On the Optimality of Bellman–Ford–Moore Shortest Path Algorithm[☆]

Stasys Jukna[a,1,*], Georg Schnitger[a]

[a] *Institute of Computer Science, Goethe University Frankfurt, Frankfurt am Main, Germany*

Abstract

We prove a general lower bound on the size of switching-and-rectifier networks over any semiring of zero characteristic, including the $(\min, +)$ semiring. Using it, we show that the classical dynamic programming algorithm of Bellman, Ford and Moore for the shortest s-t path problem is optimal, if only Min and Sum operations are allowed.

Keywords: Shortest paths, matrix multiplication, dynamic programming, tropical semiring, lower bounds

1. Introduction

Dynamic programming algorithms for discrete minimization problems are actually (recursively constructed) circuits or switching networks over the $(\min, +)$ semiring, also known as the *tropical* semiring. So, in order to understand the limitations of dynamic programming, we need lower-bound arguments for tropical circuits and switching networks.

In this paper, we present such an argument for tropical switching networks over the $(\min, +)$ semiring. These networks correspond to dynamic programming algorithms solving minimization problems $f : \mathbb{N}^n \to \mathbb{N}$ of the form

$$f(x_1, \ldots, x_n) = \min_{a \in A} \sum_{i=1}^{n} a_i x_i \,, \tag{1}$$

where $A \subset \mathbb{N}^n$ is a finite set of nonnegative integer vectors $a = (a_1, \ldots, a_n)$. We prove that every tropical switching network solving f must have at least $f(1, \ldots, 1) \cdot c(f)$ edges, where $c(f)$ is the smallest size of a subset $S \subseteq [n] = \{1, \ldots, n\}$ such that, for every vector $a \in A$, there is a position $i \in S$ with $a_i \neq 0$ (Sect. 3). We then demonstrate this general lower bound by two almost optimal lower bounds.

Shortest paths. Our first application—which was also our main motivation—concerns the classical dynamic programming algorithm of Ford [1], Bellman [2], and Moore [3] for the shortest s-t path problem. This algorithm actually solves the *shortest k-walk* problem: given an assignment of nonnegative weights to the edges of the complete graph on $[n] = \{1, \ldots, n\}$, find the minimum weight of a walk of length k from node $s = 1$ to the node $t = n$. Recall

[☆] Research supported by the DFG grant SCHN 503/6-1.

[*] Corresponding author, jukna@thi.informatik.uni-frankfurt.de

Email address: schnitger@thi.informatik.uni-frankfurt.de (Georg Schnitger)

[1] Affiliated with Institute of Mathematics and Informatics, Vilnius University, Vilnius, Lithuania.

that a *walk* is an alternating sequence of nodes and connecting edges. A walk can travel over any node (except of s and t) and any edge (including loops) any number of times. A *path* is a walk which cannot travel over any node more than once. The *length* of a walk (or path) is its number of edges, counting repetitions.

In a related *shortest k-path* problem, the goal is to compute the minimum weight of an s-t path of length *at most* k. Note that, if we give zero weight to all loops, then these two problems are equivalent. This holds because weights are nonnegative, every s-t walk of length k contains an s-t path of length $\leqslant k$, and every s-t path of length $\leqslant k$ can be extended to an s-t walk of length k by adding loops.

The Bellman–Ford algorithm gives a tropical switching network of depth k, with kn nodes and kn^2 edges solving the k-walk problem, and hence, also the shortest k-path problem. By combining our general lower bound with a result of Erdős and Gallai [4] about the maximal number of edges in graphs without long paths, we show (Theorem 1) that this algorithm is almost optimal: at least about $kn(n-k)$ edges are also necessary in *any* tropical switching network solving the k-walk problem. We also show that the same number of edges is necessary even in *boolean* switching networks, if their depth is restricted to k (Theorem 4).

Matrix multiplication. Our next application concerns the complexity of matrix multiplication over the $(\min, +)$ semiring. Kerr [5] has shown that any $(\min, +)$ circuit, simultaneously computing *all* the n^2 entries of the product of two $n \times n$ matrices over the $(\min, +)$ semiring, requires $\Omega(n^3)$ gates. This showed that the dynamic programming algorithm of Floyd [6] and Warshall [7] for the all-pairs shortest paths problem is optimal, if only Min and Sum operations are allowed. Later, Pratt [8], Paterson [9], and Mehlhorn and Galil [10] independently proved the same lower bound even over the boolean semiring.

These lower bounds, however, do not imply the same lower bound for the *single-output* version M_n of this problem: compute the sum of all entries of the product matrix. Using our general lower bound, we show that the minimum number of switches in a tropical switching network solving M_n over the $(\min, +)$ semiring is $2n^3$ (Theorem 3).

Remark 1. Let us stress that we are interested in proving lower bounds for problems that *have* very small switching networks. In both problems above, we have $N = \Theta(n^2)$ variables. These problems *have* tropical switching networks of sizes $O(kN)$ and $O(N^{3/2})$, respectively. Are these upper bounds tight?

Using known lower-bound arguments for monotone boolean and arithmetic circuits, large (even exponential) lower bounds can be derived for tropical circuits solving some minimization problems as the minimum weight spanning tree, or the minimum weight perfect matching problem (see, e.g. [11, Theorem 30] and references herein). However, these arguments are too "generous" and fail for problems that *have* small tropical complexity.

