Technische Universität Berlin

Bachelor Thesis

Finding the Most Vital Edges for Shortest Paths

Algorithms and Complexity for Special Graph Classes

presented by
Maximilian Stahlberg

supervised by
Till Fluschnik
Dr. André Nichterlein
Prof. Dr. Rolf Niedermeier

February 18, 2016

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin, den

_____

Unterschrift

Abstract

We study the NP-hard problem of finding the most vital edges for shortest paths between two terminals of an undirected graph, that are a small number of edges whose deletion increases the distance between the terminals significantly. The associated decision problem asks whether a given edge deletion budget is sufficient to achieve the desired length increase. We study this decision problem as well as three variants, where edges are assigned a length or a deletion cost or both, in the context of selected graph classes. We show that the problem remains NP-complete on all superclasses of complete graphs when either edge weight is non-uniform. In the case where both are uniform, we give linear-time decision algorithms for complete and threshold graphs and conjecture the correctness of a polynomial-time algorithm that decides the problem on proper interval graphs. We also look into multigraphs and provide two FPT algorithms for the two-terminal series-parallel graphs that have polynomial running time if at least one of the edge weights is uniform.

Zusammenfassung

Als die wichtigsten Kanten eines ungerichteten Graphs bezüglich des kürzesten Pfads zwischen zwei ausgezeichneten Knoten verstehen wir eine kleine Kantenmenge, deren Löschung die Distanz zwischen den beiden Knoten signifikant vergrößert. Wir untersuchen das NP-vollständige Entscheidungsproblem, ob eine gewünschte Distanz durch die Löschung einer gegebenen Anzahl von Kanten erreicht werden kann, sowie drei Varianten, bei denen jeder Kante eine Länge, Löschungskosten oder beides zugeordnet ist. Wir konzentrieren uns dabei auf ausgewählte Graphklassen. Wir zeigen, dass das Problem auf allen Oberklassen der vollständigen Graphen NP-schwer bleibt, wenn den Kanten mindestens eines der beiden Gewichte zugeordnet ist. Für den Fall ohne Kantengewichte zeigen wir Lösbarkeit in Linearzeit auf vollständigen Graphen und Schwellwertgraphen und geben einen Polynomialzeitalgorithmus für die Einheits-Intervallgraphen an, dessen Korrektheit aus einer plausiblen Vermutung folgt. Wir untersuchen das Problem zudem auf Multigraphen, insbesondere auf den Seriell-Parallelen Graphen. Für letztere präsentieren wir zwei FPT-Algorithmen, die für drei der vier Varianten in Polynomialzeit laufen.

Contents

1 Introduction

Finding the shortest path between pairs of vertices in a graph is a well-studied computational problem with a vast amount of applications. These reach from map navigation or network design to more particular use cases. While many interesting variations of finding a shortest path are known to be polynomial-time computable, finding a set of „most vital“ vertices or edges with respect to shortest paths, that is, a minimum set of vertices or edges whose deletion increases the length of any shortest path by a certain amount, has been identified as a computationally hard problem [3, 4]. We study the particular problem Weighted Shortest Path Most Vital Edges defined as follows.

Weighted Shortest Path Most Vital Edges (W-SP-MVE)

Input: A septuple $(G, k, \ell, s, t, \sigma, \tau)$ where $G = (V, E)$ is a simple, undirected, connected graph with edge deletion costs $\sigma : E \to \mathbb{N}^+$, edge lengths $\tau : E \to \mathbb{N}^+$, terminal vertices $s, t \in V$, and thresholds $k, \ell \in \mathbb{N}$.

Question: Is there an edge subset $X \subseteq E$ with deletion cost $\sigma(X) \leqslant k$ such that the length of a shortest s-t-path in $G - X$ is at least ℓ?

