Electronic Colloquium on Computational Complexity, Report No. 142 (2026)

Shortest Paths with Linear Edge Weights

Suryajith Chillara
CSTAR, IIIT Hyderabad, India
suryajith.chillara@iiit.ac.in

Kshitij Gajjar
CSTAR, IIIT Hyderabad, India
kshitij@iiit.ac.in

Nithish Raja*
TU Eindhoven, Netherlands
n.r.raja@tue.nl

July 31, 2026

Abstract

We study shortest paths in directed graphs whose edge weights are of the form

$$\mathsf{wt}(e) = a_{e,1}\lambda_1 + a_{e,2}\lambda_2 + a_{e,3}\lambda_3 + \cdots + a_{e,d}\lambda_d + a_{e,d+1}.$$

Here, each $a_{e,i} \in \mathbb{R}$ is a fixed constant for each edge e, whereas each λ_i is a shared variable across the entire graph. So, there could be different shortest paths in the graph for different values of the λ_i's. The number of such shortest paths is of interest in several combinatorial optimization problems. This is called the *Parametric Shortest Paths* problem, and has been studied since the 1980s.

For $d = 1$, Carstensen (1983) showed that the number of shortest paths in n-vertex graphs is at most $n^{O(\log n)}$. She also proved a matching lower bound of $n^{\Omega(\log n)}$, which was later refined by Mulmuley & Shah (2001). For $d = 2$, Gajjar & Radhakrishnan (2019) showed an upper bound of $n^{O(\log^2 n)}$. Barth, Funke & Proissl (2022) generalized their result to prove an upper bound of $n^{O_d(\log^d n)}$ for all positive integers d. The lower bound did not undergo any improvement over the years.

In this paper, we close this long line of research by showing an $n^{O(d \log n)}$ upper bound for all positive integers d, exponentially improving the previous upper bound. We observe that a matching lower bound of $n^{\Omega(d \log n)}$ can be obtained by trivially extending existing lower bound constructions for $d = 1$. We also show that our proof can be adapted to work for undirected graphs with positive edge weights. Furthermore, for directed graphs whose edge weights are univariate polynomials of degree at most q, we prove an upper bound of $n^{O(\log n + \log q)}$.

Finally, building upon work on the *Point Location* problem by Ezra, Har-Peled, Kaplan & Sharir (2020), we construct a *Shortest Path Oracle* which takes as input a point $\overline{x} \in \mathbb{R}^d$, and outputs a shortest path at $\overline{\lambda} = \overline{x}$ in sublinear time (for a wide regime of d).

All earlier upper bound proofs proceeded by arranging the vertices of the graph in layers, splitting the graph across its middle layer into two "halves", and then recursing on each half-graph. We deviate from this proof methodology by "halving" the graph in a different way: we eliminate all the odd-numbered layers and retain only the even-numbered layers, whilst maintaining requisite shortest paths of the original graph. We then view shortest paths in the half-graph as convex objects in d-dimensional space, which leads us to the required recurrence.

*Part of this work was done while the author was a Master's student at CSTAR, IIIT Hyderabad, India.

ISSN 1433-8092

Contents

1 Introduction

Computing shortest paths in graphs is a fundamental problem in computer science, dating back to over seven decades. Some of the most popular algorithms to efficiently find a shortest path in a given network, namely Dijkstra's algorithm [Dij59], the Bellman-Ford algorithm [Bel58, FJ56, Moo59], the Floyd-Warshall algorithm [Flo62, War62, R+59], and Johnson's algorithm [Joh77] are all now taught in basic Algorithms courses worldwide.

In fact, the most popular algorithm used in practice is the A^* algorithm [HNR68] to compute shortest paths (which is basically Dijkstra's algorithm with a heuristic), even more popular than the famed Fast-Fourier Transform [CT65, HJB85] that performs multiplications!

Most of the research on shortest paths is focused on graphs whose edge weights are fixed, including all the algorithms mentioned above. In most practical scenarios, however, the edge weights are changing in real time. Some common examples are the traffic on a street in a road network, the network traffic on a LAN wire connecting two nodes in a computer network, the strength of the signal being passed across two cellular masts (cellphone towers) in a telecommunication network. Such situations necessitate the study of graphs in which the edge weights are varying with time. There are mainly four lines of work in this regard:

(i) Edge weights are varying probabilistically (generally with the graph having a probability distribution, and each edge having a mean and a variance [NKBM06, FG85]).

(ii) Weighted edges are coming in an online fashion (as in temporal or dynamic graphs [WCH+14, DYQ08, RZ04, RR96]).

(iii) Edge weights are given by time-varying functions (each edge is assigned a function of time, mapping it to a real number denoting the weight of the edge [Car83, Eri10]).

(iv) Edge weights are given by time-varying functions (but path weights are functional compositions of their constituent edge weight functions [FHS14, GVCR21]).

All four of these aspects have been studied in the past. However, throughout this paper, we will only be focusing on the third aspect (item (iii) above), known as parametric shortest paths.

1.1 Parametric Shortest Paths

Research on parametric shortest paths began in the 1980s, when Carstensen [Car83] studied the minimum size of a shortest path cover (MSPC) of graphs whose edge weights are linear functions of a parameter $\overline{\lambda}$.

Definition 1 (Minimum Shortest Path Cover (MSPC))**.** *Let G be a directed graph with two special vertices s and t, whose edge weights are linear functions of d variables (also called parameters) $\lambda_1, \lambda_2, \ldots, \lambda_d$; that is,*

$$\mathsf{wt}(e) = a_{e,1}\lambda_1 + a_{e,2}\lambda_2 + \cdots + a_{e,d}\lambda_d + a_{e,d+1}$$

for each edge $e \in E(G)$, where $a_{e,i} \in \mathbb{R}$ for each $i \in \{1, 2, \ldots, d+1\}$. (See Figure 1 for an example of such a graph.) The weight of a path P from s to t is defined in the usual way as the sum of the weights of its constituent edges:

$$\mathsf{wt}(P) = \sum_{e \in P} \mathsf{wt}(e) = \sum_{e \in P} \left(a_{e,1}\lambda_1 + a_{e,2}\lambda_2 + \cdots + a_{e,d}\lambda_d + a_{e,d+1}\right).$$

*Let $\mathcal{P}_{s,t}$ be the set of all paths from s to t in G. Then, $\mathcal{S}$ is a Minimum Shortest Path Cover (*MSPC*) of G if $\mathcal{S}$ is a minimum-sized subset of $\mathcal{P}_{s,t}$ such that for every $\overline{x} \in \mathbb{R}^d$, there is a path $P \in \mathcal{S}$ which is a shortest path in G at $\overline{\lambda} = \overline{x}$; that is, the path P is a shortest path from s to t in the fixed-edge weight graph obtained upon setting $(\lambda_1, \lambda_2, \ldots, \lambda_d) = (x_1, x_2, \ldots, x_d)$ in all the edge weights of G. (See Figure 2 for an example of an MSPC.)*

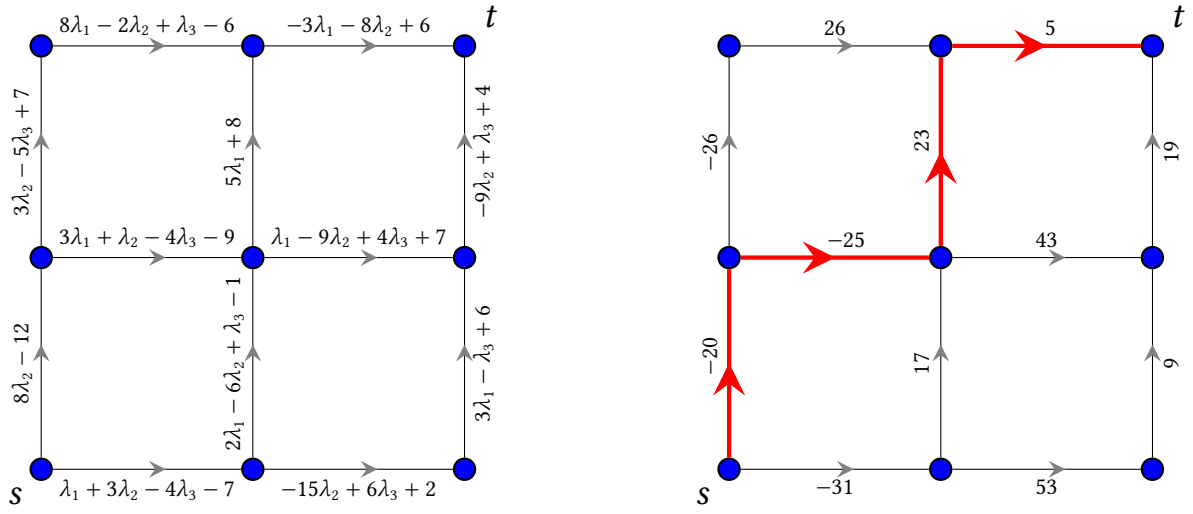


Figure 1: *Left:* A directed graph G with two special vertices s and t, whose edge weights are of the form $a_{e,1}\lambda_1 + a_{e,2}\lambda_2 + a_{e,3}\lambda_3 + a_{e,4}$ (that is, $d = 3$ in Definition 1). Such a graph is called a parametric graph. The MSPC of G is explained in Figure 2. *Right:* An instantiation of G at $\lambda_1 = 3$, $\lambda_2 = -1$, $\lambda_3 = 6$. The shortest s to t path in this instantiation of G is indicated in thick red. Such a path is called a parametric shortest path.

