Electronic Colloquium on Computational Complexity, Report No. 49 (2018)

Greedy can also beat pure dynamic programming*

Stasys Jukna† Hannes Seiwert‡

Institut für Informatik, Goethe Universität
Frankfurt am Main, Germany

Abstract

Many dynamic programming algorithms are "pure" in that they only use min or max and addition operations in their recursion equations. The well known greedy algorithm of Kruskal solves the minimum weight spanning tree problem on n-vertex graphs using only $O(n^2 \log n)$ operations. We prove that any pure DP algorithm for this problem must perform $2^{\Omega(n)}$ operations. Since the greedy algorithm can also badly fail on some optimization problems, easily solvable by pure DP algorithms, our result shows that the computational powers of these two types of algorithms are incomparable.

Keywords: Spanning tree; arborescence; arithmetic circuit; tropical circuit; lower bound

1 Introduction and result

A dynamic programming (DP) algorithm is *pure* if it only uses min or max and addition operations in its recursion equations, and the equations do not depend on the actual values of the input weights. Notable examples of such DP algorithms include the Bellman-Ford-Moore algorithm for the shortest s-t path problem [8, 15, 1], the Floyd-Warshall algorithm for the all-pairs shortest paths problem [6, 18], and the Held-Karp algorithm for the Travelling Salesman Problem [9].

It is well known and easy to show that, for some optimization problems, already pure DP algorithms can be much better than greedy algorithms. Namely, there are a lot of optimization problems which are easily solvable by pure DP algorithms (exactly), but the greedy algorithm cannot even achieve any finite approximation factor: maximum weight independent set in a path, or in a tree, the maximum weight simple s-t path in a transitive tournament problem, etc.

In this note, we show that the converse direction also holds: on some optimization problems, greedy algorithms can also be much better than pure dynamic programming. So, the computational powers of greedy and pure DP algorithms are *incomparable*. We will show that the gap occurs on the *undirected* minimum weight spanning tree problem, by first deriving an exponential lower bound on the monotone arithmetic circuit complexity of the corresponding polynomial.

Let K_n be the undirected graph on $\{1, \dots, n\}$. Assume that edges e have their associated weights x_e, considered as formal variables. Let $\mathcal{T}_n$ be the family of all $|\mathcal{T}_n| = n^{n-2}$ spanning trees in K_n,

*Research supported by the DFG grant JU 3105/1-1 (German Research Foundation).

†Affiliated with the Institute of Data Science and Digital Technologies, Faculty of Mathematics and Informatics of Vilnius University, Lithuania. E-mail: stjukna@gmail.com

‡E-mail: seiwert@thi.cs.uni-frankfurt.de

ISSN 1433-8092

each viewed as its *set* of edges. It is well known that $\mathcal{T}_n$ is the family of bases of a matroid, known as *graphic matroid*; so, on this family of feasible solutions, both optimization problems (minimization and maximization) can be solved by standard greedy algorithms. On the other hand, the theorem below states that the polynomial corresponding to $\mathcal{T}_n$ has exponential monotone arithmetic circuit complexity.

The *spanning tree polynomial* (known also as the *Kirchhoff polynomial* of K_n) is the following homogeneous, multilinear polynomial of degree $n-1$:

$$f_n(x) = \sum_{T\in\mathcal{T}_n} \prod_{e\in T} x_e \,.$$

For a multivariate polynomial f with positive coefficients, let $L(f)$ denote the minimum size of a monotone arithmetic $(+,\times)$ circuit computing f. The goal of this note is to prove that $L(f_n)$ is exponential in n.

Theorem.

$$L(f_n) \geq \frac{1}{2n}\left(\frac{9}{5}\right)^n \,.$$

A "directed version" of f_n is the *arborescences polynomial* $\vec{f_n}$. An *arborescence* (known also as a *branching* or a *directed spanning tree*) on the vertex-set $[n]=\{1,\ldots,n\}$ is a directed tree with edges oriented away from vertex 1 such that every other vertex is reachable from vertex 1. Let $\vec{\mathcal{T}_n}$ be the family of all arborescences on $[n]$. Jerrum and Snir [11] have shown that $L(\vec{f_n}) \geq n^{-1}\left(\frac{4}{3}\right)^{n-1}$ holds for the arborescences polynomial

$$\vec{f_n}(x) = \sum_{T\in\vec{\mathcal{T}_n}} \prod_{\vec{e}\in T} x_{\vec{e}} \,.$$

Note that here variables $x_{i,j}$ and $x_{j,i}$ are treated as *distinct*, and cannot both appear in the same monomial. This dependence on orientation was crucially utilized in the argument of [11, p. 892] to reduce a trivial upper bound $(n-1)^{n-1}$ on the number of monomials in a polynomial computed at a particular gate till a non-trivial upper bound $(3n/4)^{n-1}$. So, this argument does not apply to the *undirected* version f_n (where $x_{i,j}$ and $x_{j,i}$ stand for the *same* variable). This is why we use a different argument to handle the undirected case; our argument works also in the directed case.