Fortunately, there is a classical lower-bound argument of Shannon, Moore and Markov allowing to prove also small lower bounds for monotone boolean switching networks. By an extension of this argument to tropical networks, we will show that the two upper bounds above are indeed optimal.

In technical terms, none of the proofs in this paper is complicated. Our main contribution is a somewhat unexpected *connection* between different topics—some central dynamic programming algorithms, tropical mathematics, extremal graph theory, and classical lower bounds for monotone switching networks.

2. Polynomials and their switching networks

Let $(R, +, \times, \mathsf{0}, \mathsf{1})$ be a semiring with "sum" ($+$) and "product" ($\times$) operations, additive identity ("zero element") $\mathsf{0}$, and multiplicative identity $\mathsf{1}$ ("unit element"). We only consider commutative semirings, and assume the "annihilation" property $x \times \mathsf{0} = \mathsf{0}$ of the zero element. Recall that a (multivariate) polynomial over R is a formal expression of the form

$$f(x_1, \ldots, x_n) = \sum_{a \in A} c_a \prod_{i=1}^{n} x_i^{a_i}\,, \tag{2}$$

where $A \subset \mathbb{N}^n$ is a finite set of nonnegative integer vectors, and $c_a \geqslant 1$ are integer coefficients. The coefficients c_a are not necessarily elements of R: they only indicate the number of times the corresponding to them monomials appear in the polynomial. The *degree* of a monomial $\prod_{i=1}^{n} x_i^{a_i}$ is the sum $a_1 + a_2 + \cdots + a_n$ of its exponents.

Every polynomial f *defines* the function $f : R^n \to R$, whose value $f(r) = f(r_1, \ldots, r_n)$ is obtained by substituting elements $r_i \in R$ for x_i in f. Different polynomials may define the same function. Moreover, over different semirings R, these functions may be different. For example, in the *boolean* semiring, we have $R = \{0, 1\}$, $x + y := x \vee y$, $x \times y := x \wedge y$, $\mathsf{0} := 0$, and $\mathsf{1} := 1$, whereas in the *tropical* $(\min, +)$ semiring, we have $R = \mathbb{N} \cup \{+\infty\}$, $x + y := \min\{x, y\}$, $x \times y := x + y$, $\mathsf{0} := +\infty$, and $\mathsf{1} := 0$. Hence, over these two semirings, the functions defined by the polynomial (2) are, respectively,

$$f = \bigvee_{a \in A} \bigwedge_{i\colon a_i \neq 0} x_i \quad \text{and} \quad f = \min_{a \in A} \sum_{i=1}^{n} a_i x_i\,.$$

A semiring $(R, +, \times, \mathsf{0}, \mathsf{1})$ is of *zero characteristic*, if $\mathsf{1} + \mathsf{1} + \cdots + \mathsf{1} \neq \mathsf{0}$ holds for any finite sum of the unit element $\mathsf{1}$. Note that both semirings above are such.

The *support* of a monomial $p = \prod_{i=1}^{n} x_i^{a_i}$ is the set $X_p = \{x_i\colon a_i \neq 0\}$ of all variables occurring in the monomial with nonzero degree. A monomial of a polynomial f is *minimal*, if its support does not contain the support of any another monomial of f as a proper subset. Let $\mathrm{Sup}(f)$ denote the family of supports of all minimal monomials of f.

Lemma 1. *If two polynomials f and g define the same function over a semiring of zero-characteristic, then* $\mathrm{Sup}(f) = \mathrm{Sup}(g)$. *In particular, both polynomials then define the same function also over the boolean semiring.*

Proof. Let us first show that the support of every monomial of g must contain the support of at least one monomial of f, and vice versa.

Assume contrariwise that there is a monomial q of g such that $X_p \setminus X_q \neq \emptyset$ holds for all monomials p of f. If we set to $\mathsf{1}$ all variables in X_q, and set to $\mathsf{0}$ all the remaining variables, then on the resulting assignment a, we have that $f(a) = \mathsf{0}$, because every monomial of f contains at least one variable set to $\mathsf{0}$. But the monomial q of g is evaluated to $\mathsf{1}$. Since the semiring is of zero-characteristic, this yields $g(a) \neq \mathsf{0}$, a contradiction.

Assume now that $\mathrm{Sup}(f) \neq \mathrm{Sup}(g)$. Then, by symmetry, we may assume that there is a minimal monomial p of f such that $X_q \neq X_p$ holds for all monomials q of g. That is, for every q, we have either $X_q \setminus X_p \neq \emptyset$, or $X_q \subset X_p$ (proper inclusion). As we have shown in the previous paragraph, the support X_q of q must contain the support $X_{p'}$ of some monomial p' of f. Since the monomial p is minimal in f, this means that the strict inclusion $X_q \subset X_p$ is