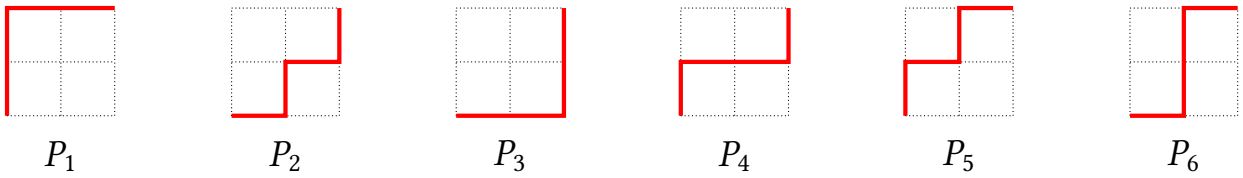


Figure 2: The set $\mathcal{P}_{s,t} = \{P_1, P_2, P_3, P_4, P_5, P_6\}$ of s to t paths in the graph G of Figure 1 (*left*). Note that P_5 is the only shortest path in G at $\overline{\lambda} = (3, -1, 6)$, as shown in Figure 1 (*right*). So, $P_5 \in \text{MSPC}(G)$. Similarly, P_2, P_4, P_6 are the only shortest paths in G at $(0, 0, -100)$, $(0, 0, 0)$, $(-200, 2, 0)$, respectively. So, $P_2, P_4, P_6 \in \text{MSPC}(G)$. Also, $\mathsf{wt}(P_1) = \mathsf{wt}(P_5)+2$ and $\mathsf{wt}(P_3) = \mathsf{wt}(P_2)+2$ for all $\overline{\lambda} \in \mathbb{R}^3$, so $P_1, P_3 \notin \text{MSPC}(G)$. Hence, $|\text{MSPC}(G)| = 4$.

1.2 Prior Work

Carstensen showed[1] that every graph G on n vertices whose edge weights are of the form $\mathsf{wt}(e) = a_{e,1}\lambda_1 + a_{e,2}$ (that is, $d = 1$) admits an MSPC of size $n^{O(\log n)}$. She also exhibited graphs whose MSPC is of size $n^{\Omega(\log n)}$, proving that her upper bound is optimal up to the constant in the exponent.

Building upon Carstensen's work, Mulmuley & Shah [MS00] proved that her $n^{\Omega(\log n)}$ lower bound can be expressed with edge weights having just poly-logarithmic bits (that is, the $a_{e,i}$'s for each edge e can be represented using $O(\log^3(n))$ bits). Gajjar & Radhakrishan [GR19] modified their construction and edge weights slightly (still keeping them poly-logarithmic), gave a better exposition of their proof, and showed that the $n^{\Omega(\log n)}$ lower bound also holds (possibly with a different constant in the exponent) for planar graphs, thereby refuting a conjecture of Nikolova [Nik09].

Gajjar & Radhakrishan [GR19] also explored graphs with edge weights of the form $a_{e,1}\lambda_1 + a_{e,2}\lambda_2 + a_{e,3}$ (that is, $d = 2$), and proved an upper bound of $n^{O(\log^2 n)}$ on the size of the MSPC in such graphs. However, their

[1] Carstensen attributed the proof of this upper bound to Gusfield.

proof did not work for three parameters and beyond. Barth, Funke & Proissl [BFP22] succeeded in extending their idea to d parameters (graphs with edge weights of the form $a_{e,1}\lambda_1 + a_{e,2}\lambda_2 + a_{e,3}\lambda_3 + \cdots + a_{e,d}\lambda_d + a_{e,d+1}$), showing an upper bound of $n^{O_d(\log^d n)}$ for all positive integers d. Chatterjee, Gajjar & Radhakrishnan [CGR23] improved this[2] to $n^{O(\log^d n)}$, which is state-of-the-art.

The lower bound, however, has remained $n^{\Omega(d \log n)}$. Though not explicitly mentioned in any of the preceding papers, this lower bound can be easily realized by considering a graph with edge weights of the form $a_{e,1}\lambda_1 + a_{e,2}$ whose MSPC is of size $n^{\Omega(\log n)}$, and then attaching d copies of this graph with itself in series (see Claim 14 for more details).

Note that the d is in the (single) exponent in the lower bound, and in the double exponent (exponent of the exponent) in the upper bound. This leaves a massive gap between the lower and upper bounds. In this work, we fully bridge this gap.

1.3 Our Contributions

Our main contribution is an upper bound on the MSPC of directed acyclic graphs with linear edge weights.

Theorem 2. *Let G be an n-vertex directed acyclic graph with edge weights of the form $a_{e,1}\lambda_1 + a_{e,2}\lambda_2 + \cdots + a_{e,d}\lambda_d + a_{e,d+1}$. Then,*

$$|\mathrm{MSPC}(G)| \in n^{O(d \log n)}.$$

We also show that our proof for Theorem 2 can be adapted to work for directed graphs without negative-weight cycles, and for undirected graphs without negative-weight edges.

Theorem 3. *Let G be an n-vertex directed graph without negative-weight cycles with edge weights of the form $a_{e,1}\lambda_1 + a_{e,2}\lambda_2 + \cdots + a_{e,d}\lambda_d + a_{e,d+1}$. Then,*

$$|\mathrm{MSPC}(G)| \in n^{O(d \log n)}.$$

Corollary 4. *Let G be an n-vertex undirected graph without negative-weight edges with edge weights of the form $a_{e,1}\lambda_1 + a_{e,2}\lambda_2 + \cdots + a_{e,d}\lambda_d + a_{e,d+1}$. Then,*

$$|\mathrm{MSPC}(G)| \in n^{O(d \log n)}.$$

We also show an upper bound on the MSPC of directed graphs with univariate polynomial edge weights.

Theorem 5. *Let G be an n-vertex directed graph with edge weights of the form $a_{e,q}\lambda^q + a_{e,q-1}\lambda^{q-1} + \cdots + a_{e,2}\lambda^2 + a_{e,1}\lambda + a_{e,0}$. Then,*

$$|\mathrm{MSPC}(G)| \in n^{O(\log n + \log q)}.$$

We also study the problem from the algorithmic standpoint. To this end, we construct a data structure that preprocesses the graph, and efficiently computes a shortest path in the graph for a given $\overline{\lambda}$ in real time.

Theorem 6. *Let G be an n-vertex directed graph with edge weights of the form $a_{e,1}\lambda_1 + a_{e,2}\lambda_2 + \cdots + a_{e,d}\lambda_d + a_{e,d+1}$. Then, there exists a data structure that takes as input an $\overline{x} \in \mathbb{R}^d$, and outputs a shortest path in the graph G at $\overline{\lambda} = \overline{x}$ in $O(d^4 \log^2 n)$ time. The data structure takes $n^{\tilde{O}(d)}$ space and $n^{\tilde{O}(d)}$ preprocessing time.*

Theorem 7. *Let $G = (V, E)$ be an n-vertex directed graph with edge weights of the form $a_{e,q}\lambda^q + a_{e,q-1}\lambda^{q-1} + \cdots + a_{e,2}\lambda^2 + a_{e,1}\lambda + a_{e,0}$. Then, there exists a data structure that takes as input an $x \in \mathbb{R}$, and outputs a shortest path in the graph G at $\lambda = x$ in $O(\log^2 n + \log n \log q)$ time. The data structure takes $(nq)^{\tilde{O}(1)}$ space and $(nq)^{\tilde{O}(1)}$ preprocessing time.*

[2] The constant behind the big-O notation in the exponent of [BFP22] is 2^d (an exponential dependence on d), whereas this constant in [CGR23] is simply the number 4 (and therefore has no dependence on d).

1.4 Proof Overview

In this section, we outline in detail the main ideas behind the proof of our main result (Theorem 2). The proofs of our other results can be easily followed once this is understood.

Let $G = G_{n,n}$ be a layered DAG with n layers, n vertices per layer, source s and sink t. Let the edge weights be linear functions of d parameters $\lambda_1, \ldots, \lambda_d$. To bound $|\mathrm{MSPC}(G)|$, a standard divide-and-conquer approach does the following – split the graph into $2n$ instances of $G_{n/2,n}$ through the vertices of the middle layer $\{v_1, \ldots, v_n\}$ and then recurse on those. In particular, each of these $G_{n/2,n}$ instances are of the form source s and sink v_i (call this G_i) for each i, or of the form source v_i and sink t (call this G'_i) for each i.

If graphs G and G' are connected in series then $|\mathrm{MSPC}(G \circ_{\mathrm{ser}} G')| \leq |\mathrm{MSPC}(G)| \cdot |\mathrm{MSPC}(G')|$, and if they are connected in parallel, $|\mathrm{MSPC}(G \circ_{\mathrm{par}} G')| \leq |\mathrm{MSPC}(G)| + |\mathrm{MSPC}(G')|$.

$$|\mathrm{MSPC}(G)| \leq \sum_{i=1}^{n} |\mathrm{MSPC}(G_i)| \cdot |\mathrm{MSPC}(G'_i)|.$$

Instead of dividing this at the middle layer, one could generalize the aforementioned divide-and-conquer argument through ℓ such layers and then merge carefully. We note that even this carefully divided approach does not give bounds that are better than $n^{O\left(\log^d(n)/\log\log^{d-1}(n)\right)}$ (for $d \geq 2$). Though this is a better bound than that of [BFP22], it is only a modest improvement.

