Electronic Colloquium on Computational Complexity, Report No. 146 (2018)

Shortest path length with bounded-alternation $(\min, +)$ formulas

Meena Mahajan[1], Prajakta Nimbhorkar[2], and Anuj Tawari[1]

[1]The Institute of Mathematical Sciences, HBNI, Chennai, India. {meena,anujvt}@imsc.res.in
[2] Chennai Mathematical Institute, India. prajakta@cmi.ac.in

August 13, 2018

Abstract

We study bounded-depth $(\min, +)$ formulas computing the shortest path polynomial. For depth $2d$ with $d \geq 2$, we obtain lower bounds parameterized by certain fan-in restrictions on $+$ gates except those at the bottom level. For depth 4, in two regimes of the parameter, the bounds are tight.

1 Introduction

For many discrete minimization problems, the most natural and intuitive way to solve them is using a $(\min, +)$ circuit. In this model, the inputs are either variables or constants from the underlying semiring. The computation is performed using the operations $\min, +$. Such circuits can compute any function expressible as the minimum over several linear polynomials with non-negative integer coefficients. The complexity measures associated with such a circuit are its size and depth, which capture the number of operations and the maximal distance between an input and output respectively.

This model captures the complexity of "pure" dynamic programming algorithms; see for instance [5, 6, 7, 8].

In this work, we consider the well-studied problem SHORTEST-PATH: Given a graph on vertex set $[n] = \{1, 2, \ldots, n\}$ with an assignment of non-negative

ISSN 1433-8092

integer weights to its edges, we want to find a $(\min, +)$ formula which computes the weight of the shortest path from $s = 1$ to $t = n$.

Our Results

A naive approach toward solving this problem would be as follows: Compute the weights of all possible paths from 1 to n and take the minimum of these weights. This yields a $(\min, +)$ circuit of $2^{O(n \log n)}$ size. For depth 2 $(\min, +)$ circuits, it is easy to see that this naive approach is indeed the best possible. We show that this is also the case for depth 3 circuits (Theorem 8). We would expect the lower bounds to degrade as we allow more depth, and the question we are interested in is, how fast do they degrade? We provide partial answers to this question, exploring restricted cases of $(\min, +)$ formulas (circuits of fan-out 1). We study restrictions of two types. In the first restriction, except at the bottom-most level, $+$ gates do not have very large fan-in. (Note that since paths in an n-vertex graph have at most $n - 1$ edges, the fan-in of useful $+$ gates will not exceed n.) Our lower bound is parameterized by the depth d and the permitted $+$ fan-in k (Theorem 10). For the depth-4 case, the lower bound is tight when $k = O(1)$ and also when $k = O(\sqrt{n})$. In the second restriction, which applies only to depth-4 formulas, most $+$ gates just below the top gate have fan-in 2. Our lower bound here is parameterized by the the number of $+$ gates with fan-in exceeding 2 (Theorem 13).

Note that any constant-depth circuit can be simulated by a formula of the same depth, with at most a polynomial blow-up in size. Therefore, our result also implies an exponential lower bound for the corresponding subclass of constant depth $(\min, +)$ circuits.

Background

Many known algorithms for solving SHORTEST-PATH are essentially recursively constructed $(\min, +)$ circuits. For instance, the classical dynamic programming algorithm by Bellman and Ford [1, 3] gives a bounded fan-in circuit of $O(n^3)$ size and depth $\Theta(n)$. Whether $\Omega(n^3)$ is necessary is still open. However the Bellman-Ford algorithms produce skew circuits, and for skew circuits, this bound is shown in [9] to be optimal. A divide-and-conquer approach gives a bounded fan-in circuit of poly(n) $(O(n^4))$ size and depth $\Theta(\log^2 n)$.

A natural question to ask is whether one can prove strong size lower bounds for bounded depth $(\min, +)$ circuits or formulas. This is a first step towards proving tight lower bounds for general circuits (and hence determining the exact complexity of the problem).

It is known that over the Boolean semiring, any bounded fan-in monotone $(\vee, \wedge)$ circuit for REACH must have depth $\Omega(\log^2 n)$ [10]. Using a natural mapping from $(\min, +)$ semiring to the boolean semiring, this result also implies that any bounded fan-in $(\min, +)$ circuit for STCON must have $\Omega(\log^2 n)$ depth, no matter what size. The divide-and-conquer approach shows that this depth lower bound is tight.