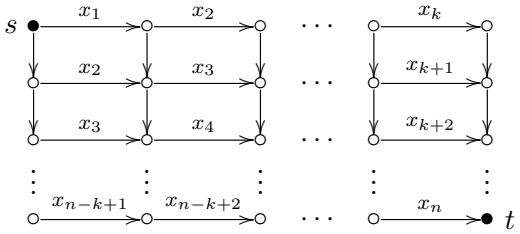

Figure 1: A switching-and-rectifying network computing the elementary symmetric polynomial

$$S_k^n(x_1,\ldots,x_n) = \sum_{|S|=k} \prod_{i\in S} x_i$$

over any semiring. The polynomial has $\binom{n}{k}$ monomials, but the network has only $k(n-k+1)$ switches.

impossible. Thus, we have that $X_q \setminus X_p \neq \emptyset$ must hold for all monomials q of g. If we set to $\mathsf{1}$ all variables in X_p, and set to $\mathsf{0}$ all the remaining variables, then on the resulting assignment a, we have that $g(a) = \mathsf{0}$. But since $q(a) = \mathsf{1}$ and the semiring is of zero-characteristic, we have that $f(a) \neq \mathsf{0}$, a contradiction. This completes the proof of the equality $\mathrm{Sup}(f) = \mathrm{Sup}(g)$.

Over the boolean semiring, this equality means that the monotone boolean functions corresponding to f and g have the same set of minterms, and hence, must coincide as boolean functions. □

A classical circuit model for computing polynomials $f(x_1,\ldots,x_n)$ over arbitrary semirings is that of *switching-and-rectifier networks*; to use a shorter term, we will call them just *switching networks*. Such a network is a directed acyclic graph G with two specified nodes, the source node s and the target node t. Paths from s to t are called *chains*. Each edge is either unlabeled (is a *rectifier*) or is labeled by some variable (is a *switch*). The graph may be a multigraph, that is, several edges may have the same endpoints. The *size* of a network is the total number of switches, and the *depth* is the maximum number of edges in a chain.

Every switching network G *produces* a unique polynomial f_G in a natural way. Namely, each chain π in G defines a monomial p_π, which is just the product of labels of switches along π. The polynomial f_G is then the sum of monomials p_π over all chains in G:

$$f_G(x_1,\ldots,x_n) = \sum_{\pi \text{ is a chain in } G} p_\pi\,.$$

The switching network G *computes* a given polynomial f over a semiring R, if both polynomials f_G and f define the same function over R. Every polynomial f can be computed by a trivial switching network which has a separate chain for each monomial of f. However, some polynomials allow much more compact representation as switching networks (see Fig. 1).

Remark 2. The example of the elementary symmetric polynomial S_k^n defined in Fig. 1 demonstrates that the presence of rectifiers (unlabeled edges) in switching networks *can* increase their power. The figure shows that, for $k = 2$, already $2n-2$ switches are enough over any semiring. On the other hand, results of Krichevskii [12] and Hansel [13] imply that, even over the boolean semiring, S_2^n requires $\Omega(n \log n)$ switches, if no rectifiers are allowed.

3. A general lower bound

Our starting point is a classical lower bound on the size of monotone boolean switching networks, i.e. networks over the boolean semiring. Recall that a *minterm* (resp., *maxterm*) of a monotone boolean function f is a minimal set of its variables such that, if all these variables are set to 1 (resp., to 0), then f outputs 1 (resp., 0) independently on the values of other variables.

The following lower bound is usually attributed to Markov [14]; a version of it for switching networks without rectifiers was earlier proved by Moore and Shannon [15]. The theorem is also reminiscent of the min-max fact—a dual to Menger's theorem attributed to Robacker [16]: the maximum number of edge-disjoint s-t cuts in a graph is equal to the minimum length of an s-t path.

Markov Theorem (Markov [14]). *If l is the minimum size of a minterm, and c the minimum size of a maxterm of a monotone boolean function f, then every monotone boolean switching network for f must have at least $l \cdot c$ switches.*

This theorem can already be used to derive lower bounds for tropical switching networks: by Lemma 1, every lower bound over the boolean semiring is also a lower bound over any semiring of zero characteristic, including the $(\min, +)$ semiring.

Example 1. Since all minterms of the boolean version of the elementary symmetric polynomial S_k^n defined in Fig. 1 have size $l = k$, and all maxterms have size $c = n - k + 1$, Markov's theorem immediately implies that every boolean switching network computing S_k^n must have at least $k(n - k + 1)$ switches. By Lemma 1, we have the same lower bound for tropical $(\min, +)$ switching networks solving the minimization problem

$$S_k^n(x_1, \ldots, x_n) = \min_{|S|=k} \sum_{i \in S} x_i \, .$$

This shows that the the naive switching network depicted in Fig. 1 is, in fact, optimal.

However, boolean versions of some important minimization problems have very short minterms. A prominent example in this respect is the k-walk polynomial f. This polynomial has, for example, the monomial $p = x_{1,2}x_{2,2}^{k-2}x_{2,n}$ of degree k corresponding to a walk of length k from 1 over 2 to n (the loop at node 2 is taken $k-2$ times). But the boolean version $x_{1,2}x_{2,2}x_{2,n}$ of this monomial has only three variables. So, Markov's theorem cannot yield any nontrivial lower bound on the *boolean* complexity of the k-walk polynomial.

To still be able to obtain nontrivial lower bounds at least for *tropical* switching networks, we will now extend Markov's theorem to arbitrary semirings of zero characteristic, including the $(\min, +)$ semiring.