We first observe that when two graphs G_1 and G_2 are connected in series, the size of the shortest path cover may be much less than the product of the sizes of the shortest path covers in these individual instances. That is, a parametric shortest path P in G_1 concatenates with a parametric shortest path P' in G_2 if and only if there is a point in $\mathbb{R}^d$ where these are simultaneously shortest in their respective graphs. We illustrate this through the following example.

Let G be a directed acyclic graph as shown in Figure 3a whose edge weights vary as linear functions of parameters λ_1 and λ_2. Note that edge 0 between s and v would be the shortest edge for all values of $(\lambda_1, \lambda_2) \in \mathbb{R}^2$ such that $\lambda_1 + 2\lambda_2 \leq 2\lambda_1 + \lambda_2$, and edge 1 would be the shortest edge for all values of $(\lambda_1, \lambda_2) \in \mathbb{R}^2$ such that $\lambda_1 + 2\lambda_2 \geq 2\lambda_1 + \lambda_2$ (illustrated in Figure 3b). Let $\pi_{s,v}$ denote this partition of $\mathbb{R}^2$ through the line $\lambda_1 = \lambda_2$. Similary, edge 0 between v and t would be the shortest edge for all values of $(\lambda_1, \lambda_2) \in \mathbb{R}^2$ such that $\lambda_1 + \lambda_2 \leq 2\lambda_1 + \lambda_2 - 3$, and edge 1 would be the shortest edge for all values of $(\lambda_1, \lambda_2) \in \mathbb{R}^2$ such that $\lambda_1 + \lambda_2 \geq 2\lambda_1 + \lambda_2 - 3$ (illustrated in Figure 3c). Let $\pi_{v,t}$ denote this partition of $\mathbb{R}^2$ through the line $\lambda_1 = 3$.

We represent each s to t path using a tuple (a, b) (for $a, b \in \{0, 1\}$) to indicate that the path takes edge a between s and v, and edge b between v and t. With this representation, there are four such paths $(0,0), (0,1), (1,0), (1,1)$ with parametric weights $2\lambda_1 + 3\lambda_2$, $3\lambda_1 + 3\lambda_2 - 3$, $3\lambda_1 + 2\lambda_2$, $4\lambda_1 + 2\lambda_2 - 3$ respectively. In particular, path $(0,0)$ is shortest for all values of (λ_1, λ_2) that simultaneously satisfy the inequalities

$$\begin{aligned} 2\lambda_1 + 3\lambda_2 &\leq 3\lambda_1 + 3\lambda_2 - 3, \\ 2\lambda_1 + 3\lambda_2 &\leq 3\lambda_1 + 2\lambda_2, \\ 2\lambda_1 + 3\lambda_2 &\leq 4\lambda_1 + 2\lambda_2. \end{aligned}$$

which simplify to $\lambda_1 \geq 3$ and $\lambda_1 \geq \lambda_2$. These set of inequalities defines a region in $\mathbb{R}^2$ where path $(0,0)$ is the shortest. Similarly, we can define regions for the other paths. This creates a partition of the space $\pi_{s,t}$. We will now claim that such a partition could simply be obtained through a *superimposition* of the partitions $\pi_{s,v}$ and $\pi_{v,t}$ by capitalizing on the independence of paths between the nodes s and v, and nodes v and t (as

illustrated in Figure 3d). Here, we also use that fact that in a DAG if a shortest s to t path passes through v, its segment between s and v, and its segment between v and t both have to be shortest by themselves.

More formally, the lines $\lambda_1 = \lambda_2$ and $\lambda_1 = 3$ together partition $\mathbb{R}^2$ into four regions and each of these regions corresponds to one of the s to t paths. For a given point, its side with respect $\lambda_1 = \lambda_2$ helps us identify which of the two edges between s and v is the shortest, and its side with respect to $\lambda_1 = 3$ helps us identify which of the two edges between v and t is the shortest.

Let G_1 and G_2 be two disjoint graphs as shown in Figure 4. In graph G_1, nodes $u_1, \ldots, u_n$ are n distinct degree-2 nodes that connect s and t by providing n disjoint paths. For each $i \in [n]$, let $L_i(\lambda_1, \ldots, \lambda_d)$ be the linear weight function associated with the path $s \to u_i \to t$. Similarly, in graph G_2, nodes $v_1, \ldots, v_n$ are n distinct degree-2 nodes that connect s' and t' by providing n disjoint paths. For each $i \in [n]$, let $L'_i(\lambda_1, \ldots, \lambda_d)$ be the linear weight function associated with the path $s' \to v_i \to t'$.

Through the generalization of the aforementioned discussion, for each $i \in [n]$ we can infer that the set of values of $(\lambda_1, \ldots, \lambda_d)$ from $\mathbb{R}^d$ for which the path $s \to u_i \to t$ in graph G_1 is the shortest is specified by the set of linear inequalities $\mathcal{J}_i = \{L_i - L_j \leq 0 \mid j \in [n] \setminus i\}$[3]. Let the corresponding hyperplanes that create this region be $\mathcal{H}_i = \{L_i - L_j = 0 \mid j \in [n] \setminus i\}$. For the sake of exposition, let us assume that all our sets of inequalities are consistent.

We first note that these hyperplanes $\mathcal{H} = \mathcal{H}_1 \cup \cdots \cup \mathcal{H}_n$ partition[4] $\mathbb{R}^d$ such that each region thus created corresponds to exactly one path from s to t. Let us denote this partition by $\pi_{s,t}$. Note that a parametric shortest path can appear with multiplicity in more than one contiguous regions created by these hyperplanes. That is, we are creating a map from the regions created through the partition of $\mathbb{R}^d$ with the set of hyperplanes $\mathcal{H}$, to the parametric shortest paths. Analogously, we can obtain a partition $\pi_{s',t'}$ with the set of hyperplanes $\mathcal{H}'$ such that each region in it corresponds to a parametric shortest path between s' and t'. Note that the cardinality of $\mathcal{H} \cup \mathcal{H}'$ is at most $2\binom{n}{2}$.

Similar to our earlier analysis, by superimposing the partitions $\pi_{s,t}$ and $\pi_{s',t'}$, we get a new partition in which the regions simultaneously indicate the shortest paths between $s \rightsquigarrow t$ and $s' \rightsquigarrow t'$. It is important to note that the above analysis also holds for any graphs G_1 and G_2 (even when they share edges or vertices).

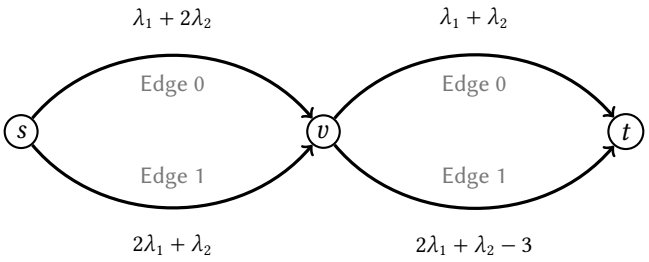
In case G_1 and G_2 were connected in series (with $t = s'$, source s and sink t'), the total number of $s \rightsquigarrow t'$ paths are n^2 many and we could have easily formed hyperplanes by comparing each path with the rest simultaneously. This could potentially create $\binom{n^2}{2}$ many hyperplanes and this is far more than the count of the regions obtained through superimposition.

Having established the superimposition principle for pairs of graphs, we now apply this technique recursively to a specific family of layered DAGs. This will allow us to derive concrete bounds on the number of parametric shortest paths.

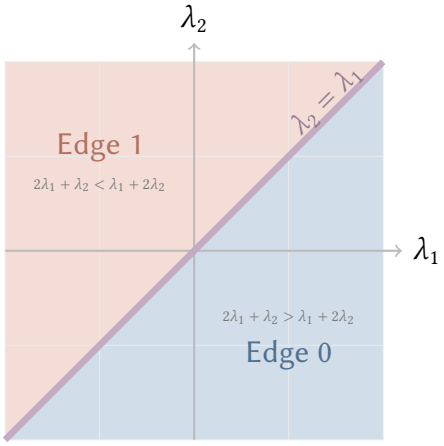
Assume that each layer is a complete bipartite graph and the edge weights vary as functions of d parameters $\lambda_1, \ldots, \lambda_d$. Let B be a set of pairs of vertices $\{(x, y) \mid x \in \text{Layer}_{2i-1}, y \in \text{Layer}_{2i+1}$ for $i \in \{1, \ldots, \lfloor \frac{n-1}{2} \rfloor\}\}$. That is, B contains ordered pairs of vertices across alternate odd layers. For each pair $(x, y) \in B$, there is a subgraph $G_{x,y}$ of length 2 with x as source and y as sink, and graph $G_{x,y}$ resembles the graphs G_1, G_2 (Figure 4) in structure. Using the aforementioned analysis, for each pair $(x, y) \in B$ we get a partition $\pi_{x,y}$ of $\mathbb{R}^d$ (through a set of hyperplanes $\mathcal{H} = \mathcal{H}_{x,y}$) such that every region in it corresponds to a parametric shortest path in $G_{x,y}$. By superimposing the partitions $\pi_{x,y}$ for all $(x, y) \in B$, we get finer regions of $\mathbb{R}^d$ such that each region *uniquely* identifies a parametric shortest path between all the pairs

[3] If the set of inequalities are inconsistent then the region would be empty. For example, with $L_1 = \lambda_1 - \lambda_2, L_2 = \lambda_2 - \lambda_1, L_3 = 1$, path 3 can never be a shortest path, as both $\lambda_1 - \lambda_2 \geq 1$ and $\lambda_2 - \lambda_1 \geq 1$ cannot be satisfied simultaneously.