2 Some consequences

Every pure DP algorithm is just a special (recursively constructed) *tropical* $(\min,+)$ or $(\max,+)$ circuit, that is, a circuit using only min (or max) and addition operations as gates; each input gate of such a circuit holds either one of the variables x_i or a nonnegative real number. So, lower bounds on the size of tropical circuits yield the same lower bounds on the number of operations used by pure DP algorithms. For optimization problems, whose feasible solutions all have the same cardinality, this latter task can be solved by proving lower bounds of the size of monotone arithmetic circuits.

Recall that a multivariate polynomial is *monic* if all its nonzero coefficients are equal to 1, *multilinear* if no variable occurs with degree larger than 1, and *homogeneous* if all monomials have the same degree. Every monic and multilinear polynomial $f(x)=\sum_{S\in\mathcal{F}}\prod_{i\in S} x_i$ defines two optimization problems: compute the minimum or the maximum of $\sum_{i\in S} x_i$ over all $S\in\mathcal{F}$.

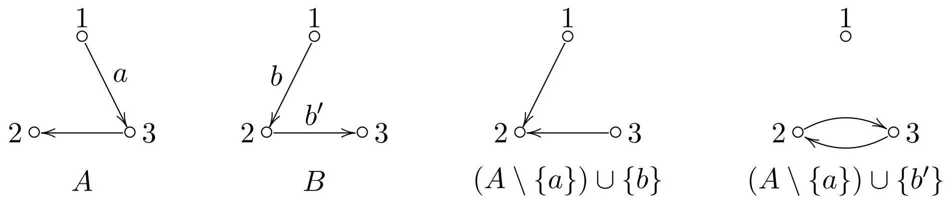

Figure 1: Two arborescences A and B on $[3] = \{1, 2, 3\}$. The only edges in $B \setminus A$ are b and b'. But we cannot add any of them to $A \setminus \{a\}$ to get an arborescence. So, the basis exchange axiom fails.

Reduction Lemma ([11, 12])**.** *If a polynomial f is monic, multilinear and homogeneous, then every tropical circuit solving the corresponding optimization problem defined by f must have at least $L(f)$ gates.*

This fact was proved by Jerrum and Snir [11, Corollary 2.10]; see also [12, Theorem 9] for a simpler proof. The proof idea is fairly simple: having a tropical circuit, turn it into a monotone arithmetic $(+, \times)$ circuit, and use the homogeneity of f to show that, after removing some of the edges entering $+$-gates, the resulting circuit will compute our polynomial f.

Greedy can beat pure DP The (weighted) *minimum spanning tree* problem $MST_n(x)$ is, given an assignment of nonnegative real weights to the edges of K_n, compute the minimum weight of a spanning tree of K_n, where the weight of a graph is the sum of weights of its edges. So, this is exactly the minimization problem defined by the spanning tree polynomial f_n:

$$MST_n(x) = \min_{T \in \mathcal{T}_n} \sum_{e \in T} x_e \,.$$

Since the family $\mathcal{T}_n$ of feasible solutions of this problem is the family of bases of the (graphic) matroid, the problem *can* be solved by the standard greedy algorithm. In particular, the well known greedy algorithm of Kruskal [14] solves MST_n using only $O(n^2 \log n)$ operations. On the other hand, since the spanning tree polynomial f_n is monic, multilinear and homogeneous, our theorem together with the Reduction lemma implies that any $(\min, +)$ circuit solving the problem MST_n must use at least $L(f_n) = 2^{\Omega(n)}$ gates and, hence, at least so many operations must be performed by any pure DP algorithm solving MST_n. This gap between pure DP and greedy algorithms is our main result.