Let $f(x_1, \ldots, x_n)$ be a polynomial over some semiring $(R, +, \times, 0, 1)$. Define the *length* $l(f)$ of f over R to be the smallest degree of a monomial in a polynomial g defining the same function over R as f. Note that $l(f)$ may crucially depend of the underlying semiring. For example, the polynomial $f = x^2y^3 + z^6$ has $l(f) = 5$ over the $(\min, +)$ semiring (then $f = \min\{2x + 3y, 6z\}$), but has $l(f) = 1$ over the boolean semiring (then $f = xy \vee z$).

The *cover number* $c(f)$ of a polynomial f is the minimum number of variables such that every monomial of f contains at least one of these variables. In other words, $c(f)$ is the minimum number of variables such that setting these variables to 0 forces f to output 0 independently on the values of the remaining variables.

Observation 1. *If two polynomials f and g define the same function over a semiring of zero-characteristic, then $l(f) = l(g)$ and $c(f) = c(g)$.*

Proof. The first equality $l(f) = l(g)$ follows directly from the definition of the length of polynomials, and holds over any semiring. The equality $c(f) = c(g)$ follows from Lemma 1, because $c(f)$ is exactly the minimum size of a set of variables of f intersecting all sets in $\mathrm{Sup}(f)$. □

Theorem 1. *Every switching network computing a polynomial f over a semiring of zero characteristic must have at least $l(f) \cdot c(f)$ switches.*

Proof. Take such a switching network G, and let f_G be the polynomial produced by G. The polynomial f_G must define the same function as f. Since the minimum degree of a monomial of f_G is exactly the minimum number of switches in a chain of G, Observation 1 implies that every chain of G must have at least $l(f)$ switches.

Define a *cut* of G to be a set of its switches such that every chain in G contains at least one switch in this set. The set of labels of each cut of G must clearly contain at least one variable of every monomial of the polynomial f_G and, by Lemma 1, also of the polynomial f. By Observation 1, every cut of G must have at least $c(f_G) = c(f)$ switches. So, to obtain the desired lower bound $l(f) \cdot c(f)$ on the total number of switches in G, it is enough to show that G must contain at least $l(f)$ edge-disjoint cuts.

For this, associate with every node u in G the minimum number d_u of switches in a path from the source node s to u. Hence, the source node s has $d_s = 0$. Since every chain of G must have at least $l(f)$ switches, the target node t has $d_t \geqslant l(f)$. Moreover, $d_v \leqslant d_u + 1$ holds for every edge $e = (u, v)$, and $d_v \leqslant d_u$ if the edge e is a rectifier. For every $0 \leqslant i \leqslant d_t - 1$, let C_i be the set of all edges (u, v) such that $d_u = i$ and $d_v = i + 1$. Since the sets C_i are clearly disjoint, and all edges in C_i must be switches, it remains to show that each C_i is a cut. For this, take an arbitrary chain $(u_1, u_2, \ldots, u_m)$ with $u_1 = s$ and $u_m = t$. The sequence of numbers $d_{u_1}, \ldots, d_{u_m}$ must reach the value $d_t \geqslant l(f)$ by starting at $d_s = 0$. At each step, the value can be increased by at most 1. So, there must be a j where a jump from $d_{u_j} = i$ to $d_{u_{j+1}} = i + 1$ happens, meaning that the edge (u_j, u_{j+1}) belongs to C_i, as desired. □

Note that, in the case of the boolean semiring, Theorem 1 is exactly the Markov theorem: in this case $l(f)$ is the minimum size of a minterm, and $c(f)$ the minimum size of a maxterm of f. What we win is that over some other semirings, including the $(\min, +)$ semiring, the length $l(f)$ of a polynomial f may be much larger than over the boolean semiring.

Observation 2. *The length of every polynomial $f(x_1, \ldots, x_n)$ over the $(\min, +)$ semiring is $l(f) = f(1, \ldots, 1)$.*

Proof. Over the $(\min, +)$ semiring, a monomial $p = \prod_{i=1}^{n} x_i^{a_i}$ turns into the sum $p = \sum_{i=1}^{n} a_i x_i$. Hence, the degree $a_1 + \cdots + a_n$ of p is then just its value $p(\vec{1})$ on the all-1 input vector $\vec{1} = (1, \ldots, 1)$. Since the value of f over the $(\min, +)$ semiring is the minimum of these sums p, we have that $f(\vec{1})$ is exactly the minimum degree of a monomial of f. If some other polynomial g defines the same function over $(\min, +)$, then $g(\vec{1}) = f(\vec{1})$ holds as well. So, $l(f) = f(\vec{1})$, as desired. □

4. Optimality of Bellman–Ford–Moore

The *k-walk polynomial* has one variable $x_{i,j}$ for each edge $\{i, j\}$ of the complete graph K_n on $[n] = \{1, \ldots, n\}$. Each of its monomials corresponds to a 1-to-n walk of length k, and has the form

$$x_{1,i_1} x_{i_1,i_2} \cdots x_{i_{k-2},i_{k-1}} x_{i_{k-1},n}$$

for not necessarily distinct nodes $i_1, \ldots, i_{k-1}$ in $\{2, \ldots, n-1\}$. In particular, we assume that each node, except 1 and n, has a loop.