The problem is NP-complete in general [3] as well as in the restricted case where both edge weights are uniform [4]. Corley and Sha [9] give a motivation in transport security similar to the VIP-Routing scenario analyzed by Omran et al. [21]; a defender wants to fortify a (small) number of roads that seem most relevant for an attacker to target in order to intercept the transport of an important person from one place to another. This scenario can be generalized to other notions of robust network design, where links with certain properties are deemed more likely to be subject to failure or interception or for which such an event is more likely to cause havoc. Different applications take the perspective of the attacker; among them hospital infection control [1] and the detection of smuggling [27].

In this paper we analyze a number of natural problem variants, defined in Section 2.3, on a handful of selected graph classes. The problem variants are formed by restricting some or all of the edge weights to be uniform, as depicted in Figure 1. See Section 2.3 for the formal definitions. As for graph classes, we analyze a number of superclasses of the class of complete graphs, being the complete graphs themselves (Section 3), threshold graphs (Section 4), split graphs (Section 5) and proper interval graphs (Section 6). In addition, we demonstrate basic techniques to handle multigraphs (Section 7) and look at the class of two-terminal series-parallel graphs (Section 8) in particular. See Section 1.2 for a summary of our results.

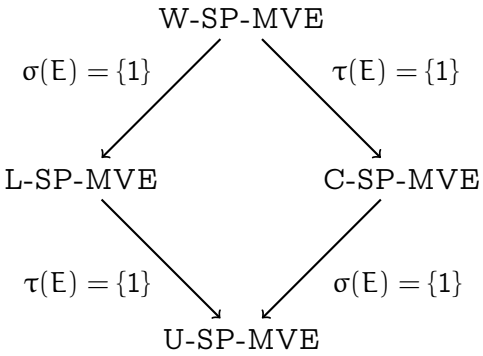

Figure 1: The decision problem variants examined in this work. Arrows point to restricted variants, arrow labels denote the restriction. Recall that σ denotes deletion cost and τ denotes length. Hence, L-SP-MVE and C-SP-MVE consider only edge lengths and edge costs, respectively. U-SP-MVE is the variant where edges are uniform.

1.1 Related work

The problem family defined above appears in the literature under a multitude of names. While the majority of authors adhere to the *most vital* superlative of earlier publications and append the terms *edges*, *arcs* or *links* [3, 4, 5, 9, 20], other contributors refer to analog definitions as *bounded cut* [2, 11, 15] or *edge interdiction* [16] problems. The variant where the edge deletion budget permits the removal of a single edge, that is $k = 1 = \sigma(e)$ for all $e \in E$, often has a distinct name prefixed with *single* [9, 19]. Usually the names admit a natural counterpart for the *node* (less frequent: *vertex*) deletion scenario [2, 4, 9, 15, 20]. Often no reference to shortest paths is given in the problem name. However, analog problems have been defined for the scenarios where vertices or edges are removed in order to increase the size of a minimum spanning tree or decrease the size of a maximum matching (e.g. [16, 24]). In the following, we give a quick survey that focuses on shortest paths and edge deletion. We write b short for the difference between the initial and the desired distance between the terminals, as formalized in Definition 19.

Corley and Sha [9] are the first to „define and motivate“ [4] the problem family of most vital vertices and edges in graphs, whereas formerly related problems have only been researched in the context of flow networks [8, 22]. They present an algorithm for finding the Single Most Vital Link, corresponding to L-SP-MVE on directed graphs for $k = 1$, which runs in $\mathcal{O}(n^4)$ time [3].

Ball et al. [3] investigate the Most Vital Arcs Problem and the Most Important Arcs Problem, which correspond to our MaxLen-W-SP-MVE and MinCost-W-SP-MVE optimization variants in the context of directed graphs, respectively. Their decision variant goes by the name of Vital Arcs Recognition Problem, which they show to be NP-complete. They also present an algorithm for L-SP-MVE on directed

graphs with $k = 1$ that runs in $\mathcal{O}(n^2)$ time, improving on the results by Corley and Sha [9]. They further claim that C-SP-MVE on directed graphs for $b = 1$ can be decided in polynomial time by finding a minimum cost cut, an idea used independently by Bazgan et al. [5] for the variant of L-SP-MVE.