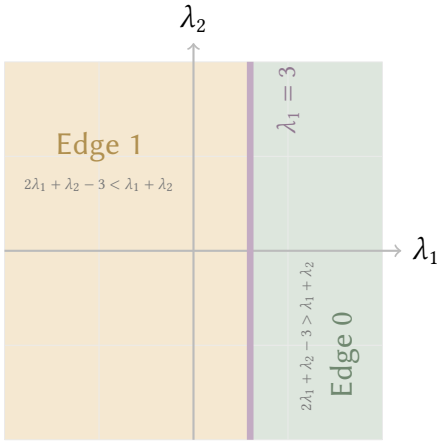
[4] That is, every point in $\mathbb{R}^d$ belongs to at least one of the regions $\mathcal{J}_1, \ldots, \mathcal{J}_n$ (assuming they are all consistent). For any arbitrary $\bar{x} \in \mathbb{R}^d$, compute the values $L_1(\bar{x}), \ldots, L_n(\bar{x})$ and take the argmin of these values. Call it i^*. It is easy to see that $\bar{x}$ simultaneously satisfies the set of inequalities $\mathcal{J}_{i^*}$ and lies in the region defined by it.



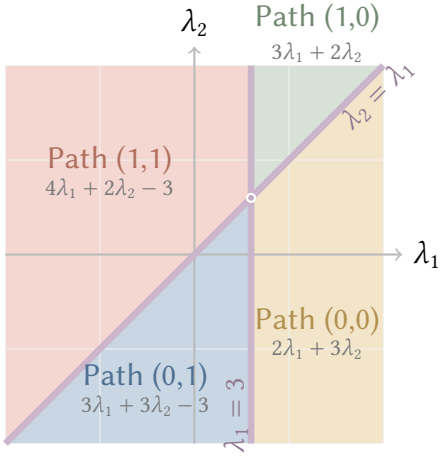
(a) Graph G with linear edge weights. Note the the edges above are denoted Edge 0, Edge 1, Edge 0, Edge 1. Correspondingly, the four paths from s to t are denoted Path (0,0), Path (0,1), Path (1,0), Path (0,1).



(b) Partition of $\mathbb{R}^2$ to indicate parametric shortest paths between s and v

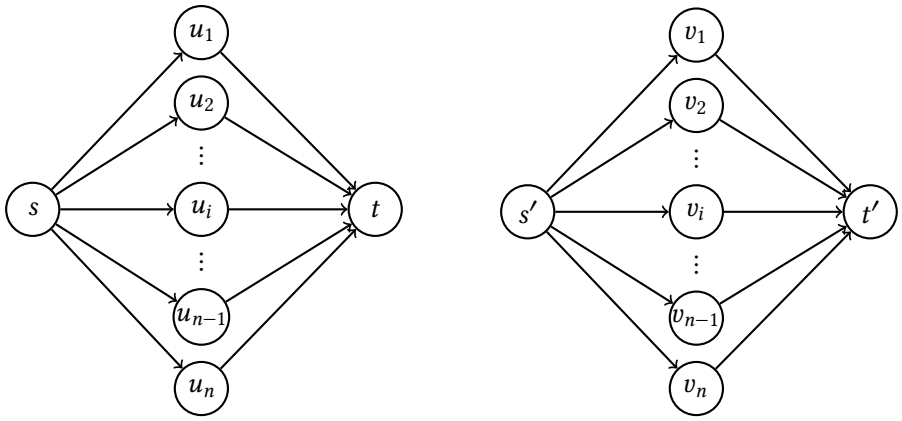


(c) Partition of $\mathbb{R}^2$ to indicate parametric shortest paths between v and t



(d) Overlaying (superimposing) the partitions in (b) and (c)

Figure 3: Partition of $\mathbb{R}^2$ by hyperplanes created by regions corresponding to shortest paths from s to t in G



(a) Graph G_1 of length 2

(b) Graph G_2 of length 2

Figure 4: Disjoint graphs G_1 and G_2

(x, y). In other words, these finer regions of $\mathbb{R}^d$ are generated by partitioning it with the set of hyperplanes

$$\mathcal{H} = \bigcup_{(x,y)\in B} \mathcal{H}_{x,y}\,.$$

For each region thus created, we create a new DAG instance $G^{(x,y)}_{n,n/2}$ from $G_{n,n}$, with the following properties. Delete all vertices in even layers, along with the incident edges. Connect each pair $(x, y) \in B$ using a new edge such that the weight of this new edge (x, y) is the cumulative weight of the shortest $x \rightsquigarrow y$ path identified through this region.

For each region created through the hyperplanes, we create a new instance of the parametric shortest path problem on a graph of length $n/2$. The number of new instances created is equal to the number of regions generated by the hyperplanes $\mathcal{H}$ and this quantity can be bound efficiently (see Theorem 9). We recursively partition $\mathbb{R}^d$ through these new instances until we reach a trivial base case. Thus,

$$|\text{MSPC}(G_{n,n})| \leq (\#\text{ of regions created by } \mathcal{H}) \cdot \max_{(x,y)\in B} \left(|\text{MSPC}(G^{(x,y)}_{n,n/2})| \right)\,.$$

In this paper, we consider edge weights that either vary as linear functions of d real-valued parameters, and also edge weights that are univariate polynomials of degree at most d. The proof technique for the latter is not very different from the one for the former.

1.5 Paper Roadmap

We set up some necessary background for our proofs in Section 2. We explain the full details of our partitioning of $\mathbb{R}^d$ using $\mathsf{poly}(n)$ hyperplanes in Section 3.1. We then present our proof for directed acyclic graphs (DAGs) (Theorem 2) in Section 3.2. We show that the result for DAGs can be lifted to count the size of an MSPC over a *feasible*[5] region $S \subseteq \mathbb{R}^d$, via a reduction to directed graphs with no negative-weight cycles (Theorem 3) and undirected graphs with no negative-weight edges (Corollary 4) in Section 3.3. We prove our upper bound for univariate polynomial edge weights (Theorem 5) in Section 3.4. Following that,

[5]A feasible region is the subset of $\mathbb{R}^d$ (or the set of values of $\overline{\lambda} = (\lambda_1, \lambda_2 \ldots, \lambda_d)$) wherein the given directed graph (respectively, undirected graph) contains no negative-weight cycles (respectively, negative-weight edges). See Definition 8 for a formal definition.

we explain our shortest path oracle for linear edge weights (Theorem 6) in Section 4.1, and for univariate polynomial edge weights (Theorem 7) in Section 4.2. We conclude with a discussion on open problems in Section 5.

2 Preliminaries

In this section, we establish the terminologies and definitions that will be used throughout the remainder of this work.

2.1 Basic Notation

For a positive integer $n \in \mathbb{N}$, we use the notation $[n]$ to denote the set $\{1, 2, \ldots, n\}$. Unless otherwise explicitly stated, all graphs considered in this work are simple (containing no multi-edges or self-loops), directed, and connected, with vertex set V where $|V| = n$. All logarithms considered in this paper are taken to the base 2 unless explicitly stated otherwise.

When working with parameter vectors, we employ the notation $\overline{\lambda} = (\lambda_1, \lambda_2 \ldots, \lambda_d)$ to represent a d-tuple of real-valued parameters $\lambda_i \in \mathbb{R}$ for $i \in [d]$. For a subset $S \subseteq \mathbb{R}^d$ of the parameter space, we define $\mathrm{MSPC}_S(G)$ as the collection of parametric shortest paths from a designated source vertex s to a sink vertex t when the parameter vector takes values in S. As a notational convenience, when $S = \mathbb{R}^d$ (i.e., when considering the entire parameter space), we use the abbreviated notation $\mathrm{MSPC}(G) := \mathrm{MSPC}_{\mathbb{R}^d}(G)$. In cases where the source and sink vertices differ from s and t, we will specify them explicitly.

2.2 Feasible Regions

The shortest path problem over directed graphs is studied only on graphs with no negative-weight cycles, as their presence could make the shortest path have a length of $-\infty$.

Moreover, the standard conversion of an undirected graph to a directed graph (by replacing each undirected edge by two directed edges in opposite directions) could create negative cycles if the undirected graph that we started with had negative weight edges.

To deal with such situations, we define the notion of a feasible region (similar feasible regions have been used in earlier works [Car83, GR19]).

Definition 8. *With respect to parametric shortest paths, a feasible region is defined as follows.*

- *Let $G = (V, E)$ be a directed graph whose edge weights are all linear functions of $\overline{\lambda} = (\lambda_1, \lambda_2 \ldots, \lambda_d)$. Then, a region $S \subseteq \mathbb{R}^d$ is called a feasible region for G if for every point $\overline{x} \in \mathbb{R}^d$, the fixed-edge weight graph obtained by substituting $\overline{\lambda} = \overline{x}$ in G contains no negative-weight cycles.*
- *Let $G = (V, E)$ be an undirected graph whose edge weights are all linear functions of $\overline{\lambda} = (\lambda_1, \lambda_2 \ldots, \lambda_d)$. Then, a region $S \subseteq \mathbb{R}^d$ is called a feasible region for G if for every point $\overline{x} \in \mathbb{R}^d$, the fixed-edge weight graph obtained by substituting $\overline{\lambda} = \overline{x}$ in G contains no negative-weight edges.*

2.3 Supporting Results from Combinatorial Geometry

The analysis of parametric shortest paths often requires results from combinatorial geometry concerning hyperplane arrangements. We state here a classical result that bounds the complexity of such arrangements.