Bellman–Ford–Moore Lemma (Bellman [2], Ford [1], Moore [3]). *Over any semiring, the k-walk polynomial can be computed by a switching network of depth k with at most kn nodes and at most kn^2 edges.*

Proof. The Bellman–Ford–Moore dynamic programming algorithm is amazingly simple. It computes the k-walk polynomial by recursively computing the polynomials $f_j^{(l)}$ whose monomials correspond to walks of length l from node 1 to node j. It first sets $f_j^{(1)} = x_{1,j}$ for all $j = 2, \ldots, n-1$, and uses the recursion

$$f_j^{(l+1)} = \sum_{i=2}^{n-1} f_i^{(l)} \times x_{i,j} \,.$$

To construct the desired switching network, arrange the nodes of the network into $k+1$ layers of nodes $V_0, V_1, \ldots, V_k$, where $V_0 = \{s\}$, $V_k = \{t\}$ and $|V_1| = \ldots = |V_{k-1}| = n-2$; each V_i for $i = 1, \ldots, k-1$ is a disjoint copy of the set of nodes $\{2, \ldots, n-1\}$. Edges go only from one layer to the next layer. For every $i, j \in \{2, \ldots, n-1\}$, the j-th node on the $(l+1)$-th layer is entered by a switch labeled by $x_{i,j}$ from the i-th node on the previous l-th layer. The network has $(k-1)(n-2)+2 \leqslant kn$ nodes, $(k-2)(n-2)^2 + 2(n-2) \leqslant kn^2$ edges, and its depth is k. □

Remark 3. Two widely considered versions of switching networks are *contact schemes* and *series-parallel contact schemes*. The only difference of contact schemes from switching networks is that their underlying graphs are *undirected*. Series-parallel schemes have an additional restriction that these graphs (also undirected) must be series-parallel (this model coincides with that of formulas, i.e. fanout-1 circuits). Results of Karchmer and Wigderson [17] imply that monotone boolean parallel-sequential contact schemes computing the k-walk polynomial f must have $n^{\Omega(\log k)}$ edges. Potechin [18] has proved that the directed version of f (for directed graphs) requires this number of edges in monotone contact schemes. It is also known that f requires bounded-depth (even non-monotone) circuits of super-polynomial size; see Rossman's paper [19] and the literature therein. On the other hand, the switching network of Bellman–Ford–Moore computes both directed and undirected versions of f using only $O(kn^2)$ edges. This shows that both features of switching networks arising from dynamic programming algorithms—directed edges and overlap of sub-networks—are essential.

We now use Theorem 1 to show that, over the $(\min, +)$ semiring, the upper bound given by the Bellman–Ford–Moore lemma cannot be substantially improved. To show that the k-walk polynomial has large cover number, we will use the following classical result proved by Erdős and Gallai [4] using a method due to Dirac [20]: for $l \geqslant 1$, every graph of average degree more than $l-1$ contains a path of length l. Since $\binom{m}{2} - (l-1)m/2 = m(m-l)/2$, this result can be re-stated as:

Erdős–Gallai Theorem (Erdős and Gallai [4]). *For $l \geqslant 1$, at least $m(m-l)/2$ edges must be removed from K_m in order to destroy all paths of length l.*

Note that this bound cannot be improved. Indeed, if $m = ql$ is a multiple of l, then we can split the nodes of K_m into q disjoint sets of size l, and remove all edges lying between these sets. The resulting graph has no paths of length l, and we have removed only $\binom{q}{2}l^2 = m(m-l)/2$ edges.

To spare parenthesis, we say that a function $f(n)$ is *at least about* $g(n)$, if $f(n) = \Omega(g(n))$.

Theorem 2. *Every switching network computing the k-walk polynomial over the* $(\min,+)$ *semiring requires at least about* $kn(n-k)$ *switches.*

Proof. Set to 0 all variables $x_{i,j}$ such that $\{i,j\}\cap\{1,n\}\neq\emptyset$. That is, we set to 0 all variables incident with the start node 1, or with the target node n of K_n. Let f be the resulting (tropical) polynomial. Its variables correspond to the edges of the complete graph K_{n-2} on $\{2,\ldots,n-1\}$ and, over the $(\min,+)$ semiring, the polynomial computes the function

$$f(x)=\min\left\{x_{i_1,i_2}+x_{i_2,i_3}+\cdots+x_{i_{k-2},i_{k-1}}\right\}, \tag{3}$$

where $i_1,\ldots,i_{k-1}$ are not necessarily distinct nodes in $\{2,\ldots,n-1\}$. that is, each sum of f corresponds to a walk in K_{n-2} of length $k-2$. Observation 2 implies that $l(f)=f(1,\ldots,1)\geqslant k-2$.