Malik et al. [19] give an algorithm for finding the Single Most Vital Arc, corresponding to L-SP-MVE on directed graphs for $k = 1$, that has the same running time as Dijkstra's algorithm, that is $\mathcal{O}(m + n \log n)$ [14]. They give another algorithm for arbitrary k which Bar-Noy et al. [4] argue to be faulty by providing a counter-example. The latter also correct the proof of correctness for the first algorithm.

Bar-Noy et al. [4], in addition to the rectification above, show that U-SP-MVE on both directed and undirected graphs is NP-complete by reduction from Vertex Cover. They argue that the claim holds by analogy for the most vital vertices scenario.

Baier et al. [2] introduce an approximation perspective. Along with the problem family of Length-Bounded Edge Cut, which corresponds to our W-SP-MVE and restricted variants, they discuss the related problems Length-Bounded Node Cut and Length-Bounded Flow and the interplay between all three, in both directed and undirected graphs. They show inapproximability for MinCost-U-SP-MVE within a factor of 1.1377 for $\ell \in \{5, \ldots, n^{1-\epsilon}\}$ for all $\epsilon > 0$ and give a polynomial $(\ell - \mathrm{dist}(s, t))$-approximation for MinCost-W-SP-MVE, which implies that W-SP-MVE can be solved in polynomial time for $b = 1$. In addition, they show that W-SP-MVE is NP-complete on series-parallel and outerplanar graphs.

Golovach and Thilikos [15] focus on parameterized complexity and analyze both edge and vertex deletion scenarios in the contexts of both directed and undirected graphs as well as a related family of Bounded Disjoint Paths problems. They show that their Bounded Edge Undirected Cut (BEUC), which corresponds to our U-SP-MVE, is W[1]-hard with respect to k but FPT with respect to (k, ℓ).

Bazgan et al. [5] also focus on the perspective of parameterized complexity. They show that U-SP-MVE is NP-complete even for $b = 2$ on bipartite graphs with degeneracy 2 and diameter 8 and they rediscover that L-SP-MVE can be solved in polynomial time for $b = 1$, as shown previously by Baier et al. [2] for the more general W-SP-MVE. They elevate Baier et al.'s [2] result that U-SP-MVE is FPT with respect to (k, ℓ) to the more general problem variant of L-SP-MVE, based on the same observation. They further show that L-SP-MVE is FPT with respect to *feedback edge set number* and *cluster vertex deletion number* and XP with respect to the *maximum degree*.

Dvořák and Knop [11] show that U-SP-MVE is W[1]-hard when parameterized only by the pathwidth of the input graph but is FPT with respect to ℓ and the graph's treewidth as well as with respect to the graph's treedepth alone. They further introduce a multi-terminal problem variant for which they provide FPT results when the number of terminals is considered as the third parameter along with ℓ and the treewidth.