In this work, we consider the alternation depth of $(\min, +)$ circuits. This corresponds to allowing semi-unbounded fan-in and even unbounded fan-in in some cases. In this setting, exponential lower bounds are easy to prove. One way to do so is to use the reduction (via projections) from parity to REACH, and use known lower bounds for (non-monotone) circuits for parity [4]; see Proposition 1. We are looking for lower bounds better than those obtained this way.

In [2], such small-depth lower bounds are obtained for the decision version of "short distance connectivity": is there a path using at most k edges? These lower bounds can also be transferred to $(\min, +)$ circuits computing the corresponding optimisation problem: Compute the weight of the shortest path which uses at most k edges.

2 The Computation Model

Fix any semiring $\mathcal{S} = (S, \oplus, \otimes)$. A circuit over $\mathcal{S}$ is a directed acyclic graph with a unique designated sink node called the output node. Source nodes, also called leaf nodes, are labeled by variables or elements of S. Each internal node is labeled by one of the semiring operations $\oplus$ or $\otimes$, and is also called a gate of the circuit. If the underlying graph is a tree, then the circuit is called a formula. The size of a circuit is the number of nodes and edges in it. For a formula F, it is equivalent and often more convenient to consider the number of leaves $L(F)$ as the formula size. The depth of a circuit is the length of the longest path from a leaf to the output node. The fan-in of a gate is the number of incoming edges. The alternation depth of a circuit is the maximum, over all leaf-to-output paths, of the number of maximal blocks of gates of the same type along the path. With $\oplus$ and $\otimes$ being associative, we can transform any circuit C into an equivalent unbounded fan-in circuit C' where the parent of each gate is a gate of a different type; the alternation depth of C is then just the depth of C'. In this paper, when we say depth of a circuit, we mean alternation depth.

A circuit C syntactically *produces* a polynomial p_C over $\mathcal{S} = (S, \oplus, \otimes)$ in a natural way; at a leaf, the polynomial produced is the leaf label, and at intermediate nodes, the polynomial produced is obtained by combining polynomials produced at the children using the operation labeling the gate. Using the distributivity of $\otimes$ over $\oplus$, the polynomial produced at the output gate can be represented as a $\oplus$ sum of monomials, where within each monomial we use the $\otimes$ product.

A circuit C *computes* a polynomial p if the polynomial p_C produced by C agrees with the polynomial p at all input settings. Over the arithmetic semiring $A = (\mathbb{N}, +, \times)$, computing and producing are equivalent in terms of the size of the circuit required. However, over other semirings, there can be significant differences. In particular, this is the case for the tropical semiring Min, the focus of this paper. We use the notation $\mathcal{S}(p)$ to denote the size of the smallest circuit computing (not producing) p over the semiring $\mathcal{S}$.

The tropical semiring Min is the semiring $\mathrm{Min} = (\mathbb{N}, \min, +)$, with 0 being the identity for $+$ and ∞ the identity for min. A circuit over Min, with variables $x_1, \ldots, x_n$, produces a polynomial of the form $\min\{\ell_1, \ell_2, \ldots, \ell_t\}$ where each monomial ℓ_r is of the form $c_0 + c_1x_1 + \ldots + c_nx_n$, for non-negative integers c_i.

For a polynomial $p(X)$, $\mathrm{Mon}(p)$ denotes the set of monomials of p. Let $\preceq$ be the following (partial) ordering amonst monomials over the variable set $X = \{x_1, \ldots, x_n\}$: $c_0 + c_1x_1 + \ldots + c_nx_n \preceq d_0 + d_1x_1 + \ldots + d_nx_n$ if $c_i \leq d_i$ for all i. Then $\mathrm{Mon}_{le}(p)$, the *lower envelope* of monomials of p, denotes the set of those monomials in $\mathrm{Mon}(p)$ that are minimal with respect to $\preceq$, and the lower envelope of p, denoted p_{le}, is the minimum (the $\oplus$ sum) of these monomials. Over the semiring Min, if polynomials p and q have the same lower envelope, then they compute the same values everywhere. Thus for a polynomial p, $\mathrm{Min}(p)$ is the size of the smallest $(\min, +)$ circuit producing a polynomial whose lower envelope is p_{le}.