To lower bound the cover number $c(f)$ of f, let Y be a set of $|Y|=c(f)$ variables of f such that every sum of f contains at least one of these variables. For every *path* of length $k-2$ in K_{n-2} (no loops and no repeated edges), there is a corresponding sum in (3) whose variables correspond to the edges of that path. This sum must contain at least one variable in Y. Thus, removal from K_{n-2} of all edges corresponding to variables in Y destroys all paths of length $k-2$ in K_{n-2}. When applied with $m=n-2$ and $l=k-2$, the Erdős–Gallai theorem gives $c(f)=|Y|\geqslant m(m-l)/2=(n-2)(n-k)/2$. Since $l(f)\geqslant k-2$, Theorem 1 implies that every switching network computing f over the $(\min,+)$ semiring must have at least $l(f)\cdot c(f)$ switches, which is at least about $kn(n-k)$. □

5. Matrix multiplication

We now consider the problem of computing the sum of all entries of the product of two matrices over the tropical semiring:

$$M_n(x,y)=\sum_{i,j\in[n]}\min_{k\in[n]}\{x_{i,k}+y_{k,j}\}.$$

Theorem 3. *The minimum number of switches in a switching network computing M_n over the* $(\min,+)$ *semiring is* $2n^3$.

Proof. The upper bound $2n^3$ is trivial, since each minimum $g_{i,j}=\min_k\{x_{i,k}+y_{k,j}\}$ can be computed using a bunch of $2n$ switches. To prove the lower bound, we will directly apply Theorem 1 to the (tropical) polynomial $f=M_n$ itself.

Since $f(1,1,\ldots,1)=2n^2$, the length of f is $l(f)\geqslant 2n^2$. On the other hand, in order to force f to output $+\infty$ (recall that $+\infty$ is the "zero element" $\mathsf{0}$ in the tropical semiring), there must be at least one pair $i,j\in[n]$ such that the minimum $g_{i,j}$ outputs $+\infty$. Thus, at least n variables must be set to $+\infty$, implying that $c(f)\geqslant n$. By Theorem 1, any switching network computing $f=M_n$ over the $(\min,+)$ semiring must have at least $l(f)\cdot c(f)\geqslant 2n^3$ switches. □

6. Can Markov's theorem be improved?

Markov's theorem (as well as its extension given in Theorem 1) can give a nontrivial lower bound only if *all* chains (s-t paths) in a switching network are long enough; by the *length* of a chain we mean the number of its switches (labeled edges). So, a natural question is: can

Markov's theorem be modified so that it works also when networks *have* short chains? The next lemma shows that, in general, this is *not* possible. That is, there exists no analogue of Markov's theorem when short paths are present.

Define an *l-cut* in a switching network G to be a set of its switches whose removal destroys all chains in G with l or more switches; shorter chains may survive! Let $c_l(G)$ denote the minimum size of an l-cut in G. Let also $e(G)$ denote the total number of switches in G. In these terms, Markov's theorem states:

- If *all* chains in G have at least l switches , then $e(G) \geqslant l \cdot c_l(G)$.

But what if G *has* some short chains, shorter than l (we do not need to destroy them)—will then the total number of switches always be at least about l *times* $c_l(G)$?

It turns out that the answer depends on how close is the "critical" length l to the depth d of the network. The following lemma shows that no analogue of Markovs theorem exists, if $l \ll d$.

Lemma 2. *There exists a sequence of constant-degree directed acyclic graphs G_n on n nodes with the following property: for every constant $0 \leqslant a < 1$ there is a constant $b > 0$ such that $e(G_n) \leqslant b \cdot c_l(G_n)$ holds for $l = n^a$.*

That is, even a *constant* portion of all edges must be removed in order to remove only very long s-t paths.

Proof. Using expander graphs, the existence of a sequence of directed acyclic graphs H_n of constant maximum degree d on $n = m2^m$ nodes is constructed in [21] with the following property:

- For every constant $0 \leqslant a < 1$ there is a constant $b > 0$ such that, if any subset of at most bn nodes is removed from H_n, the remaining graph contains a path of length at least 2^{am}.

Take now two new nodes s and t, and draw an edge from s to every node of H_n, and an edge from every node of H_n to t. The resulting graph G_n still has at most $2n + dn = O(n)$ edges, and has the desired property:

Claim 1. For every constant $0 \leqslant a < 1$, there is a constant $b' > 0$ such that, if any subset of at most $b'n$ edges is removed from G_n, the remaining graph contains an s-t path with $l = 2^{am}$ or more edges.

To show this, call the nodes of H_n *inner* nodes of G_n. Remove any subset of at most $b'n$ edges from G_n, where $b' = b/2$. After that, remove an inner node if it was incident to a removed edge. Note that at most $2b'n = bn$ inner nodes are removed in this way. None of the edges incident to surviving nodes was removed. In particular, each surviving inner node is still connected to *both* nodes s and t. By the above property of H_n, there must remain a path of length 2^{am} consisting entirely of surviving inner nodes. Since the endpoints of this path survived, the path can be extended to an s-t path in G_n. □

By the previous lemma, no analogue of Markov's theorem exists when $l \ll d$. Still, such an analogue exists when the "critical" length l is indeed very close to the depth d of a network, i.e. when $l \geqslant d - O(1)$.

Lemma 3. *Let G be a switching network of depth d. Then, for every $1 \leqslant l \leqslant d$, we have that $e(G) \geqslant s \cdot c_l(G)$, where s is the maximal integer sucht that $s \leqslant l/(d-l+1)$.*

Note that for $l = d$, the lower bound on $e(G)$ is $l \cdot c_l(G)$, as in Markov's theorem. But already for $l \leqslant d/2$, the lower bound is at most $2 \cdot c_l(G)$.