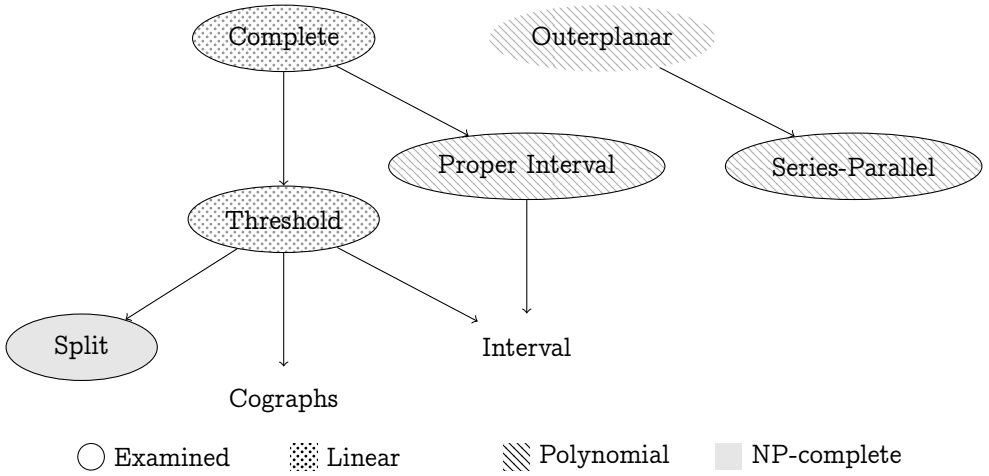

Figure 2: The graph classes examined in this document and a number of related landmark classes. Arrows point towards superclasses and can be read as „are contained in“. Time complexity results displayed are for U-SP-MVE.

1.2 Our results

We give either tractability or hardness results for all combinations of the decision problem variants discussed in Section 2.3 and the graph classes shown in Figure 2 that are marked with an outline. Table 1 gives a summary of all results.

First we show that, while U-SP-MVE is linear-time decidable on complete graphs, all three problem variants with non-uniform edge weights yield NP-hardness on all superclasses of complete graphs.

We then draw a line between two close relatives among these superclasses: We show that, while U-SP-MVE is linear-time decidable on threshold graph, the same problem variant is NP-complete on the class of split graphs, which contains the threshold graphs.

Next we conjecture that U-SP-MVE on proper interval graphs, another superclass of the complete graphs, is polynomial-time decidable. We support this claim with an algorithm whose correctness is subject to a plausible but unproven statement about the structure of optimum solutions in the context of this graph class.

Last, we look into multigraphs. We give two basic linear-time reductions that can be used to convert instances of multigraph-aware U-SP-MVE to equivalent instances of either U-SP-MVE or C-SP-MVE on ordinary graphs.

We close with the class of two-terminal series-parallel graphs as previously studied by Baier et al. [2] who have shown (weak) NP-hardness of W-SP-MVE on this class. We give two pseudo-polynomial algorithms that decide this problem variant in $\mathcal{O}(m^2k)$ and $\mathcal{O}(m^2\ell)$ time, respectively. This implies that the other three variants can be decided in polynomial time on this graph class, using one of the two algorithms.

Graph class	U-SP-MVE	C-SP-MVE	L-SP-MVE	W-SP-MVE
Complete	Linear (Theorem 3.3)	NP-complete (Theorem 3.2)	NP-complete (Theorem 3.1)	NP-complete ⟵
Threshold	Linear (Theorem 4.2)	NP-complete (Theorem 3.2)	NP-complete (Theorem 3.1)	NP-complete ⟵
Split	NP-complete (Theorem 5.1)	NP-complete (Theorem 3.2)	NP-complete (Theorem 3.1)	NP-complete ⟵
Proper Interval w.c.t.	$\mathcal{O}(n^5)$ (Theorem 6.6, subject to Conjecture 6.1)	NP-complete (Theorem 3.2)	NP-complete (Theorem 3.1)	NP-complete ⟵
Series-Parallel w.c.t.	$\mathcal{O}(m^3)$ (Corollary 8.5)	$\mathcal{O}(m^3)$ (Corollary 8.5)	$\mathcal{O}(m^3)$ (Corollary 8.5)	NP-complete (Baier et al. [2]), $\mathcal{O}(m^2k)$, $\mathcal{O}(m^2\ell)$ (Theorem 8.4)

Table 1: Time complexity results for the examined graph classes and all decision problem variants. Except for the mutually independent C-SP-MVE and L-SP-MVE, hardness results propagate towards the right-hand side. Propagation is marked by an arrow towards the source. „w.c.t." means „with certain terminals".

2 Preliminaries

In this section we introduce the notation used throughout the document. Section 2.1 introduces fundamental mathematical notation, Section 2.2 is dedicated to graph theoretical definitions and Section 2.3 contains the formal definitions of the problem variants that we analyze. In Section 2.4 we discuss observations on the most vital edges scenario that are not specific to any particular graph class.

