Theoretical Computer Science 628 (2016) 101–109

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Note

On the optimality of Bellman–Ford–Moore shortest path algorithm ☆

Stasys Jukna [*,1], Georg Schnitger

Institute of Computer Science, Goethe University Frankfurt, Frankfurt am Main, Germany

ARTICLE INFO

Article history:
Received 7 August 2015
Received in revised form 6 March 2016
Accepted 8 March 2016
Available online 10 March 2016
Communicated by T. Erlebach

Keywords:
Shortest paths
Matrix multiplication
Dynamic programming
Tropical semiring
Lower bounds

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.

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

☆ Research supported by the DFG grant SCHN 503/6-1.
* Corresponding author.
E-mail addresses: jukna@thi.informatik.uni-frankfurt.de (S. Jukna), schnitger@thi.informatik.uni-frankfurt.de (G. Schnitger).
[1] Affiliated with Institute of Mathematics and Informatics, Vilnius University, Vilnius, Lithuania.

http://dx.doi.org/10.1016/j.tcs.2016.03.014
0304-3975/

shortest k-walk problem: given an assignment of nonnegative weights to the edges of the complete graph on $[n]=\{1,\dots,n\}$, find the minimum weight of a walk of length k from node $s=1$ to the node $t=n$. Recall that a *walk* is an alternating sequence of nodes and connecting edges. A walk can travel over any node (except 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 $\le k$, and every s-t path of length $\le k$ 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 such 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 one to also prove 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 $(R,+,\times,0,1)$ be a semiring with “sum” $(+)$ and “product” $(\times)$ operations, additive identity (“zero element”) 0, and multiplicative identity 1 (“unit element”). We only consider commutative semirings, and assume the “annihilation” property $x\times 0=0$ of the zero element. Recall that a (multivariate) polynomial over R is a formal expression of the form

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

where $A\subset\mathbb{N}^n$ is a finite set of nonnegative integer vectors, and $c_a\ge 1$ are integer coefficients. The coefficients c_a are not necessarily elements of R: they only indicate the number of times the corresponding 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 $f(r)=f(r_1,\dots,r_n)$ 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 $R=\{0,1\}$, $x+y:=x\vee y$, $x\times y:=x\wedge y$, $0:=0$, and $1:=1$, whereas in the *tropical* $(\min,+)$ semiring, we have $R=\mathbb{N}\cup\{+\infty\}$, $x+y:=\min\{x,y\}$, $x\times y:=x+y$, $0:=+\infty$, and $1:=0$. Hence, over these two semirings, the functions defined by the polynomial (2) are, respectively,

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