Remark (Directed versus undirected spanning trees). The arborescences polynomial $\vec{f_n}$ is also monic, multilinear and homogeneous, so that the Reduction lemma, together with the above mentioned lower bound on $L(\vec{f_n})$ of Jerrum and Snir [11], also yields the same lower bound on the size of $(\min, +)$ circuits solving the minimization problem on the family $\vec{\mathcal{T}_n}$ of arborescences. But this does not separate DP from greedy, because the downward closure of $\vec{\mathcal{T}_n}$ is *not* a matroid (see Fig. 1 for a simple counter-example). As observed by Edmonds [4], this family is an intersection of two matroids, meaning that unlike for MST_n, the greedy algorithm can only approximate the minimization problem on $\vec{\mathcal{T}_n}$ within the factor 2. Polynomial time algorithms solving this problem exactly were found by several authors, starting from Edmonds [5]. The fastest algorithm for the problem is due to Tarjan [16], and solves the problem in time $O(n^2 \log n)$, that is, with the same time complexity as Kruskal's greedy algorithm for undirected graphs [14]. But these are no more greedy algorithms. So, $\vec{\mathcal{T}_n}$ does not separate greedy and pure DP algorithms.

Subtraction can speed up $(+, \times)$ circuits There is a well-known determinantal formula, due to Kirchhoff [13] (see, e.g., [2, Theorem II.12]), known as the (weighted) *matrix tree theorem*. This formula provides a way to compute the spanning tree polynomial f_n by an arithmetic $(+, \times, -)$ circuit of polynomial size. Together with our theorem, this gives yet another explicit example where the use of subtraction in arithmetic circuits leads to exponential saving in their size; that already one subtraction gate can lead to such saving was shown already by Valiant [17] using another polynomial corresponding to perfect matchings in planar graphs.

Division can speed up $(+, \times)$ circuits Fomin, Grigoriev and Koshevoy [7] have recently proved that the spanning tree polynomial f_n *can* be computed by an arithmetic $(+, \times, \div)$ circuit using only $O(n^3)$ gates. Their recursion (given by Lemma 8.3 in [7]) turns to the following procedure for computing f_n: in order to compute $f_n(x)$ on the vertex-set $\{1, \ldots, n\}$ with edge-weights $x_{i,j}$, first compute the sum $X_n := x_{1,n} + \cdots + x_{n-1,n}$ of the weights of edges incident to the last vertex n, then compute $f_{n-1}(x')$ on the first $n-1$ vertices under the new edge-weights

$$x'_{i,j} := x_{i,j} + \frac{x_{i,n} \cdot x_{j,n}}{X_n} .$$

Lemma 8.3 in [7] shows that then $f_n(x) = X_n \cdot f_{n-1}(x')$ holds. Together with our theorem, this shows that, like subtraction gates, *division* gates in monotone arithmetic circuits can also lead to exponential savings. Actually, it is shown in [7] that also the arborescences polynomial $\vec{f}_n$ can be computed by an arithmetic $(+, \times, \div)$ circuit using $O(n^3)$ gates. So, the same gap between $(+, \times, \div)$ and $(+, \times)$ circuits follows also from the aforementioned lower bound of Jerrum and Snir [11] for $\vec{f}_n$.

Subtraction can speed up pure DP algorithms Since "division" in tropical $(\min, +)$ and $(\max, +)$ semirings corresponds to *subtraction*, the result of [7] also implies that the minimum weight spanning tree problem can be solved using only $O(n^3)$ $\min, +, -$ gates. So, subtraction operation can exponentially speed-up pure DP algorithms: in order to compute $MST_n(x)$ on the vertex-set $\{1, \ldots, n\}$ with edge-weights $x_{i,j}$, first compute the minimum $X_n := \min\{x_{1,n}, \ldots, x_{n-1,n}\}$ of weights of edges incident to the last vertex n, then compute $MST_{n-1}(x')$ on the first $n-1$ vertices under the new edge-weights

$$x'_{i,j} := \min\{x_{i,j},\ x_{i,n} + x_{j,n} - X_n\} ,$$

and output $MST_n(x) = X_n + MST_{n-1}(x')$.

3 Proof of the theorem

We will use the following well-known decomposition result, first proved by Hyafil [10, Theorem 1] and Valiant [17, Lemma 3]. A polynomial is *nonnegative* if it has no negative coefficients.