2.1 Fundamentals

The following definitions clarify the meaning of mathematical objects and operations that can be found in the literature under a different notation or with varying semantics.

Definition 1 (Numbers)**.** We write $\mathbb{N}$ for the natural numbers including zero and $\mathbb{N}^+$ as a shorthand for $\mathbb{N} \setminus \{0\}$. The symbols $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$ denote the usual sets of integers, rationals, and real numbers.

Definition 2 (Power Set)**.** Given a set S we write $\mathcal{P}(S) := \{S' \subseteq S\}$ for the set of all subsets of S.

Definition 3 (Interval)**.** For $a, b \in \mathbb{R}$ we write $[a, b] := \{x \in \mathbb{R} \mid a \leqslant x \leqslant b\}$ for the closed real interval from a to b.

Definition 4 (Order)**.** An order is a tuple $(U, \prec)$ with a universe U and a binary relation $\prec : U \times U$ such that $\prec$ is anti-symmetric, transitive and either reflexive or anti-reflexive. If $\prec$ is anti-reflexive, then we call $(U, \prec)$ a *strict* order. If $\prec$ is total, then we call $(U, \prec)$ a *total* order, otherwise we call it a *partial* order. If not stated otherwise, an order in this document is a strict, total order.

Definition 5 (Predecessor and Successor)**.** Given a strict, total order $(U, \prec)$ we write $\text{pred}(x)$ and $\text{succ}(x)$ to denote the predecessor and successor of x, respectively.

Within the algorithms that we present, we sometimes use ∞ as a value that denotes a partial solution that cannot exist. When combining partial solutions, we allow limited arithmetic operations involving ∞ to reduce the number of necessary case distinctions.

Definition 6 (Addition of infinite values)**.** We define $\infty + \infty := \infty$.

2.2 Graph Theory

In this work we consider only undirected graphs. If not stated otherwise, graphs are simple graphs without loops or parallel edges. We use the well-known formalization given as follows.

Definition 7 (Graph)**.** A (*simple*) *graph* is a tuple $G = (V, E)$ such that V is a set of *vertices* and $E \subseteq \binom{V}{2}$ is a set of *edges*.

Under the name of multigraphs we understand graphs that allow both loops and parallel edges, that are multiple edges between any two vertices, defined as follows. We choose this representation as it allows us to formally treat multigraphs like normal graphs, most of the time.

Definition 8 (Multigraph)**.** Let V be a set of vertices and let E be a multiset of edges with $e \in \binom{V}{2} \cup \binom{V}{1}$ for all $e \in E$. If $|e| = 1$ holds for some $e \in E$, then we call e a *loop*, otherwise we call e an *ordinary edge*. We call a tuple $G = (V, E)$ a *multigraph* and we assign to each edge $e \in E$ a *multiplicity* $1_E : E \to \mathbb{N}$ such that $1_E(e)$ denotes the number of occurrences of e in E.

Convention 1 (Size of a graph)**.** Whenever a single graph or multigraph $G = (V, E)$ is given within context, we write n short for $|V|$ and m short for $|E|$.

In the following we define graph operations and properties. The edge complement set is the only graph property that is specific to simple graphs and is defined first.

Definition 9 (Edge Complement)**.** Let $G = (V, E)$ be a graph. We write $E^C := \binom{V}{2} \setminus E$ to denote the *edge complement* set of G.

In the remaining definitions of Section 2.2, $G = (V, E)$ denotes either a graph or a multigraph. In the latter case any subset of E is also a multiset.

Definition 10 (Neighborhood)**.** Let $v \in V$. We call $N(v) := \{u \in V \mid \{u, v\} \in E\} \setminus \{v\}$ the *open neighborhood* and $N[v] := N(v) \cup \{v\}$ the *closed neighborhood* of v.

Definition 11 (Twins)**.** Let $u, v \in V$. We call u and v *true twins* if $N[u] = N[v]$ and *false twins* if $N(u) = N(v)$.