Theorem 9 (Hyperplane Arrangement Complexity [VC15, Win66])**.** *Consider a d-dimensional Euclidean space $\mathbb{R}^d$ that is partitioned by t hyperplanes. Let r denote the number of regions formed by these hyperplanes. Then,*

$$r \leq \sum_{i=0}^{d} \binom{t}{i} \leq t^d + 1.$$

Theorem 9 provides a fundamental upper bound on the complexity of hyperplane arrangements and has direct implications for the number of parametric shortest paths in the setting where the edge weights vary as linear functions of d parameters.

2.4 Davenport-Schinzel Sequences

The above characterization does not help when the edge weights vary as univariate polynomials of degree d. In this setting, like the earlier works, we depend on the combinatorial characterization through Davenport-Schinzel sequences.

Definition 10. *Given a finite set of symbols X, a sequence $U = (u_1, u_2, \ldots, u_t)$ is a Davenport-Schinzel sequence of order s if it satisfies the following properties.*

- *$\forall i \in [t]$, u_i is a symbol coming from X,*
- *No two consecutive symbols in the sequence U are the same,*
- *If $x_1, x_2 \in X$ are distinct symbols, then U doesn't contain a subsequence $(\ldots, x_1, \ldots, x_2, \ldots, x_1, \ldots, x_2, \ldots)$ consisting of $s + 2$ alternations between x_1 and x_2.*

In this work, we use Davenport-Schinzel sequences to study the lower envelope formed by a set of univariate polynomials of degree at most d. Since any two degree d univariate polynomials can be equal in at most d points, They can alternate at most $d + 1$ many times. Therefore, the order of the corresponding Davenport-Schinzel sequence will be d. The Davenport-Schinzel sequences have tight bounds when the order of the sequence is constant (see [Dav71, ASS89, Niv10, Pet15]). However, when the degree d is arbitrary, we do not have good upper bounds on the size of Davenport-Schinzel sequences (of arbitrary order) that we can use. The only known upper bound on the size of Davenport-Schinzel sequences that is applicable here is the trivial $\binom{N}{2}d + 1$ (see [Kla02, p. 3]).

3 Proofs

3.1 Partitioning $\mathbb{R}^d$ using $\mathsf{poly}(n)$ Hyperplanes

We first present an abstracted result for distinct settings of edge weights that we consider in the paper. We then invoke the necessary space partitioning lemmas in each case and get the final bounds.

Theorem 11. *Let n, ℓ be natural numbers such that $\ell \leq n$. Let $S \subseteq \mathbb{R}^d$. Let G be a single-source and single-sink layered directed acyclic graph with ℓ layers and at most n vertices per layer. Let the edge weights be multivariate polynomials, denoted by $f_e(\lambda_1, \ldots, \lambda_d)$. Let $N := n^5$, and let the number of partitions of S through N hyperplanes be $T(N)$. Then,*

$$|\mathrm{MSPC}(G)| \in T(N)^{O(\log(\ell))}.$$

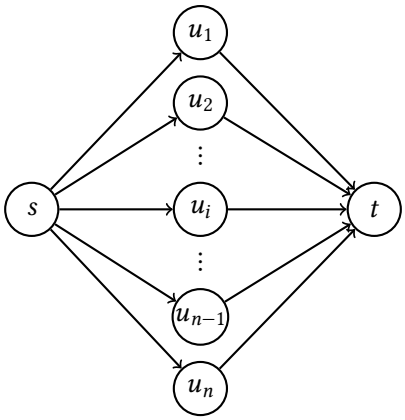


Figure 5: Graph G of length 2

Proof. Without loss of generality, assume that ℓ is a power of 2. Proof of this theorem proceeds via induction on the length of G.

Base case: Let $\ell = 2$. That is, the source s and the sink t are connected via at most n intermediate nodes $u_1, \ldots, u_n$ as shown in Figure 5. For all $i \in [n]$, ith path $s \to u_i \to t$ is the shortest path for those points $\overline{x} \in S$ that simultaneously satisfy the inequalities

$$\mathcal{J}_i = \{f_{(s,u_i)}(\overline{\lambda}) + f_{(u_i,t)}(\overline{\lambda}) \leq f_{(s,u_j)}(\overline{\lambda}) + f_{(u_j,t)}(\overline{\lambda}) \mid j \in [n] \setminus \{i\}\}.$$

Further, every point in S lies on or in one of the sides of these hyperplanes defined by the following equations.

$$\mathcal{H}_0 = \{f_{(s,u_i)}(\overline{\lambda}) + f_{(u_i,t)}(\overline{\lambda}) = f_{(s,u_j)}(\overline{\lambda}) + f_{(u_j,t)}(\overline{\lambda}) \mid 1 \leq i < j \leq n\}.$$

In other words, these hyperplanes partition the space S such that each point $\overline{x} \in S$ satisfies at least one of $\mathcal{J}_1, \ldots, \mathcal{J}_n$ and constructively we get that index by computing $\operatorname{argmin}_{i \in [n]}\{f_{(s,u_i)}(\overline{x}) + f_{(u_i,t)}(\overline{x})\}$. It is easy to see that $|\mathcal{H}_0|$ is at most n^2. In each of the regions in the partition created by $\mathcal{H}_0$, there is a unique parametric shortest path. So, the number of shortest paths is at most the number of regions formed by $\mathcal{H}_0$.

$$|\text{MSPC}(G)| \leq T(n^2) \leq T(n^5).$$

Inductive hypothesis: Assume that the statement is true for all lengths $\leq \ell/2$.

$$|\text{MSPC}(G)| \leq T(n^5)^{\log(\frac{\ell}{2})}.$$

Increment step: Let length of G be ℓ. Let B be the set of pairs of vertices in the alternate layers as follows.

$$B = \left\{(u,v) \mid u \in \text{Layer}_{2i-1}, v \in \text{Layer}_{2i+1} \text{ for all } 1 \leq i \leq \frac{\ell}{2} - 1\right\}.$$

For each pair $(u,v) \in B$, let the graph $G^{(u,v)}$ be the subgraph of G induced on the vertices u, v and all the vertices in the layer between u and v. This is similar in structure to the graph in the base case (see Figure 5). Similar to the base case, we obtain a partition $\pi_{(u,v)}$ of S through the hyperplanes $\mathcal{H}_{(u,v)}$ and within each region thus created, parametric shortest path between u and v is the same for every point.

Let $\mathcal{H} = \cup_{(u,v)\in B}\mathcal{H}_{(u,v)}$. As before, $|\mathcal{H}_{(u,v)}| \leq n^2$ and thus $|\mathcal{H}| \leq n^5$ (since $|B| \leq n^3$). Let π be the partition of the space S through the set of hyperplanes $\mathcal{H}$. Let $T = T(n^5)$ and $S_1, \ldots, S_T$ be the regions created by π.

Observation 12. *For every j, all points $\overline{x}$ within S_j are on the same side with respect to all hyperplanes in $\mathcal{H}$. In particular, for every pair $(u,v) \in B$, the shortest path from u to v is the same at all points within S_j. For points that lie on the separating hyperplanes, a shortest path is chosen according to a tie-breaking convention (order the vertices of the graph is some arbitrary way, and choose the path that occurs lexicographically earlier).*

Using this observation, we construct T many new graphs G_j corresponding to each region S_j, as follows.

- Vertex set of G_j consists of all vertices in the odd layers of the graph G, and t.
- For each pair $(u,v) \in B$, connect it with an edge whose edge weight is given by the parametric weight of the unique parametric shortest path between u and v corresponding to the region S_j.
- Retain all the incoming edges into t along with the original edge weights.

We now make the following claim.

Claim 13. *Let $\mathrm{MSPC}_S(G)$ denote the set of s to t paths that show up as shortest paths of G in S. Let $\mathrm{MSPC}_{S_j}(G_j)$ be defined similarly. Then,*

$$|\mathrm{MSPC}_S(G)| \leq \sum_{j=1}^{T} |\mathrm{MSPC}_{S_j}(G_j)|\,.$$

Proof of Claim 13: Towards the proof, it is sufficient to establish an injective map from $\mathrm{MSPC}_S(G)$ to $\cup_{i=1}^{T}\mathrm{MSPC}_{S_j}(G_j)$, in two steps.

1. For every path $P \in \mathrm{MSPC}_S(G)$ there is a corresponding path $\tilde{P} \in \bigcup_{i=1}^{T} \mathrm{MSPC}_{S_j}(G_j)$.
2. Two distinct paths P_1, P_2 cannot map to the same path in $\bigcup_{i=1}^{T} \mathrm{MSPC}_{S_j}(G_j)$.