Computation over the semiring Min lies somewhere in between monotone Boolean computation and monotone arithmetic computation. For any polynomial p described as an $\oplus$ sum of $\otimes$ monomials, the following relation holds: $B(p) \leq \mathrm{Min}(p) \leq A(p)$. Here B and A denote the Boolean and arithmetic semirings respectively.

In general, a $(\min, +)$ circuit may have constants from $\mathbb{N}$ at the leaves. However, for computing monic polynomials with no constant term, using constants in the formula cannot really help. In the rest of the paper, we always assume that the $(\min, +)$ circuits or formulas are constant-free and have only variables at the leaves. For convenience in describing the upper bounds, we may use the values $0, \infty$ at the leaves, but these can be propagated upwards without increasing the size.

For more details about computation over semirings in general and the Min in particular, the reader is referred to [5, 6, 7].

3 The Shortest Path problem

3.1 Problem Definition

Let G be a directed graph on a set of n vertices. Edges are labeled with costs that are non-negative and integer-valued. The cost of a path is the total cost of all edges in the path. For designated source vertex s and target vertex t, the SHORTEST-PATH$_n$ problem is to find the minimum cost of a path from s to t. (The subscript n indicates the graph size; we drop it when implicit from context.)

To compute SHORTEST-PATH by $(\min, +)$ circuits or formulas, we assume that G is the complete directed graph with vertex set $[n]$, and for each $i, j \in [n]$ with $i \neq j$, the variable $x_{i,j}$ is the cost of the edge directed from vertex i to vertex j. All variables take values in the set $\mathbb{N} \cup \{\infty\}$. While solving SHORTEST-PATH on any input graph, we will set $x_{i,j}$ to be the actual cost of the edge from i to j. We will set $x_{i,j}$ to ∞ for edges absent in G. We may also assume that there are variables $x_{i,i}$ for $i \in [n]$; these variables are all set to 0. With these conventions, the following expression is the desired SHORTEST-PATH value.

$$\textsc{shortest-path} = \min \left\{ \mathrm{cost}(\rho) \mid \rho \text{ is a simple } s\text{-to-}t \text{ path} \right\},$$

where

$$\mathrm{cost}(\rho) = \sum_{\langle i,j \rangle \in \rho} x_{i,j}.$$

We denote by REACH$_n$ the decision problem of deciding whether an n-vertex graph has an s-to-t path.

3.2 Known upper bounds

Viewed as a polynomial over the semiring Min, the SHORTEST-PATH polynomial has the set of monomials

$$\mathrm{Mon}(\textsc{shortest-path}) = \left\{ \mathrm{cost}(\rho) \mid \rho \text{ is a simple } s\text{-to-}t \text{ path} \right\}.$$

It is known that any circuit producing the SHORTEST-PATH polynomial must be exponentially large [6]. However, to compute this polynomial, it suffices to design a circuit producing a polynomial whose lower envelope has exactly the monomials in Mon(SHORTEST-PATH), and this is a considerably easier task.

Incremental dynamic programming, extending sub-paths by a single edge at a time, gives a bounded fan-in circuit of $O(n^3)$ size and depth $\Theta(n)$.

Dynamic programming, merging roughly equal-length sub-paths (equivalently, dividing each path roughly mid-way), gives a bounded fan-in circuit of poly(n) ($O(n^4)$) size and depth $\Theta(\log^2 n)$.

In this paper, we are concerned with alternation depth, or equivalently, unbounded fan-in circuits and formulas. Dynamic programming, where r sub-paths are merged in each merge step for some parameter r, gives a recurrence as follows. Let $f(i,j,\ell)$ denote the minimum cost of a path from i to j amongst all paths with at most ℓ edges. Our goal is to compute $f(s,t,n-1)$, we have $f(i,i,\ell) = 0$ forall ℓ, and we have the following expressions for $i \neq j$:

$$\begin{aligned} f(i,j,0) &= \infty \\ f(i,j,1) &= x_{i,j} \\ f(i,j,\ell) &= \min_{k_1,\ldots,k_{r-1}} \left\{ f\left(i, k_1, \frac{\ell}{r}\right) + f\left(k_1, k_2, \frac{\ell}{r}\right) + \ldots + f\left(k_{r-1}, j, \frac{\ell}{r}\right) \right\} \end{aligned}$$

The depth of recursion is given by $p = \frac{\log n}{\log r}$. Each level of recursion corresponds to a layer of min gates followed by a layer of + gates. Thus the corresponding circuit has depth $d = 2p$. Conversely, to get a $(\min, +)$ circuit of depth $d = 2p$, it suffices to take $r = n^{\frac{2}{d}}$. The fan-in of the min gates is at most n^{r-1} whereas the fan-in of + gates is at most r. It is easily verified that this gives rise to a circuit of size $\exp(O(n^{\frac{2}{d}} \log n))$, or a formula of size $\exp(O(dn^{\frac{2}{d}} \log n))$. In particular, the depth-2 formula is of size $\exp(O(n \log n))$.