Definition 12 (Induced Subgraph)**.** Let $V' \subseteq V$ and $E' \subseteq E$. We write

$$G[V'] := (V', \{\{u, v\} \in E \mid u, v \in V'\}) \text{ and}$$
$$G[E'] := (\{v \in V \mid \exists e \in E' \,.\, v \in e\}, E')$$

to denote the *induced subgraph* of G with respect to V' and E', respectively.

Definition 13 (Edge deletion)**.** Let $X \subseteq E$. We write $G - X$ short for $(V, E \setminus X)$.

Paths play a central role in this document. We define them as sequences of edges as opposed to sequences of vertices so that they behave appropriately in the context of multigraphs. However, we often prefer to refer to paths by the vertices that they visit. For example, we want to write $s \rightsquigarrow v \to t$ to refer to all paths that start at the vertex s, visit v after any number of edges (potentially zero, that is $s = v$) and then visit an edge $\{v, t\}$ to end at the vertex t. We consider only simple paths, that are those that do not contain a vertex twice. This notion of paths suffices our purpose as we ultimately argue about shortest paths, which are always simple as otherwise a loop or circle could be removed to obtain a shorter path, leading to a contradiction. In the following we define paths and give the semantics for the notation mentioned above.

Definition 14 (Path)**.** A (*simple*) *path* (*in* G) is a sequence of edges $P := (e_1, \ldots, e_p) \in E^p$ with $p \in \mathbb{N}^+$ such that for all $i, j \in \{1, \ldots, p\}$ with $i \neq j$ it holds that

- $|i - j| = 1 \Rightarrow |e_i \cap e_j| = 1$ and
- $|i - j| > 1 \Rightarrow e_i \cap e_j = \emptyset$.

We also refer to paths by giving a sequence of vertices. For all $s, t \in V$ with $s \neq t$, let

$$s \rightsquigarrow t := \big\{P := (e_1, \ldots, e_p) \in E^p \;\big|\; p \in \mathbb{N}^+ \wedge P \text{ is a path } \wedge s \in e_1 \wedge t \in e_p \wedge p > 1 \Rightarrow (s \notin e_2 \wedge t \notin e_{p-1})\big\} \text{ and}$$
$$s \to t := \big\{P \in s \rightsquigarrow t \;\big|\; |P| = 1\big\},$$

where $|P|$ denotes the length of the sequence P. For all $\mathcal{W} \subseteq s \rightsquigarrow v$ with $s, v \in V$ and $\mathcal{W} \neq \emptyset$ and for all $t \in V \setminus \{s, v\}$, let

$$\mathcal{W} \to t := \big\{P \in s \rightsquigarrow t \;\big|\; \exists P' \in \mathcal{W} \,.\, P' \cdot \{v, t\} = P\big\} \text{ and}$$
$$\mathcal{W} \rightsquigarrow t := \big\{P \in s \rightsquigarrow t \;\big|\; \exists P' \in \mathcal{W} \,.\, \exists P'' \in v \rightsquigarrow t \,.\, P' \cdot P'' = P\},$$

where $\cdot$ denotes concatenation of sequences. We further define $v \rightsquigarrow v := v$ for all $v \in V$. If $P \in s \rightsquigarrow t$ for $s, t \in V$ with $s \neq t$ then we call P an s-t-*path* (*in* G). We call s and t the *terminals* of an s-t-path. Furthermore, we say that P *is a path of the form* $\mathcal{W}$ (*in* G) if $P \in \mathcal{W}$ holds for a set of paths $\mathcal{W}$.

Definition 15 (Path Length)**.** If an edge length function $\tau : E \to \mathbb{N}^+$ is given, then the *length* of a path P in G is defined as $\sum_{i \in \{1,\dots,|P|\}} \tau(P_i)$. If no such function is given, then the length of P is $|P|$.