Proof. The lower bound clearly holds, if no chain has l switches because then $c_l(G) = 0$. So, assume that at least one chain contains l of more switches. Associate with every node u the *maximum* number l_u of switches in a path from the source node s to u; hence, $l_s = 0$. For the target node t we have $l_t = m$ for some $l \leqslant m \leqslant d$. Split the set of nodes of G into layers $V_0, V_1, \ldots, V_m$, where $V_i = \{u \colon l_u = i\}$. Let (u, v) be an edge in G with $u \in V_i$ for some i, and let V_j be the layer containing the endpoint v. Then clearly $j \geqslant i$. If $j = i$, then the edge (u, v) must be a rectifier, because otherwise we would have that $l_v \geqslant l_u + 1$. So, only rectifiers can lie within each of the sets V_i, and every switch must go from one layer V_i to another layer V_j with $j > i$. We say that an edge e *leaves* a node u, if $e = (u, v)$ for some node v. Set $r := d - l + 1$, and let π be a chain with at least l switches.

Claim 2. For every subset of r layers, at least one switch of π must leave at least one of these layers.

Proof. The chain π has at most d edges, at least l of which are switches. Each of these switches must leave some of the m layers $V_0, \ldots, V_{m-1}$, and different switches of π must leave different layers. So, if none of the r given layers is left by a switch of π, then π can have at most $m - r \leqslant d - r = d - (d - l + 1) = l - 1$ switches in total, a contradiction. □

Let s be the maximal integer such that $sr \leqslant m$, split the layers into s subsequent blocks, and let B_j be the set of nodes of the j-th block. Let C_j be the set of all switches leaving the nodes of the j-th block B_j. It is clear that the sets C_j are disjoint (one edge can leave only one node). Moreover, by Claim 2, each C_j is an l-cut in G (contains at least one switch in every chain with l of more switches). Hence, $e(G) \geqslant \sum_{j=1}^{s} |C_j| \geqslant s \cdot c_l(G) \geqslant l/(d-l+1) \cdot c_l(G)$, as desired. □

Over the boolean semiring, the k-walk polynomial turns into a well-known *distance-k connectivity function*. This is a monotone boolean function which, given a subgraph of K_n, decides whether there is a 1-to-n path with at most k edges. By the Bellman–Ford–Moore lemma, this function can be computed by a monotone boolean switching network with $O(kn^2)$ edges. An additional feature of this network is that its depth is only k. Using Lemma 3, we can show that any monotone boolean switching network of depth k must have almost this number of switches.

Theorem 4. *Every monotone boolean switching network of depth k for the distance-k connectivity function must have at least about $k(n-k)n$ switches.*

Proof. Let G' be a monotone boolean switching network computing the distance-k connectivity function f. Since $l(f) \leqslant 3$ holds over the boolean semiring, Markov's theorem cannot yield any lower larger than $\Omega(n^2)$. So, we additionally assume that G' has depth at most k.

Replace by rectifiers all switches labeled by variables $x_{i,j}$ such that $\{i, j\} \cap \{1, n\} \neq \emptyset$, and let G be the resulting network. Minterms of f correspond to 1-to-n paths with at most k edges in K_n. By Lemma 1, for every such path, there must be a chain in G' whose switches are labeled exactly the by edges of this path. So, for every path $(i_1, i_2, \ldots, i_l)$ in $\{2, \ldots, n-1\}$,

there must be a chain π in G' whose set of labels is $\{x_{1,i_1}, x_{i_1,i_2}, \ldots, x_{i_{l-1},i_l}, x_{i_l,n}\}$, with all nodes $i_j \in \{2, \ldots, n-1\}$ distinct. In G, the set of labels of π is then $\{x_{i_1,i_2}, \ldots, x_{i_{l-1},i_l}\}$. Thus, for every path of length $l = k-2$ in K_m (for $m = n-2$) there must be a chain with l switches in the network G labeled by the edges of this path. Together with the Erdős–Gallai theorem, this implies that $c_l(G) \geqslant (n-2)(n-k)/2$ for $l = k-2$. Since, by our assumption, the network G has depth $d \leqslant k$, Lemma 3 implies that $e(G) \geqslant l/3 \cdot c_l(G)$, which is at least about $k(n-k)n$. □

7. Conclusion and open problems

We extended Markov's theorem to arbitrary semirings of zero characteristic, including the tropical $(\min, +)$ semiring. We then applied this extension to the k-walk problem f: given an assignment of nonnegative weights to the edges of K_n, find the minimum weight of a walk of length k from node 1 to node n. The Bellman–Ford–Moore dynamic programming algorithm gives a switching network with $O(kn^2)$ switches solving this problem. Using the Erdős–Gallai theorem about long paths in graphs, we showed that this network is optimal: about $kn(n-k)$ switches are necessary in any tropical $(\min, +)$ switching network solving this problem.