3.3 Lower bounds implied from known work

The SHORTEST-PATH polynomial, interpreted over the Boolean ring B, decides REACH. Hence $B(\text{SHORTEST-PATH}) \leq \text{Min}(\text{SHORTEST-PATH})$; any monotone Boolean circuit size lower bound for REACH is also a lower bound for Min(SHORTEST-PATH).

For bounded (alternation) depth, one lower bound for Boolean circuits for REACH is derived from the lower bound for circuits for parity [4]. Although this is folklore, for completeness we include a full proof here.

Proposition 1 (folklore) *Depth d Boolean circuits (and hence also monotone Boolean circuits) for* REACH$_{2n}$ *must be of size* $\exp(\Omega(n^{\frac{1}{d-1}}))$. *Hence any depth d* $(\min, +)$ *circuit computing* SHORTEST-PATH$_{2n}$ *must have size* $\exp(\Omega(n^{\frac{1}{d-1}}))$.

Proof. Given n bits $y_1, \ldots, y_n$, the PARITY$_n$ function outputs 1 if an odd number of y_i's are set to 1, and 0 otherwise. In [4] it is shown that Boolean circuits for PARITY$_n$ with alternation depth d must have size $\exp(\Omega(n^{\frac{1}{d-1}}))$.

The PARITY_n function reduces to REACH_{2n} by projections as follows. The REACH_{2n} instance is a graph G with $2n$ vertices, and it is convenient to think of the vertex set as $\{(i,b) \mid i \in [n-1], b \in \{0,1\}\} \cup \{(0,0),(n,1)\}$, with source vertex $s = (0,0)$ and sink vertex $t = (n,1)$. The edges of the graph are determined as follows: There is an edge from a vertex $(i-1,b)$ to (i,b) for $b \in \{0,1\}$ if and only if $y_i = 0$. Similarly, there is an edge from $(i-1,b)$ to $(i,1-b)$ for $b \in \{0,1\}$ if and only if $y_i = 1$. That is, $x_{(i-1,b),(i,b)} = \bar{y}_i$, and $x_{(i-1,b),(i,1-b)} = y_i$. If $i \neq j-1$, then $x_{(i,b),(j,b')} = 0$. It is easy to see that there is a path from $(0,0)$ to $(n,1)$ in G if and only if $y_1 + \ldots + y_n \equiv 1 \bmod 2$. Hence any Boolean circuit for REACH_{2n}, with alternation depth d, must also have size $\exp(\Omega(n^{\frac{1}{d-1}}))$. $\square$

In [2], the restriction of REACH to short path lengths is studied. We denote by $\text{SHORTEST-PATH}_{n,k}$ the restriction of the SHORTEST-PATH_n polynomial to the monomial set

$$\{\text{cost}(\rho) \mid \rho \text{ is a simple } s\text{-to-}t \text{ path of length at most } k\},$$

and let $\text{REACH}_{n,k}$ denote the corresponding decision version (decide whether the graph has an s-to-t path of length at most k). In [2], the following result is shown:

Proposition 2 (Theorem 1 in [2]) *1. For any $k(n) \leq n^{1/5}$ and any $d = d(n)$, any depth-d circuit computing $\text{SHORTEST-PATH}_{n,k}$ must have size $n^{\Omega(k^{1/d}/d)}$.*

2. For any $k(n) \leq n$ and any $d = d(n)$, any depth-d circuit computing $\text{SHORTEST-PATH}_{n,k}$ must have size $n^{\Omega(k^{1/5d}/d)}$.

Note that this bound applies for any Boolean circuit, not just monotone circuits. At $k = n$, it gives a lower bound of $\exp(\Omega(\frac{n^{1/5d}\log n}{d}))$ for depth-d circuits computing REACH_n. Hence

Corollary 3 *Any depth-d $(\min,+)$ circuit for SHORTEST-PATH must have size $\exp(\Omega(\frac{n^{1/5d}\log n}{d}))$.*

4 New Lower Bounds

The following fact is easy to verify.