Definition 16 (Shortest Path)**.** Let P be an s-t-path in G. We call P a *shortest* s-t-*path* if there is no s-t-path P' in G such that the length of P' is smaller than the length of P.

Definition 17 (Distance)**.** Let $s, t \in V$ with $s \neq t$. Then $\mathrm{dist}_G(s, t)$ denotes the length of a shortest s-t-path in G or ∞ if no such path exists. We define $\mathrm{dist}_G(v, v) := 0$ for all $v \in V$. Additionally, let

$$\mathrm{dist}_G(v_1, \dots, v_p) := \sum_{1 \leqslant i \leqslant p-1} \mathrm{dist}_G(v_i, v_{i+1}).$$

Definition 18 (Cut)**.** Let $X \subseteq E$ and $s, t \in V$ with $s \neq t$. If $\mathrm{dist}_{G-X}(s, t) = \infty$, then we call X an s-t-*cut of size* $|X|$.

2.3 Problem variants

Recall that a W-SP-MVE instance is represented as a septuple $(G, k, \ell, s, t, \sigma, \tau)$, where σ denotes edge deletion cost and τ denotes edge length. We distinguish the following problem variants.

- L-SP-MVE: In the unit-cost case $\sigma(e) = 1$ for all $e \in E$ we accept a sextuple as input, where σ is omitted. We refer to this problem variant as Length-Weighted Shortest Path Most Vital Edges.
- C-SP-MVE: In the unit-length case $\tau(e) = 1$ for all $e \in E$ we also accept a sextuple as input, here τ is omitted. This problem variant is called Cost-Weighted Shortest Path Most Vital Edges.
- U-SP-MVE: If both $\sigma(e) = 1$ and $\tau(e) = 1$ hold for all $e \in E$ we accept a quintuple as input, where both σ and τ are omitted. This problem variant is referred to as Uniform Shortest Path Most Vital Edges.

Figure 1 on page 6 depicts the relationship among all variants. In the cases where $\sigma(e) = 1$ holds for all $e \in E$, note that $\sigma(X) = |X|$.

It is possible to turn a decision problem instance $\mathcal{I} = (G, k, \ell, s, t, \sigma, \tau)$ of W-SP-MVE into an optimization problem instance by either fixing ℓ and asking for the smallest k such that $\mathcal{I}$ is a *yes*-instance, which we refer to as MinCost-W-SP-MVE, or by fixing k

and asking for the biggest ℓ such that $\mathcal{I}$ is a *yes*-instance, referred to as MaxLen-W-SP-MVE. Both optimization variants can be extended to the other decision problem variants by analogy.

In this work we focus on the decision problem variants. However, as our tractability results have constructive proofs—often in the form of algorithms—it should be straightforward to construct algorithms that solve the associated optimization variants.

In the context of all problem variants, we often refer to the difference between the distance between s and t in the input graph and the desired distance ℓ. We introduce a shorthand for this difference.

Definition 19 (Length increase)**.** Given a W-SP-MVE instance $(G, k, \ell, s, t, \sigma, \tau)$, let $b := \max\{0, \ell - \mathrm{dist}_G(s, t)\}$ be the desired shortest path length increase.

Note that an algorithm deciding a problem instance would need to compute the value of b before it can be used as it is not part of the input. This can be done in $\mathcal{O}(m+n \log n)$ time on general graphs using Dijkstra's algorithm [14].

2.4 Basic observations

In this work we often claim that a certain variant of Shortest Path Most Vital Edges is NP-complete under a number of restrictions, such as when the input graph is restricted to a certain graph class. In the following we show that W-SP-MVE, and thus any restriction thereof, is contained in NP. To show NP-completeness of a restricted problem, it is then sufficient to show NP-hardness.