Decomposition Lemma (Hyafil, Valiant)**.** *Let f be a nonnegative homogeneous polynomial of degree m. If $L(f) \leq t$, then f can be written as a sum $f = g_1 \cdot h_1 + \cdots + g_t \cdot h_t$ of products of nonnegative homogeneous polynomials, each of degree at most $2m/3$.*

Proof. (Due to Valiant [17].) Induction on the circuit size t. Since the polynomial f is homogeneous, the polynomial g_v computed at each gate v must be also homogeneous. Starting from the output gate, we walk backwards, by always choosing that of two input gates whose polynomial has larger degree. Proceeding in this way, we will arrive at a gate v with $m/3 \leq \deg(g_v) \leq 2m/3$. Let f_v be the polynomial computed by the circuit, after the inputs of gate v are replaced by zeroes. Then $f = g_v \cdot h + f_v$ for some polynomial h of degree $\deg(h) = m - \deg(g_v) \leq 2m/3$. Since the (also homogeneous) polynomial f_v is computable by a circuit of size $t-1$ (the need of the gate v is already eliminated), we can apply the induction hypothesis to f_v. $\square$

The "combinatorial core" of our argument is the following property of forests, viewed as sets of their edges. Let $\mathcal{A}, \mathcal{B}$ be two nonempty families of forests in K_n such that $\mathcal{A}$ contains a forest with a edges, and $\mathcal{B}$ contains a forest with b edges.

Forest Lemma. *If the intersection of any two forests $A \in \mathcal{A}$ and $B \in \mathcal{B}$ is empty, and their union is a spanning tree of K_n, then the connected components of all forests in $\mathcal{A}$ have the same sets of vertices, and the same holds for forests in $\mathcal{B}$. Moreover, then $|\mathcal{A}| \cdot |\mathcal{B}| \leq a^a \cdot b^b$ and $a + b = n - 1$ hold.*

Proof. To prove the first claim, it is enough to show that every pair of vertices is either connected in *all* forests in $\mathcal{A}$, or is disconnected in *all* forests in $\mathcal{A}$.

To show this, suppose contrariwise that some two vertices x and y are disconnected in one forest $A \in \mathcal{A}$, but are connected in some other forest $A' \in \mathcal{A}$. Take an arbitrary forest $B \in \mathcal{B}$. Since $A \cup B$ must be a spanning tree of K_n, and vertices x and y are disconnected in the forest A, there must be a path between x and y in the forest B. But then the graph $A' \cup B$ contains a cycle throughout vertices x and y, a contradiction with this graph being a tree. This shows the claim for the forests in $\mathcal{A}$. The argument for the forests in $\mathcal{B}$ is the same.

So, let $U_1, \ldots, U_p \subset [n]$ be the sets of vertices of the connected components of forests in $\mathcal{A}$, and $V_1, \ldots, V_q \subset [n]$ be the sets of vertices of the connected components of forests in $\mathcal{B}$. Let also $u_i = |U_i|$ and $v_i = |V_i|$ be the numbers of vertices in these components. Then every forest in $\mathcal{A}$ has $a = \sum_{i=1}^{p}(u_i - 1)$ edges, and every forest in $\mathcal{B}$ has $b = \sum_{i=1}^{q}(v_i - 1)$ edges. Since the forests are disjoint and their unions must be spanning trees (with $n-1$ edges), we also have that $a + b = n - 1$.

Every forest in $\mathcal{A}$ consists of spanning trees of complete graphs on the sets $U_1, \ldots, U_p$, and similarly for forests in $\mathcal{B}$. By a classical result of Cayley [3], the number of trees on n labeled vertices is n^{n-2}. This implies that there are only $u_i^{u_i-2}$ spanning trees of the complete graph on each U_i, and only $v_j^{v_j-2}$ spanning trees of the complete graph on each V_j. So, using the inequalities $r^{r-2} \leq (r-1)^{r-1}$ and $r^r \cdot s^s \leq (r+s)^{r+s}$ for integers $r, s \geq 2$, we obtain:

$$
\begin{aligned}
|\mathcal{A}| \cdot |\mathcal{B}| &\leq \prod_i u_i^{u_i-2} \cdot \prod_j v_j^{v_j-2} \leq \prod_i (u_i - 1)^{u_i-1} \cdot \prod_j (v_j - 1)^{v_j-1} \\
&\leq \Big[\sum_i (u_i - 1)\Big]^{\sum_i (u_i-1)} \cdot \Big[\sum_j (v_j - 1)\Big]^{\sum_j (v_j-1)} = a^a \cdot b^b \,. \quad \square
\end{aligned}
$$

Proof of the theorem. Let $t = L(f_n)$. The spanning tree polynomial f_n is multilinear and homogeneous of degree $m = n-1$: every spanning tree T of K_n has $|T| = n-1$ edges. By the Decomposition lemma, we can express this polynomial as a sum of at most t products $g \cdot h$ of nonnegative homogeneous polynomials, where the degree of g and of h is at most $2m/3$. The polynomial $g \cdot h$ must be a homogeneous polynomial of degree m. So, if we denote the degrees of g and h by a and b, then $a + b = m$ and $m/3 \leq a, b \leq 2m/3$ must hold.