Fact 4 *Let $P(n)$ denote the number of distinct st paths in the complete n-vertex directed graph. Then $P(n) = 2^{\Theta(n\log n)}$. More specifically,*

$$2((n-2)!) < P(n) < e((n-2)!).$$

For any formula F computing SHORTEST-PATH, as discussed in Section 2, the polynomial produced by F must have exactly the monomials of SHORTEST-PATH in its lower envelope. A direct graph-theoretic way to see this is given in the following proposition. In its proof, as well as later in the paper, we use the notation G_ρ to denote the graph with only the edges of ρ, for any simple st path ρ.

Proposition 5 *Let F be a formula computing* SHORTEST-PATH*. Let p be the polynomial syntactically produced by F, and $Mon_{le}(p)$ be the set of minimal monomials of this polynomial (the lower envelope). Then $Mon_{le}(p)$ equals the set of monomials of* SHORTEST-PATH*, $\{cost(\rho) \mid \rho \text{ is a simple } s\text{-to-}t \text{ path}\}$.*

Proof. Let ρ be any simple st path, and let G_ρ be the graph with only the edges of ρ. On setting $x_{i,j}$ to 1 for $(i,j) \in \rho$ and to ∞ for all other edges, F should evaluate to $|\rho|$. So at least one linear form (recall, in the semiring Min, monomials are linear forms) should use only the variables from ρ (otherwise it evaluates to ∞). However, if for any such linear form, ℓ, $\text{var}(\ell)$ is a proper subset of $\text{var}(\rho)$, then some variable x_{uv} with value 1 does not appear in ℓ. Deleting edge uv from G_ρ (changing the value of x_{uv} to ∞) disconnects s and t in the resulting graph, so F should now evaluate to ∞. But ℓ is still finite on this modified graph, a contradiction. Hence every linear form using only variables from ρ must use all variables from ρ. Since the correct value on G_ρ is ρ, at least one such linear form must use all variables from ρ exactly once, producing the monomial $\text{cost}(\rho)$. By the above argument, this linear from is minimal, and hence in $\text{Mon}_{le}(p)$.

To show the other direction, let m be a monomial in $\text{Mon}_{le}(p)$. Consider the setting where variables in m are set to 1, and all other variables are set to ∞; let this be the graph H. On H, F evaluates to a finite value, so H must have an s-to-t path. Let ρ be a shortest such path. By construction, the variables on edges of ρ are all in m. Hence for the monomial $\text{cost}(\rho)$ we have the order $\text{cost}(\rho) \preceq m$. We have already proved that $\text{cost}(\rho) \in \text{Mon}_{le}(p)$. Since m is also in $\text{Mon}_{le}(p)$ ie minimal, it follows that $m = \text{cost}(\rho)$. □

This gives us the following useful property.

Property 6 *Let F be a minimal* $(\min, +)$ *formula computing* SHORTEST-PATH*. The top gate of F must be a* min *gate.*

Proof. By Proposition 5, the polynomial produced by F must have exactly the monomials of SHORTEST-PATH in its lower envelope. One of the monomials is the single variable x_{st}, which cannot be further split by addition. If the top gate of F is a $+$ gate with more than one child (since F is minimal, it has no gates with fan-in 1), then to produce this monomial, all but one of the children must return the value 0, making them redundant. □

We start off with some simple lower bounds in the very special case when the depth is 2 or 3.

Proposition 7 *Any depth-2 formula computing* SHORTEST-PATH *must have size* $2^{\Omega(n \log n)}$.

Proof. Let F be a depth-2 formula computing SHORTEST-PATH. By Property 6, the top gate of F must be a min gate. Let this gate have ℓ children. Then the polynomial produced by F has at most ℓ monomials. Hence, by Proposition 5 and Fact 4, $\ell \geq P(n)$. □

Theorem 8 *Any depth-3 circuit computing* SHORTEST-PATH *must have size* $2^{\Omega(n \log n)}$.

Proof. Let F be a depth-3 formula for SHORTEST-PATH. Let p be the polynomial p produced by F. By Proposition 5, $\mathrm{Mon}_{le}(p)$ equals Mon(SHORTEST-PATH), which by Fact 4 has size $P(n)$. Further, by Property 6, the top gate of F must be a min gate. Let this gate have ℓ children. We prove below that each $+$ gate can produce at most one monomial from $\mathrm{Mon}_{le}(p)$. Hence $\ell \geq P(n)$.