Note, however, that this lower bound degrades severely as k approaches n, just because then the Erdős–Gallai lower bound on $c(f)$ is only linear in n. So, an interesting problem is to prove that also then $\Omega(n^3)$ switches are necessary. Still, the most interesting in our context problem is to prove an $\Omega(kn^2)$ lower bound for switching networks computing f over the *boolean* semiring. Our Markov-type argument fails here by another reason: over this semiring, f has very small length $l(f) \leqslant 3$. In this semiring, we were only able to prove a lower bound $\Omega(kn^2)$ for depth-k networks. In Lemma 2 we argued that, if the depth is unbounded, then the presence of short chains is an *inherent* difficulty, and no analogue of Markov's theorem exists in this case.

The next open problem concerns a natural generalization of the model we considered. In the tropical switching networks considered above, switches are labeled by single variables x_i. Thus, already a switches are necessary to compute the single term ax_i. One can extend the model of $(\min, +)$ switching networks by allowing the labels of switches to be arbitrary linear combinations $\sum_{i \in S} a_i x_i$ with integer coefficients. Albeit the Bellman–Ford–Moore $(\min, +)$ switching network does not use this additional feature, it may be helpful for some other minimization problems. Consider, for example, the problem

$$f(x_1, \ldots, x_n) = \min_{a \in \mathbb{N}^n} \left\{ \sum_{i=1}^n a_i x_i \colon \sum_{i=1}^n a_i = k \right\}.$$

Since $l(f) = f(1, \ldots, 1) = k$ and $c(f) = n$ (all n variables must be set to $+\infty$ to force $f = +\infty$), every (ordinary) $(\min, +)$ switching network for f must have at least kn switches. But since $f(x) = \min\{kx_1, \ldots, kx_n\}$, already n switches are enough for extended networks. So, it would be interesting to know whether extended $(\min, +)$ switching networks for the k-walk polynomial must still be of size $\Omega(kn^2)$. Note that Theorem 1 fails for extended $(\min, +)$ switching networks. The reason is that then the number of switches in a chain may be much smaller than $l(f)$.

Acknowledgments

We are thankful to both referees for very helpful comments. The first author also thanks David Eppstein and Igor Sergeev for interesting discussions.

References

[1] L. Ford, Network flow theory, Tech. Rep. P-923, The Rand Corp. (1956).

[2] R. Bellman, On a routing problem, Quarterly of Appl. Math. 16 (1958) 87–90.

[3] E. Moore, The shortest path through a maze, in: Proc. Internat. Sympos. Switching Theory, Vol. II, Harvard Univ. Press 1959, 1957, pp. 285–292.

[4] P. Erdős, T. Gallai, On maximal paths and circuits in graphs, Acta Math. Acad. Sci. Hungar. 10 (1959) 337–356.

[5] L. Kerr, The effect of algebraic structure on the computation complexity of matrix multiplications, Ph.D. thesis, Cornell Univ., Ithaca, N.Y. (1970).

[6] R. Floyd, Algorithm 97, shortest path, Comm. ACM 5 (1962) 345.

[7] S. Warshall, A theorem on boolean matrices, J. ACM 9 (1962) 11–12.

[8] V. Pratt, The power of negative thinking in multiplying boolean matrices, SIAM J. Comput. 4 (3) (1975) 326–330.

[9] M. Paterson, Complexity of monotone networks for boolean matrix product, Theoret. Comput. Sci. 1 (1) (1975) 13–20.

[10] K. Mehlhorn, Z. Galil, Monotone switching circuits and boolean matrix product, Computing 16 (1-2) (1976) 99–111.

[11] S. Jukna, Lower bounds for tropical circuits and dynamic programs, Theory of Comput. Syst. 57 (1) (2015) 160–194.

[12] R. Krichevski, Complexity of contact circuits realizing a function of logical algebra, Doklady Akad. Nauk SSSR 151 (4) (1963) 803–806, English translation in: Soviet Physics Doklady 8 (1963), 770–772.

[13] G. Hansel, Nombre minimal de contacts de fermeture necessaires pour realiser une function booleenne symetrique de n variables, C. R. Acad. Sci. 258 (25) (1964) 6037–6040.

[14] A. Markov, Minimal relay-diode bipoles for monotonic symmetric functions, Problemy Kibernetiki 8 (1962) 117–121, English transl. in: Problems of Cybernetics 8 (1964), 205–212.

[15] E. Moore, C. Shannon, Reliable circuits using less reliable relays, J. Franklin Inst. 262 (3) (1956) 281–297.

[16] J. Robacker, Min-max theorems on shortest chains and disjoint cuts of a network, Tech. Rep. RM-1660, The Rand Corp. (1956).

[17] M. Karchmer, A. Wigderson, Monotone circuits for connectivity require super-logarithmic depth, SIAM J. Discrete Math. 3 (2) (1990) 255–265.

[18] A. Potechin, Bounds on monotone switching networks for directed connectivity, in: Proc. of 51th Ann. IEEE Symp. on Foundations of Comput. Sci., FOCS, 2010, pp. 553–562.

[19] B. Rossman, Formulas vs. circuits for small distance connectivity, in: Proc. of 46-th ACM Symp. on Theory of Computing, STOC, 2014, pp. 203–212.

[20] G. Dirac, Some theorems on abstract graphs, Proc. London Math. Soc. 2 (1952) 69–81.

[21] G. Schnitger, On depth-reduction and grates, in: Proc. of 24th Ann. IEEE Symp. on Foundations of Comput. Sci., FOCS, 1983, pp. 323–328.