Let P be an arbitrarily chosen path from $\mathrm{MSPC}_S(G)$. Let $R_P \subseteq S$ be the region of parameter values over which P is the shortest path from s to t in G. Since the regions $S_1, \ldots, S_T$ partition S, and $R_P \subseteq S$, we have that R_P must intersect at least one of the regions from $S_1, \ldots, S_T$. Let $R_P \cap S_j \neq \emptyset$ for some $j \in [T]$. A key observation that we make here is that between any two vertices $u \in \mathrm{Layer}_{2i-1}$ and $v \in \mathrm{Layer}_{2i+1}$ that P passes through, the sub-path of P from u to v must be the same as the unique shortest path between u and v in G for every $\overline{x} \in S_j$. For each pair of consecutive odd layers, we replace each segment of P with the unique shortest path between its endpoints in S_j, and concatenating these gives a corresponding path $\tilde{P} \in \mathrm{MSPC}_{S_j}(G_j)$.

Let us suppose that two distinct paths $P_1, P_2 \in \mathrm{MSPC}_S(G)$ correspond to the same projected path $\tilde{P} \in \mathrm{MSPC}_{S_j}(G_j)$. Then P_1 and P_2 must pass through the same vertices in the odd layers. Now we argue that for every pair (u,v) in the consecutive odd layers, the intermediate vertex between u and v on both the paths must be the same[6]. If the intermediate nodes for paths P_1 and P_2 were distinct within a region S_j for a pair $(u,v) \in B$, this contradicts the uniqueness of the shortest path between u and v for that region (where the uniqueness is guaranteed by the definition of S_j and construction of G_j). □

Claim 13 shows that a path $P \in \mathrm{MSPC}_S(G)$ could appear as a projection in various $\mathrm{MSPC}_{S_j}(G_j)$ but two distinct paths $P_1, P_2 \in \mathrm{MSPC}_S(G)$ do not project down to the same path in $\mathrm{MSPC}_{S_j}(G_j)$. Graph G_j has length $\ell/2$ (and $\ell/2+1$ layers) and from the inductive hypothesis, we have $|\mathrm{MSPC}_{S_j}(G_j)| \leq T^{\log(\ell/2)}$ for each j. Thus,

$$|\mathrm{MSPC}_S(G)| \leq T \cdot T^{\log(\ell/2)} = T^{1+\log(\ell/2)} = T^{\log(\ell)} \in T(n^5)^{O(\log(\ell))}.$$ □

[6]Here, we implicitly invoke the tie-breaking convention described in Observation 12.

3.2 Main Result: Directed Acyclic Graphs (Theorem 2)

Here, we prove our main result for directed, acyclic graphs with linear edge weights.

Theorem 2. *Let G be an n-vertex directed acyclic graph with edge weights of the form $a_{e,1}\lambda_1 + a_{e,2}\lambda_2 + \cdots + a_{e,d}\lambda_d + a_{e,d+1}$. Then,*

$$|\mathrm{MSPC}(G)| \in n^{O(d \log n)}.$$

Proof. This proof simply invokes Theorem 11 for the specific case of linear edge weights. The edge weights are of the form

$$\mathsf{wt}(e) = \sum_{i \in [d]} a_{e,i}\lambda_i + a_{e,d+1}.$$

Then, the set $\mathcal{H}$ contains at most n^5 hyperplanes and $T(n^5)$ refers to the number of regions formed by the hyperplanes in $\mathcal{H}$. Due to Theorem 9, we get

$$T(n^5) \leq n^{5d} + 1.$$

This along with the fact $\ell \leq n$ gives us the required upper bound.

$$|\mathrm{MSPC}(G)| \leq (n^{5d} + 1)^{\log(n)} \in n^{O(d \log(n))}. \qquad \square$$

We now supplement our upper bound with a matching lower bound.

Claim 14. *For all positive integers n, there exists a directed acyclic graph $G^* = (V, E)$ with edge weights of the form $a_{e,1}\lambda_1 + a_{e,2}\lambda_2 + \cdots + a_{e,d}\lambda_d + a_{e,d+1}$ such that $|\mathrm{MSPC}(G^*)| \in n^{\Omega(d \log n)}$.*

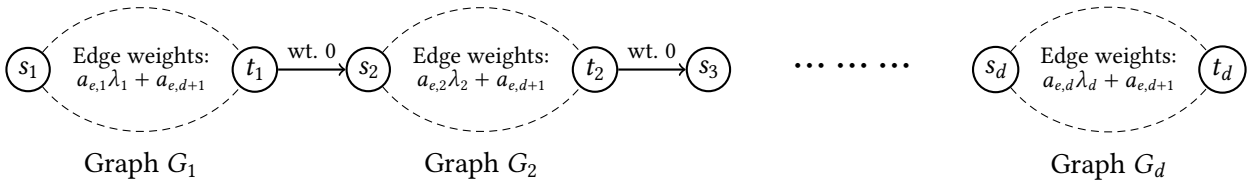


Figure 6: A schematic of the graph G^* used in the proof of Claim 14

Proof Sketch. Fix an n and a d. We know there exists an n-vertex DAG G with edge weights of the form $a_{e,1}\lambda_1 + a_{e,2}$ whose MSPC is of size $n^{\Omega(\log(n))}$. Such graphs can be found in any of the following papers: [Car83, MS00, GR19]. Now, let us describe our construction of the DAG G^* (Figure 6).

Let $G_1, G_2, \ldots, G_d$ be d identical and disjoint copies of G; thus, we have $|\mathrm{MSPC}(G_i)| \in n^{\Omega(\log(n))}$ for each G_i. The graph G_1 has start and end vertices s_1 and t_1 and edge weights of the form $a_{e,1}\lambda_1 + a_{e,d+1}$, the graph G_2 has start and end vertices s_2 and t_2 and edge weights of the form $a_{e,2}\lambda_2 + a_{e,d+1}$, and so on, the graph G_d has start and end vertices s_d and t_d and edge weights of the form $a_{e,d}\lambda_d + a_{e,d+1}$.

Finally, these graphs are connected with each other in series to obtain G^*: there is a 0-weight directed edge from t_1 to s_2, there is a 0-weight directed edge from t_2 to s_3, and so on, there is a 0-weight directed

edge from t_{d-1} to s_d. Hence, the start vertex of the graph G^* is s_1 and its end vertex is t_d. Since G_i and G_j (for all $i \neq j$) do not have any variables in common, we obtain the following.

$$\begin{aligned}|\mathrm{MSPC}(G^*)| &= |\mathrm{MSPC}(G_1)| \times |\mathrm{MSPC}(G_2)| \times \cdots \times |\mathrm{MSPC}(G_d)| \\ &= n^{\Omega(\log(n))} \times n^{\Omega(\log(n))} \times \cdots \times n^{\Omega(\log(n))} \\ &= n^{\Omega(d \log(n))}.\end{aligned}$$

□

3.3 Directed and Undirected Graphs (Theorem 3 and Corollary 4)

In this section, we show a reduction from undirected and directed graphs to directed acyclic graphs (DAGs).

Let $G = (V, E)$ be an n-vertex graph (either a directed graph with no negative-weight cycles or an undirected graph with non-negative edge weights[7]) where $V = \{1, \ldots, n\}$ and $s, t \in V$. The weight of each edge $e \in E$ is a function of d real-valued parameters, $\overline{\lambda} = (\lambda_1, \ldots, \lambda_d)$:

$$\mathsf{wt}_e(\overline{\lambda}) = \sum_{j=1}^{d} a_{e,j}\lambda_j + a_{e,d+1}\,.$$

The total weight of an s-t path P is $f_P(\overline{\lambda}) = \sum_{e \in P} \mathsf{wt}_e(\overline{\lambda})$. Since we may assume that we are working in a feasible region (Definition 8) of G, for all substitutions of $\overline{\lambda}$, all directed cycles in G are of non-negative weight.

We will now construct a directed acyclic graph $G' = (V', E')$ from G as follows. The new vertex set V' consists of n copies of each vertex spread across n layers. Formally, $V' = \{(v, \ell) \mid v \in V, \ell \in \{0, 1, \ldots, n-1\}\}$. Thus the total number of vertices in G' is $|V'| = n^2$.

A directed edge (u, i) to $(v, i+1)$ is added to the edge set E' in G' if and only if the directed edge (u, v) appears as the $i+1^{\text{th}}$ edge in any of the simple s to t paths of the original graph G and $i \leq n-1$. The weight of an edge in G' is inherited from the corresponding edge in G.

$$\mathsf{wt}'_{((u,i),(v,i+1))}(\overline{\lambda}) = \mathsf{wt}_{(u,v)}(\overline{\lambda})$$

The new source is $s' = (s, 0)$ and the new set of sinks is $T' = \{(t, \ell) \mid \ell \in \{1, \ldots, n-1\}\}$. Edges only run from layer i to layer $i+1$, and this ensures that the graph is acyclic. A shortest path in G' at a substitution $\overline{x}$ is the shortest among all $(s, 0)$ to (t, i) (for all i) paths. It is important to note that there could be some s to t walks in G appear as simple paths in G'. Just from the construction, it is not evident that the parametric shortest paths in G are *preserved* in G' due to addition of some walks of G. We will now show that the preservation does happen when we start with a directed graph G with no negative-weight cycles.