Consider a $+$ gate g with fan-in k. Every monomial produced by g has degree k (i.e. k summands). Hence g cannot produce monomials corresponding to paths of length greater than k. In fact, it cannot even produce monomials corresponding to simple paths of length less than k – a shorter path has fewer than k variables while the monomial has exactly k summands, so at least one variable will have to appear with coefficient greater than 1, whereas the monomials $\mathrm{cost}(\rho)$ for simple paths have 0-1 coefficients.

Suppose g produces monomials corresponding to two distinct paths $\eta \neq \rho$, both of length k. We will consider the two graphs G_ρ and G_η.

Let $g_1, \ldots, g_k$ be the children of g and let S_i be the set of children of g_i, $1 \leq i \leq k$. Since the circuit has depth 3, each element of S_i is a variable. Let $\rho = \langle i_0 = s, i_1, \ldots, i_k = t \rangle$, and without loss of generality, let the variable $x_{i_{p-1}} x_{i_p} \in S_p$ for $1 \leq p \leq k$.

First, we show that for each variable x_{ab} in S_p, it must be the case that $a \in \{i_0, \ldots, i_{p-1}\}$ and $b \in \{i_p, \ldots, i_k\}$. To see this, consider the graph $G' = G_\rho \setminus \{(i_{p-1}, i_p)\} \cup (a, b)$. Each g_i still evaluates to 1, and hence g evaluates to k on G'. Therefore G' must be connected, which implies that $a \in \{i_0, \ldots, i_{p-1}\}$ and $b \in \{i_p, \ldots, i_k\}$.

Since the path η is constructed using the variables in the sets S_i, this implies that η cannot have a vertex that is not present in ρ. However, η has the same length as ρ, by assumption, so it uses all the vertices of ρ. Then it must use them in a different order.

Let p be the smallest index where η and ρ differ. Thus the sub-path $\langle i_0, \ldots, i_{p-1} \rangle$ of ρ is also a sub-path of η, and the edge $(i_{p-1}, i_p) \in \rho$, and $(i_{p-1}, i_p) \notin \eta$. Let the edge in η from S_p be (i_q, i_r). By the argument above, $q \in \{0, \ldots, p-1\}$ and $r \in \{p, \ldots, k\}$, and furthermore, $r \neq q+1$. There are two cases to consider. One possibility is that $q < p-1$. Then η has two edges out of i_q, contradicting the assumption that η is a simple path. The other possibility is that $q = p-1$ but $r > p$. Then, to eventually visit vertex i_p, η must use an edge that is a "back-edge" with respect to ρ. But we have shown above that the variable sets S_i prohibit such back-edges.

Therefore such a path η does not exist. This completes the proof. $\square$

We now consider formulas of depth $2d$. By Property 6, the top gate is a min gate, and hence the gates at the lowest level are + gates. Without loss of generality, we assume that all paths from the root to the leaves are of length exactly $2d$. (If necessary, add dummy gates with fan-in 1; this does not change the formula size i.e. number of leaves).

Let $\mathcal{G}$ denote the set of all + gates in C except those at the leaf level. That is, a + gate is in $\mathcal{G}$ if and only if it has as a child another gate of the formula. Let $\mathcal{G}_k$ denote the set of gates in $\mathcal{G}$ with fan-in bounded by k.

Lemma 9 *Let F be an alternating formula, with a* min *gate on top, and of depth $2d$ for some $d \geq 1$. Let the polynomial syntactically produced by F have M monomials.*

If for some $k \in \mathbb{N}$, all + gates except those at the leaves have fan-in at most k (that is, $\mathcal{G} \subseteq \mathcal{G}_k$), then $M \leq (L(F))^{k^{d-1}}$.