Every monomial of f_n is a multilinear monomial of the form $\prod_{e\in T} x_e$ for some spanning tree T. Since the polynomials g and h in the decomposition of f_n are nonnegative, there can be no cancellations. This implies that all the monomials of $g \cdot h$ must be also monomials of f, that is, must correspond to spanning trees. So, the monomials of g and h correspond to forests. Let $\mathcal{A}$ be the family of forests corresponding to monomials of the polynomial g, and $\mathcal{B}$ the family of forests corresponding to monomials of the polynomial h.

Since all monomials of $g \cdot h$ must also be monomials of f_n, the pair $\mathcal{A}, \mathcal{B}$ fulfills the conditions of the Forest lemma for some a between $m/3$ and $2m/3$ (the disjointness property follows from the multilinearity of $g \cdot h$). So, the number of monomials in the polynomial $g \cdot h$ cannot exceed

$$|\mathcal{A}| \cdot |\mathcal{B}| \leq a^a (m-a)^{m-a} \leq (m/3)^{m/3} \cdot (2m/3)^{2m/3} = m^m \left(2^{2/3}/3\right)^m < m^m \left(\frac{5}{9}\right)^m ,$$

where the second inequality holds because the function $x^x(1-x)^{1-x}$ is convex in the interval $(0, 1)$. Since the polynomial f_n has n^{n-2} monomials, the desired lower bound on $t = L(f_n)$ follows:

$$t \geq \frac{n^{n-2}}{|\mathcal{A}| \cdot |\mathcal{B}|} \geq \frac{m^{m-1}}{|\mathcal{A}| \cdot |\mathcal{B}|} \geq \frac{1}{m}\left(\frac{9}{5}\right)^m = \frac{1}{n-1}\left(\frac{9}{5}\right)^{n-1} \geq \frac{1}{2n}\left(\frac{9}{5}\right)^n . \quad \square$$

Finally, note that the same argument works also for the arborescences polynomial $\vec{f_n}$: in this case, the (directed) forests in $\mathcal{A}$ and $\mathcal{B}$ must fulfill an *additional* restriction: also the *directions* of "docking" edges must be consistent; so, the number $|\mathcal{A}| \cdot |\mathcal{B}|$ of monomials in each polynomial $g \cdot h$ can only be smaller.

References

[1] R. Bellman. On a routing problem. *Quarterly of Appl. Math.*, 16:87–90, 1958.

[2] B. Bollobás. *Modern graph theory, Springer.* Springer, 1998.

[3] A. Cayley. A theorem on trees. *Quart. J. Pure Appl. Math.*, 23:376–378, 1889.

[4] J. Edmonds. Optimum branchings. *J. of Res. of the Nat. Bureau of Standards*, 71B(4):233–240, 1967.

[5] J. Edmonds. Edge-disjoint branchings. In B. Rustin, editor, *Combinatorial Algorithms*, pages 91–96. Academic Press, 1973.

[6] R.W. Floyd. Algorithm 97, shortest path. *Comm. ACM*, 5:345, 1962.

[7] S. Fomin, D. Grigoriev, and G. Koshevoy. Subtraction-free complexity, cluster transformations, and spanning trees. *Found. Comput. Math.*, 15:1–31, 2016.

[8] L.R. Ford. Network flow theory. Technical Report P-923, The Rand Corp., 1956.

[9] M. Held and R.M. Karp. A dynamic programming approach to sequencing problems. *SIAM J. on Appl. Math.*, 10:196–210, 1962.

[10] L. Hyafil. On the parallel evaluation of multivariate polynomials. *SIAM J. Comput.*, 8(2):120–123, 1979.

[11] M. Jerrum and M. Snir. Some exact complexity results for straight-line computations over semirings. *J. ACM*, 29(3):874–897, 1982.

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

[13] G. Kirchhoff. Über die Auflösung der Gleichungen, auf welche man bei der Untersuchungen der linearen Verteilung galvanischer Ströme geführt wird. *Ann. Phys. Chem.*, 72:497–508, 1847.

[14] J.B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. of AMS*, 7:48–50, 1956.

[15] E.F. Moore. The shortest path through a maze. In *Proc. Internat. Sympos. Switching Theory*, volume II, pages 285–292, 1957.

[16] R.E. Tarjan. Finding optimum branchings. *Networks*, 7(1):25–35, 1977.

[17] L.G. Valiant. Negation can be exponentially powerful. *Theor. Comput. Sci.*, 12:303–314, 1980.

[18] S. Warshall. A theorem on boolean matrices. *J. ACM*, 9:11–12, 1962.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il