Let $P = (s, u_1, u_2, \ldots, u_{\ell-1}, t)$ be a shortest path of length ℓ between s and t in G for a substitution of $\overline{\lambda}$ by $\overline{x} \in \mathbb{R}^d$. From the construction of G', we get that this path P manifests as a path $P' = ((s, 0), (u_1, 1), \ldots, (u_{\ell-1}, \ell-1), (t, \ell))$ in G'. We need to show that P' is also a parametric shortest path in G' at $\overline{x}$. Through the inheritance of edge weights in the construction of G', we get that path weights of P and P' are equal. That is, $f_P(\overline{x}) = f_{P'}(\overline{x})$

For the sake of contradiction, let us suppose that there is a path $Q' = ((s, 0), (v_1, 1), \ldots, (v_{r-1}, r-1), (t, r))$ that is the shortest path in G' at $\overline{x}$ and $f_P(\overline{x}) > f_{Q'}(\overline{x})$. Let the walk $W = (s, v_1, v_2, \ldots, v_{r-1}, t)$ in G correspond to Q' and thus the weight of the path Q' in G' is equal to the weight of the walk W in G. If W were also a simple path and then $f_W(\overline{x}) = f_{Q'}(\overline{x}) < f_{P'}(\overline{x}) = f_P(\overline{x})$. This contradicts the optimality of P at

[7] See Definition 8 for the meaning of negative-weight cycles and negative-weight edges when the edge weights are not fixed.

$\overline{x}$. On the other hand, if W was actually a walk and not a simple path, then elimination of cycles in W (with non-negative weights) would have created a path P_W with a shorter weight or shorter length[8], at $\overline{x}$. This again contradicts the optimality of P at $\overline{x}$. Using similar arguments, we can handle the case when $f_{P'}(\overline{x}) = f_{Q'}(\overline{x})$.

Combining Theorem 2 with this reduction, we obtain the following.

Theorem 3. *Let G be an n-vertex directed graph without negative-weight cycles with edge weights of the form $a_{e,1}\lambda_1 + a_{e,2}\lambda_2 + \cdots + a_{e,d}\lambda_d + a_{e,d+1}$. Then,*

$$|\mathrm{MSPC}(G)| \in n^{O(d \log n)}.$$

Corollary 4. *Let G be an n-vertex undirected graph without negative-weight edges with edge weights of the form $a_{e,1}\lambda_1 + a_{e,2}\lambda_2 + \cdots + a_{e,d}\lambda_d + a_{e,d+1}$. Then,*

$$|\mathrm{MSPC}(G)| \in n^{O(d \log n)}.$$

3.4 Univariate Polynomial Edge Weights (Theorem 5)

Theorem 5. *Let G be an n-vertex directed graph with edge weights of the form $a_{e,q}\lambda^q + a_{e,q-1}\lambda^{q-1} + \cdots + a_{e,2}\lambda^2 + a_{e,1}\lambda + a_{e,0}$. Then,*

$$|\mathrm{MSPC}(G)| \in n^{O(\log n + \log q)}.$$

Proof. We simply invoke Theorem 11 for the specific case of univariate polynomial edge weights; that is, edge weights of the form

$$\mathsf{wt}(e) = \sum_{i=0}^{q} a_i \lambda^i.$$

Then, the set $\mathcal{H}$ contains at most n^5 degree q, univariate polynomials and $T(n^5)$ refers to the number of line segments the real number line gets split into by polynomials in $\mathcal{H}$. To upper bound $T(n^5)$, we use the trivial upper bound on Davenport-Schinzel sequences (see Section 2.4).

$$T(n^5) \leq \binom{n^5}{2} q + 1.$$

This along with the fact that $l \leq n$ gives us the required upper bound.

$$|\mathrm{MSPC}(G)| \leq \left(\binom{n^5}{2} q + 1\right)^{\log(n)} \in (nq)^{O(\log(n))} \in n^{O(\log(n) + \log(q))}. \qquad \square$$

4 Shortest Path Oracles

4.1 Linear Edge Weights (Theorem 6)

In this subsection, we present a shortest path identification data structure for a parametric graph $G = (V, E)$ with linear edge weights. The data structure does the following: Given $\bar{x} \in \mathbb{R}^d$ as input, it outputs the shortest path from s to t when $\bar{\lambda} = \bar{x}$.

[8]Note that if we started with an undirected graph, this lifting argument would only work if all its edge weights were positive. In particular, if the undirected graph G has a negative-weight edge, it will lead to a negative-weight cycle of length 2, and a walk that goes back and forth over that 2-cycle could lead to a non-contradictable path Q' in G' of weight lower than all s to t paths in G.

Our construction of this data structure relies on the existing literature on *Point-Location* problem, and we will recall that briefly here.

Let $\mathcal{H}$ denote a set of hyperplanes in the space $\mathbb{R}^d$. The point-location problem is to preprocess $\mathcal{H}$ into a data structure that supports efficient point-location queries. A point-location query inputs a point $\bar{x} \in \mathbb{R}^d$ and asks to identify the cell (formed by hyperplanes in H) that contains $\bar{x}$. Toward this, Meiser [Mei93] introduced a data structure which was later improved by Ezra, Har-Peled, Kaplan & Sharir [EHPKS20].

Theorem 15 (Theorem 5.4 [EHPKS20])**.** *Given a set $\mathcal{H}$ of hyperplanes in the space $\mathbb{R}^d$, there exists a data structure that answers point-location queries in time $O(d^4 \log(|\mathcal{H}|))$. The data structure requires $n^{O(d)}$ space and $n^{\tilde{O}(d)}$ preprocessing time.*

Due to Theorem 2, $|\mathrm{MSPC}(G)| \in n^{O(d \log(n))}$. That is, $\mathbb{R}^d$ is partitioned into $n^{O(d \log(n))}$ many regions with each region corresponding to exactly one path. If we knew all the hyperplanes that partition $\mathbb{R}^d$ into these regions, then we could directly use the data structure from Theorem 15. Since that is not the case at any intermediate step in the proof of Theorem 11, we recursively nest the point-location data structure instead.

Theorem 6. *Let G be an n-vertex directed graph with edge weights of the form $a_{e,1}\lambda_1 + a_{e,2}\lambda_2 + \cdots + a_{e,d}\lambda_d + a_{e,d+1}$. Then, there exists a data structure that takes as input an $\overline{x} \in \mathbb{R}^d$, and outputs a shortest path in the graph G at $\overline{\lambda} = \overline{x}$ in $O(d^4 \log^2 n)$ time. The data structure takes $n^{\tilde{O}(d)}$ space and $n^{\tilde{O}(d)}$ preprocessing time.*

Proof. We begin by observing that every region in any given recursion step in the proof of Theorem 11 gets partitioned by at most $O(n^5)$ hyperplanes. Given a $\bar{x} \in \mathbb{R}^d$, we can query the point-location data structure to identify the region it belongs to in the first level of recursion. With this knowledge of the region in the first level of recursion, we get $O(n^5)$ more hyperplanes per region in the next level of recursion. In this way, we adaptively query the point-location data structure. Following through the recursion to its last level, we observe that each region in the last level corresponds to exactly one path from s to t. This path is the shortest path for all points in that region (including $\bar{x}$). Therefore, we output this path.

To construct the path identification data structure, we nested the point-location data structures in each of the $O(\log(n))$ levels. Note that the number of instances of the point-location data structure at a depth i of the recursion is at most $O(n^{5id})$.

$$\text{Total number of instances} = \sum_{i=0}^{O(\log n)} O(n^{5id}) \in n^{O(d \log(n))}.$$

The original point-location data structure has a query time of $O(k^4 \log(n))$, requires $n^{O(d)}$ space and uses $n^{\tilde{O}(d)}$ preprocessing time. Thus, the path identification data structure, as constructed above, for a graph with n vertices and edge weights linear in d parameters has the following properties.

- Overall query time is $O(d^4 \log^2(n))$,
- Total space required is $n^{O(d)} \cdot n^{O(d \log(n))} \in n^{\tilde{O}(d)}$,
- Preprocessing time required to construct the data structure is $n^{\tilde{O}(d)} \cdot n^{O(d \log(n))} \in n^{\tilde{O}(d)}$. □

4.2 Univariate Polynomial Edge Weights (Theorem 7)

Theorem 7. *Let $G = (V, E)$ be an n-vertex directed graph with edge weights of the form $a_{e,q}\lambda^q + a_{e,q-1}\lambda^{q-1} + \cdots + a_{e,2}\lambda^2 + a_{e,1}\lambda + a_{e,0}$. Then, there exists a data structure that takes as input an $x \in \mathbb{R}$, and outputs a shortest path in the graph G at $\lambda = x$ in $O(\log^2 n + \log n \log q)$ time. The data structure takes $(nq)^{\tilde{O}(1)}$ space and $(nq)^{\tilde{O}(1)}$ preprocessing time.*

Proof. The data structure we construct is an array in which each element corresponds to a line segment. Every element of the array will be a tuple containing the endpoints of a line segment of $\mathbb{R}$ and the label of the shortest path from s to t corresponding to the line segment.

To construct this, we first begin with a single element $(-\infty, \infty, G)$ in the array. In the preprocessing step, as shown in the proof of Theorem 11, we obtain the subgraphs $\{G_{S_i}\}_{i\in[T]}$ and the endpoints of each S_i. We update the array to store this in sorted order. We repeat this for each subgraph until each line segment in the array corresponds to a single path from s to t.

Query time: When we get a query with the value of λ, we simply perform a binary search on the array to find the line segment that contains λ. Then, we output the path that corresponds to the line segment. Since the array can contain at most $n^{O(\log(n)+\log(q))}$ elements, the time taken to respond to a query will be $O(\log^2(n) + \log(n)\log(q))$.