Proof. The proof is by induction on d.
Base Case: $d = 1$**.** In this case, to syntactically produce M monomials, the top gate of F must have fan-in M, and so $L(F) \geq M$.
Inductive Step: $d > 1$**.**

Let g_i be the + gates just below the output gate, and let $h_{i,j}$ be the min gates feeding into g_i. (Note that $j \leq k$, by assumption.) Let s_i and $s_{i,j}$ denote the leaf-sizes of the formulas rooted at g_i and $h_{i,j}$ respectively.

$$\begin{aligned}
M &= \text{number of monomials produced by } F \\
&\leq \sum_i (\text{number of monomials produced by } g_i) \\
&\leq \sum_i \prod_j (\text{number of monomials produced by } h_{i,j})
\end{aligned}$$

$$\leq \sum_i \prod_j (s_{i,j})^{k^{d-2}} \qquad \text{(by induction)}$$

$$\leq \sum_i \prod_j (s_i)^{k^{d-2}} \leq \sum_i (s_i)^{k^{d-1}} \leq \left(\sum_i s_i\right)^{k^{d-1}} = (L(F))^{k^{d-1}}.$$

□

Combining Proposition 5 and Fact 4 with Lemma 9, we obtain the following lower bound.

Theorem 10 *If F is a depth $2d$ formula for* SHORTEST-PATH*, where all $+$ gates except those in the bottom level have fan-in at most k, then*

$$L(F) \geq \exp\left(\Omega\left(\frac{n \log n}{k^{d-1}}\right)\right).$$

Proof. By Proposition 5, the polynomial $p(F)$ produced by F must have the monomials of SHORTEST-PATH as its lower envelope. By Fact 4, SHORTEST-PATH has $P(n)$ monomials. Lemma 9 bounds the number of monomials of $p(F)$ from above. Hence

$$(L(F))^{k^{d-1}} \geq \text{number of monomials of } p(F) \geq P(n) = 2^{\Omega(n \log n)},$$

giving the claimed bound on formula size. □

Corollary 11 *Let F be a depth 4 formula for* SHORTEST-PATH*, where all $+$ gates below the top* min *gate have fan-in at most k. Then*

$$L(F) \geq \exp\left(\Omega\left(\frac{n \log n}{k}\right)\right).$$

Remark 12

1. *If $k = O(1)$, then $L(F) = 2^{\Omega(n \log n)}$. This size is achievable even with a depth-2 formula and so this bound is tight*
2. *If $k = O(\sqrt{n})$, then $L(F) = 2^{\Omega(\sqrt{n} \log n)}$. This size is achievable with the depth-4 formula constructed by dynamic programming with all $+$ gates having fan-in $O(\sqrt{n})$ and so this bound is tight.*

A special case of Corollary 11 is when $k = 2$. That is, each path (monomial of SHORTEST-PATH) of length more than 1 is broken at the second level into just two parts. In this case, Corollary 11 tells us that $2^{\Omega(n \log n)}$ size is necessary; by the remark following it, this is also sufficient. What if we relax the condition slightly, and allow a few second-level $+$ gates to have fan-in more than 2? Can we get non-trivial savings in size? We explore this question next.

For natural numbers n, r, k, let $L(n, k, r)$ denote the (leaf-)size of the smallest depth-4 formula that solves SHORTEST-PATH on n-vertex graphs, and where at most r of the $+$ gates at the second level have fan-in exceeding k.

Theorem 13

$$L(n, 2, r) \geq \exp\left(\Omega\left(\frac{n}{2^r}\log\frac{n}{2^r}\right)\right).$$

To prove this theorem, we gradually decrease r while reducing the size of the graphs handled, in Lemma 14 below. This works for any value of k. Finally when $k = 2$ and r has been brought down to 0, we use the bound given by Corollary 11, namely

$$L(n, 2, 0) = \exp(\Omega(n \log n)).$$

Lemma 14 *For $r \geq 1$, $L(n, k, r) \geq L(n(\frac{k-1}{k}), k, r-1)$. In particular,*

$$L(n, 2, r) \geq L\left(\frac{n}{2}, 2, r-1\right).$$

Proof. Let F be the smallest depth-4 formula solving SHORTEST-PATH on n-vertex graphs, where the number of $+$ gates at the second level with fan-in exceeding k is at most r. Let g be the $+$ gate at level 2 that has the largest fan-in. Without loss of generality, assume fan-in of g to be $q \geq k+1$, otherwise the lemma statement trivially holds. Let $g_1, \ldots, g_q$ be the children of g. Let L_i be the set of monomials produced by g_i.

For a monomial ℓ, let $E(\ell) = \{(u, v) \mid x_{uv} \in \ell\}$. By minimality of F, and Proposition 5, g must produce at least one monomial from Mon(SHORTEST-PATH), say corresponding to an st path ρ. Then there exist $\ell_i \in L_i$ such that $\cup_{i=1}^{q} E(\ell_i)$ gives exactly the edges of G_ρ.

