on tropical circuits as defined in this paper can be generalized to more expressive variants of tropical circuits (see the book by Jukna [22]). Similarly, an interesting, albeit seemingly more difficult, question is whether our introduced ideas could be used to prove matching unconditional lower bounds on the extension complexity of TSP polytopes in terms of treewidth, or whether the methods from [5] are actually captured by extension complexity and allow for an extended formulation of the TSP polytope on n-vertex graph of treewidth k of complexity $2^{\mathcal{O}(k)}n^{\mathcal{O}(1)}$.

References

1 Vladimir L Arlazarov, EA Dinic, MA Kronrod, and IA Faradzev. On economical construction of the transitive closure of a directed graph. In *Dokl. Akad. Nauk SSSR*, volume 194, pages 1209–1210, 1970.

2 Christopher Beck and Russell Impagliazzo. Strong ETH holds for regular resolution. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 487–494. ACM, 2013. doi:10.1145/2488608.2488669.

3 Richard Bellman. Dynamic programming treatment of the Travelling Salesman Problem. *J. ACM*, 9(1):61–63, 1962. doi:10.1145/321105.321111.

4 Andreas Björklund. Determinant sums for undirected Hamiltonicity. *SIAM J. Comput.*, 43(1):280–299, 2014. doi:10.1137/110839229.

5 Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Inf. Comput.*, 243:86–111, 2015. doi:10.1016/J.IC.2014.12.008.

6 Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009. doi:10.1016/J.JCSS.2009.04.001.

7 Asier Calbet. K_r-saturated graphs and the two families theorem. *arXiv preprint*, 2023. arXiv:2302.13389.

8 Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. Extended formulations in combinatorial optimization. *4OR*, 8(1):1–48, 2010. doi:10.1007/S10288-010-0122-Z.

9 Radu Curticapean, Nathan Lindzey, and Jesper Nederlof. A tight lower bound for counting Hamiltonian cycles via matrix rank. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1080–1099. SIAM, 2018. doi:10.1137/1.9781611975031.70.

10 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.

11 Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Fast Hamiltonicity checking via bases of perfect matchings. *J. ACM*, 65(3):12:1–12:46, 2018. doi:10.1145/3148227.

12 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. *ACM Trans. Algorithms*, 18(2):17:1–17:31, 2022. doi:10.1145/3506707.

13 Mateus de Oliveira Oliveira and Wim Van den Broeck. Optimal extended formulations from optimal dynamic programming algorithms. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pages 1881–1888. ijcai.org, 2024. URL: https://www.ijcai.org/proceedings/2024/208.

14 Samuel Fiorini, Serge Massar, Sebastian Pokutta, Hans Raj Tiwary, and Ronald de Wolf. Exponential lower bounds for polytopes in combinatorial optimization. *J. ACM*, 62(2):17:1–17:23, 2015. doi:10.1145/2716307.

15 Robert W. Floyd. Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345, 1962. doi:10.1145/367766.368168.

16 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *J. ACM*, 63(4):29:1–29:60, 2016. doi:10.1145/2886094.

17 Lance Fortnow and Rahul Santhanam. Infeasibility of instance compression and succinct PCPs for NP. *J. Comput. Syst. Sci.*, 77(1):91–106, 2011. doi:10.1016/J.JCSS.2010.06.007.

18 Mika Göös, Rahul Jain, and Thomas Watson. Extension complexity of independent set polytopes. *SIAM J. Comput.*, 47(1):241–269, 2018. doi:10.1137/16M109884X.

19 Michael. Held and Richard M. Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, 10(1):196–210, 1962. doi:10.1137/0110015.

20 Christoph Hertrich and Georg Loho. Neural networks and (virtual) extended formulations. *arXiv preprint*, 2024. doi:10.48550/arXiv.2411.03006.

21 Mark Jerrum and Marc Snir. Some exact complexity results for straight-line computations over semirings. *J. ACM*, 29(3):874–897, 1982. doi:10.1145/322326.322341.

22 Stasys Jukna. *Tropical Circuit Complexity: Limits of Pure Dynamic Programming*. Springer International Publishing, Cham, 2023. doi:10.1007/978-3-031-42354-3.

23 Stasys Jukna and Hannes Seiwert. Greedy can beat pure dynamic programming. *Inf. Process. Lett.*, 142:90–95, 2019. doi:10.1016/J.IPL.2018.10.018.

24 Richard M. Karp, Eli Upfal, and Avi Wigderson. Constructing a perfect matching is in random NC. *Comb.*, 6(1):35–48, 1986. doi:10.1007/BF02579407.

25 Leslie Robert Kerr. The effect of algebraic structure on the computational complexity of matrix multiplication. Technical report, Cornell University, 1970.

26 Petr Kolman, Martin Koutecký, and Hans Raj Tiwary. Extension complexity, MSO logic, and treewidth. *Discret. Math. Theor. Comput. Sci.*, 22(4), 2020. doi:10.23638/DMTCS-22-4-8.

27 Tuukka Korhonen. Lower bounds on dynamic programming for maximum weight independent set. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, Glasgow, Scotland (Virtual Conference), July 12-16, 2021*, volume 198 of *LIPIcs*, pages 87:1–87:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.87.

28 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Known algorithms on graphs of bounded treewidth are probably optimal. *ACM Trans. Algorithms*, 14(2):13:1–13:30, 2018. doi:10.1145/3170442.

29 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Slightly superexponential parameterized problems. *SIAM J. Comput.*, 47(3):675–702, 2018. doi:10.1137/16M1104834.

30 László Lovász and Michael E. Saks. Lattices, Möbius functions and communication complexity. In *29th Annual Symposium on Foundations of Computer Science, White Plains, New York, USA, 24-26 October 1988*, pages 81–90. IEEE Computer Society, 1988. doi:10.1109/SFCS.1988.21924.

31 Shachar Lovett. Recent advances on the log-rank conjecture in communication complexity. *Bull. EATCS*, 112, 2014. URL: http://eatcs.org/beatcs/index.php/beatcs/article/view/260.

32 R. Kipp Martin. Using separation algorithms to generate mixed integer model reformulations. *Oper. Res. Lett.*, 10(3):119–128, 1991. doi:10.1016/0167-6377(91)90028-N.

33 R. Kipp Martin, Ronald L. Rardin, and Brian A. Campbell. Polyhedral characterization of discrete dynamic programming. *Oper. Res.*, 38(1):127–138, 1990. doi:10.1287/OPRE.38.1.127.

34 Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Comb.*, 7(1):105–113, 1987. doi:10.1007/BF02579206.

35 Jesper Nederlof. Bipartite TSP in $O(1.9999^n)$ time, assuming quadratic time matrix multiplication. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 40–53. ACM, 2020. doi:10.1145/3357713.3384264.

36 Ran Raz and Boris Spieker. On the "log rank"-conjecture in communication complexity. *Combinatorica*, 15(4):567–588, 1995.

37 Thomas Rothvoss. The matching polytope has exponential extension complexity. *J. ACM*, 64(6):41:1–41:19, 2017. doi:10.1145/3127497.

38 Virginia Vassilevska Williams and R. Ryan Williams. Subcubic equivalences between path, matrix, and triangle problems. *J. ACM*, 65(5):27:1–27:38, 2018. doi:10.1145/3186893.

39 Stephen Warshall. A theorem on boolean matrices. *J. ACM*, 9(1):11–12, 1962. doi:10.1145/321105.321107.

40 Mihalis Yannakakis. Expressing combinatorial optimization problems by linear programs. *J. Comput. Syst. Sci.*, 43(3):441–466, 1991. doi:10.1016/0022-0000(91)90024-Y.