Space required and preprocessing time: The number of line segments that are formed in iteration i is given by $(n^{10}q)^i$. Furthermore, there is a $\mathsf{poly}(n)$ overhead for each iteration.

$$\text{Number of line segments} = \sum_{i=0}^{\log n} (n^{10}q)^i \in (nq)^{O(\log(n))}.$$

This array contains an entry for each parametric shortest path. Therefore, the space required is $n^{O(\log(n)+\log(q))} \in (nq)^{\tilde{O}(1)}$. The total preprocessing time is $(nq)^{\tilde{O}(1)}\mathsf{poly}(n)$. □

5 Future Directions

There are several directions of research that can be pursued for parametric shortest paths. Here, we outline three of them.

(i) For all undirected n-vertex graphs G with edge weights of the form $a_{e,1}\lambda_1 + a_{e,2}\lambda_2 + \cdots + a_{e,d}\lambda_d + a_{e,d+1}$, is it true that

$$|\text{MSPC}(G)| \in n^{O(d\log n)}?$$

This seems like the easiest and most immediate open problem to tackle. Since we have already proved the upper bound for DAGs and directed graphs, and we know that such results hold for constant d for undirected graphs, it seems reasonable that this should be true as well.

(ii) The upper bound for univariate polynomial edge weights (of degree q) is either $n^{O(\log(n)+\log(q))}$ or $n^{\log(n)+(\alpha(n)+O(1))^q}$, depending on how q varies with n. Improving this upper bound (or proving a matching lower bound) is another interesting avenue. This seems like a tricky problem, but it may require only a few new ideas to go along with the ones we already have.

(iii) Finally, the "holy grail" of parametric shortest paths would be multivariate polynomials (d variables, degree q). The only known lower bound is $n^{\Omega(d\log(n))}$ and no non-trivial upper bound is known. Note that a good upper bound in this setting should be able to recover the upper bound in item (i) (by setting $q = 1$), and also recover the upper bound in item (ii) (by setting $d = 1$). For example, something like $n^{O(d\log(n)+\log(q))}$ would make sense.

However, simply combining ideas from the $d = 1$ and $q = 1$ cases seems to be far from enough. The behavior of regions and curves with arbitrary d's and q's varies quite wildly (e.g., self-intersecting curves, disjoint regions for the same weight function), and it seems difficult to capture all of them in a nice combinatorial way. Solving this problem in its full generality may require substantial mathematical insights.

6 Acknowledgments

We are deeply grateful to Aryaman Manish Kolhe for many helpful discussions in the earlier stages of this work, especially for studying various toy examples and plotting their partitions in $\mathbb{R}^2$, which greatly aided our analysis. K.G. thanks Jaikumar Radhakrishnan for hosting him at ICTS-TIFR, elucidating the proof of Barth, Funke & Proissl [BFP22], and the subsequent discussions about it with Prerona Chatterjee, which resulted in a slight improvement to the upper bound [CGR23]. S.C. and K.G. thank Kavitha Telikepalli for suggesting that this work might be useful in shortest path oracles. N.R. thanks Emanuel Juliano for helpful discussions. We also thank Jaikumar Radhakrishnan for carefully verifying the proof of our main result. Finally, we thank the anonymous reviewers of this paper for pointing out some minor errors and for making several helpful suggestions that enhanced its presentation.

N.R. acknowledges partial support from the Dutch Ministry of Education, Culture, and Science through Gravitation project "Challenges in Cyber Security – 024.006.037" for this work.

Statement of AI use: The authors did not use any AI tools or AI assistants at any stage of this work.

References

[ASS89] P. K. Agarwal, M. Sharir, and P. Shor. Sharp upper and lower bounds on the length of general Davenport-Schinzel sequences. *J. Combin. Theory Ser. A*, 52(2):228–274, 1989. 11

[Bel58] Richard Bellman. On a routing problem. *Quarterly of applied mathematics*, 16(1):87–90, 1958. 3

[BFP22] Florian Barth, Stefan Funke, and Claudius Proissl. An upper bound on the number of extreme shortest paths in arbitrary dimensions. In *30th annual European Symposium on Algorithms*, volume 244 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 14, 12. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2022. 5, 6, 19

[Car83] Patricia June Carstensen. *THE COMPLEXITY OF SOME PROBLEMS IN PARAMETRIC LINEAR AND COMBINATORIAL PROGRAMMING*. ProQuest LLC, Ann Arbor, MI, 1983. Thesis (Ph.D.)–University of Michigan. 3, 10, 14

[CGR23] Prerona Chatterjee, Kshitij Gajjar, and Jaikumar Radhakrishnan. Unpublished manuscript, 2023. 5, 19

[CT65] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965. 3

[Dav71] H. Davenport. A combinatorial problem connected with differential equations. II. *Acta Arith.*, 17:363–372, 1970/71. 11

[Dij59] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959. 3

[DYQ08] Bolin Ding, Jeffrey Xu Yu, and Lu Qin. Finding time-dependent shortest paths over large graphs. In *Proceedings of the 11th international conference on Extending database technology: Advances in database technology*, pages 205–216, 2008. 3

[EHPKS20] Esther Ezra, Sariel Har-Peled, Haim Kaplan, and Micha Sharir. Decomposing arrangements of hyperplanes: VC-dimension, combinatorial dimension, and point location. *Discrete Comput. Geom.*, 64(1):109–173, 2020. 17

[Eri10] Jeff Erickson. Maximum flows and parametric shortest paths in planar graphs. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 794–804. SIAM, 2010. 3

[FG85] Alan M Frieze and Geoffrey R Grimmett. The shortest-path problem for graphs with random arc-lengths. *Discrete Applied Mathematics*, 10(1):57–77, 1985. 3

[FHS14] Luca Foschini, John Hershberger, and Subhash Suri. On the complexity of time-dependent shortest paths. *Algorithmica*, 68(4):1075–1097, 2014. 3

[FJ56] Lester R Ford Jr. Network flow theory. Technical report, 1956. 3

[Flo62] Robert W Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345–345, 1962. 3

[GR19] Kshitij Gajjar and Jaikumar Radhakrishnan. Parametric shortest paths in planar graphs. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science*, pages 876–895. IEEE Comput. Soc. Press, Los Alamitos, CA, [2019] ©2019. 4, 10, 14

[GVCR21] Kshitij Gajjar, Girish Varma, Prerona Chatterjee, and Jaikumar Radhakrishnan. Generalized parametric path problems. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021, Virtual Event, July 27-30, 2021*, volume 161 of *Proceedings of Machine Learning Research*, pages 536–546. PMLR, 2021. 3

[HJB85] Michael T. Heideman, Don H. Johnson, and C. Sidney Burrus. Gauss and the history of the fast Fourier transform. *Arch. Hist. Exact Sci.*, 34(3):265–277, 1985. 3

[HNR68] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. 3

[Joh77] Donald B Johnson. Efficient algorithms for shortest paths in sparse networks. *Journal of the ACM (JACM)*, 24(1):1–13, 1977. 3

[Kla02] Martin Klazar. Generalized Davenport-Schinzel sequences: results, problems, and applications. *Integers*, 2:A11, 39, 2002. 11

[Mei93] S. Meiser. Point location in arrangements of hyperplanes. *Inform. and Comput.*, 106(2):286–303, 1993. 17

[Moo59] Edward F Moore. The shortest path through a maze. In *Proc. of the International Symposium on the Theory of Switching*, pages 285–292. Harvard University Press, 1959. 3

[MS00] Ketan Mulmuley and Pradyut Shah. A lower bound for the shortest path problem. In *15th Annual IEEE Conference on Computational Complexity (Florence, 2000)*, pages 14–21. IEEE Computer Soc., Los Alamitos, CA, 2000. 4, 14

[Nik09] Evdokia Nikolova. *Strategic algorithms*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2009. 4

[Niv10] Gabriel Nivasch. Improved bounds and new techniques for Davenport-Schinzel sequences and their generalizations. *J. ACM*, 57(3):Art. 17, 44, 2010. 11

[NKBM06] Evdokia Nikolova, Jonathan A. Kelner, Matthew Brand, and Michael Mitzenmacher. Stochastic shortest paths via quasi-convex maximization. In *Algorithms—ESA 2006*, volume 4168 of *Lecture Notes in Comput. Sci.*, pages 552–563. Springer, Berlin, 2006. 3

[Pet15] Seth Pettie. Sharp bounds on Davenport-Schinzel sequences of every order. *J. ACM*, 62(5):Art. 36, 40, 2015. 11

[R+59] Bernard Roy et al. Transitivité et connexité. *CR Acad. Sci. Paris*, 249(216-218):182, 1959. 3

[RR96] Ganesan Ramalingam and Thomas Reps. An incremental algorithm for a generalization of the shortest-path problem. *Journal of Algorithms*, 21(2):267–305, 1996. 3

[RZ04] Liam Roditty and Uri Zwick. On dynamic shortest paths problems. In *European Symposium on Algorithms*, pages 580–591. Springer, 2004. 3

[VC15] V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity*, pages 11–30. Springer, Cham, 2015. Reprint of Theor. Probability Appl. **16** (1971), 264–280. 11

[War62] Stephen Warshall. A theorem on boolean matrices. *Journal of the ACM (JACM)*, 9(1):11–12, 1962. 3

[WCH+14] Huanhuan Wu, James Cheng, Silu Huang, Yiping Ke, Yi Lu, and Yanyan Xu. Path problems in temporal graphs. *Proceedings of the VLDB Endowment*, 7(9):721–732, 2014. 3

[Win66] R. O. Winder. Partitions of N-space by hyperplanes. *SIAM J. Appl. Math.*, 14:811–818, 1966. 11

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il