Define $S_i = \{u \mid \exists v, (u, v) \in E(\ell_i)\}$. Without loss of generality, the vertex s is in the set S_1.

Claim 15 *For every path η such that g produces the monomial cost(η), and for every $i \in [q]$, the path η visits at least one vertex in S_i.*

Proof. For $i = 1$, this is trivially true because the path η starts at vertex s which is in S_1.

For some $i > 1$, suppose η avoids the set S_i. Suppose g constructs cost(η) by using, for each $j \in [q]$, the monomial $\ell'_j \in L_j$; hence $\cup_{i=j}^{q} E(\ell'_j) = G_\eta$. Consider the graph $H = G_\eta \setminus E(\ell'_i) \cup E(\ell_i)$. Clearly, $g(H) < \infty$. However, since $E(\ell_i)$ is vertex-disjoint from η, H cannot have an st path, contradicting the correctness of F. $\square$

The above claim implies that for each $i > 1$, g does not produce any monomial corresponding to a path avoiding S_i (i.e. a path on $[n] \setminus S_i$). Thus, if we remove g from F to get F', then F' still correctly computes SHORTEST-PATH on the vertex set $[n] \setminus S_i$. We now show that there is $i > 1$ such that $|S_i| \leq (n-2)/(q-1)$.

The set $S = \cup_{i=1}^{q} S_i$ contains all the vertices of ρ except t, so $|S| \leq n-1$ and $|S \setminus \{s\}| \leq n-2$. Further, the sets S_i are disjoint, thereby partitioning S into q parts. Among the $q-1$ parts that do not contain s, by an averaging argument, the smallest part contains no more than $(n-2)/(q-1)$ vertices. Thus F' correctly computes SHORTEST-PATH on $m \geq n - (n-2)/(q-1)$ vertices. Since $q > k$, $m \geq n(k-1)/k$. $\square$

Proof. (of Theorem 13) Using Lemma 14 r times and then Corollary 11, we get

$$L(n, 2, r) \geq L\left(\frac{n}{2}, 2, r-1\right) \geq \ldots \geq L\left(\frac{n}{2^r}, 2, 0\right) = \exp\left(\Omega\left(\frac{n}{2^r} \log \frac{n}{2^r}\right)\right).$$

$\square$

Theorem 13 gives a non-trivial size lower bound for depth-4 formulas when at most say, $O(\log \log n)$ of the second level $+$ gates have fan-in more than 2.

5 Conclusion

Understanding the limits of dynamic programming is an interesting and challenging exercise. In particular, it is surprising that we do not yet know whether with bounded fan-in $(\min, +)$ circuits, $\Omega(n^3)$ is necessary to compute SHORTEST-PATH. In this paper we have focussed on unbounded fan-in $(\min, +)$ formulas. We still do not have a complete understanding of how to optimally exploit additional depth, but we have obtained some partial results. A complete characterisation of the exact complexity of SHORTEST-PATH in this setting, parameterised by depth, as is known for Boolean circuits computing the parity function [4] remains open.

References

[1] Bellman, R.: On a routing problem. Quarterly of Applied Mathematics **16**, 87–90 (1956)

[2] Chen, X., Oliveira, I.C., Servedio, R.A., Tan, L.Y.: Near-optimal small-depth lower bounds for small distance connectivity. In: Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing, STOC '16, pp. 612–625 (2016)

[3] Ford Jr, L.R.: Network flow theory. Tech. Rep. P-923, Rand Corporation (1956)

[4] Håstad, J.: Computational Limitations of Small-depth Circuits. MIT Press, Cambridge, MA, USA (1987)

[5] Jukna, S.: Limitations of incremental dynamic programming. Algorithmica **69**(2), 461–492 (2014)

[6] Jukna, S.: Lower bounds for tropical circuits and dynamic programs. Theory of Computing Systems **57**(1), 160–194 (2015)

[7] Jukna, S.: Tropical complexity, Sidon sets, and dynamic programming. SIAM Journal on Discrete Mathematics **30**(4), 2064–2085 (2016)

[8] Jukna, S.: Incremental versus non-incremental dynamic programming. Electronic Colloquium on Computational Complexity (ECCC) **25**, 42 (2018)

[9] Jukna, S., Schnitger, G.: On the optimality of bellman-ford-moore shortest path algorithm. Theoretical Computer Science **628**, 101–109 (2016)

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

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il