Theorem 2.1. W-SP-MVE *is contained in* NP*.*

Proof. Let $\mathcal{I} := (G, k, \ell, s, t, \sigma, \tau)$ be a problem instance of W-SP-MVE and let $X \subseteq E$ be a potential solution. We can verify whether X is a solution in polynomial time as follows. First, verify that $\sigma(X) \leqslant k$ holds. Next, find a shortest s-t-path P in $G - X$. This can be done in $\mathcal{O}(m + n \log n)$ time using Dijkstra's algorithm [14]. If no shortest s-t-path in $G - X$ exists, then $\mathrm{dist}_{G-X}(s, t) = \infty > \ell$ holds and thus X is a solution. Otherwise compare the length p of P with ℓ. It holds that X is a solution if and only if $p \geqslant \ell$. ■

The next observation justifies the restriction that the input graph of a problem instance needs to be connected; we show that this is merely a formal restriction that does not make the problem easier to decide.

Observation 2.2. *Let* $\mathcal{I} := (G, k, \ell, s, t, \sigma, \tau)$ *be a problem instance of* W-SP-MVE*. The restriction that* G *needs to be connected can be lifted without making the problem harder, assuming that linear time is the best achievable running time for any meaningful restriction of problem inputs.*

Proof. Let $\mathcal{I} := (G, k, \ell, s, t, \sigma, \tau)$ be a problem instance of a variant of W-SP-MVE where G is not necessarily connected. Then it is possible to find the connected component that contains s in linear time via breadth-first-search. If this component does not contain t, then the instance must be a *yes*-instance as $\text{dist}_G(s,t) = \infty > \ell$. If it does contain t, then the problem input can be reduced to the connected component that contains both s and t, as no vertex in a different connected component holds any relevance to the distance between s and t, given that deleting edges is the only graph modification of interest. This reduction can be computed in linear time, given that reachable vertices are marked during the breadth-first-search. ■

Note that, when we do assume that the input graph is connected, then we also know that any minimal solution that is not itself an s-t-cut leaves the entire graph connected. This is formalized by the next observation.

Observation 2.3. *Let $\mathcal{I} := (G, k, \ell, s, t, \sigma, \tau)$ be a yes-instance of* W-SP-MVE*. Let X be a minimal solution for $\mathcal{I}$ such that s and t are connected in $G - X$, that is $\text{dist}_{G-X}(s,t) \in \mathbb{N}$. Then it holds that $G - X$ is connected.*

Proof. Let $\mathcal{I}$ and X as above, that is $\mathcal{I} = (G = (V, E), k, \ell, s, t, \sigma, \tau)$ is a *yes*-instance of W-SP-MVE and X is a minimal solution for $\mathcal{I}$ such that s and t are connected in $G - X$. Recall that G is connected by definition of W-SP-MVE. Assume towards a contradiction that $G - X$ is not connected. Let $C_1 \subseteq V$ be the connected component in $G - X$ that contains s and t. Let $w \in V$ such that $w \notin C_1$. Next, let $P \in s \rightsquigarrow u \to v \rightsquigarrow w$ be a shortest path from s to w in G such that $u \in C_1$ and $v \in C_2$ for a connected component $C_2 \subseteq V$ in $G - X$ with $C_2 \neq C_1$. It holds that $\{u, v\} \in X$ since otherwise C_1 and C_2 would be connected in $G - X$, implying $C_1 = C_2$. Let $X' := X \setminus \{\{u, v\}\}$. Assume towards a contradiction that there is an s-t-path P' in $G - X'$ that contains v. It holds that every s-v-path as well as every v-t-path in $G - X'$ contains $\{u, v\}$ as it is the only edge connecting C_1 and C_2. Since a path cannot cannot contain an edge twice, this contradicts the existence of P'. Hence, every (shortest) s-t-path in $G - X'$ is also a path in $G - X$. Since X is a solution, it follows that X' is a solution. Since $X' \subset X$ this contradicts that X is minimal. ■

3 Complete Graphs

We start our analysis with the fundamental class of complete graphs. While our expectations are matched in the case of U-SP-MVE, which is easy to decide on this graph class as shown in Section 3.2, it may be more surprising that all other decision problem variants discussed in this document remain NP-hard even when restricted to complete graphs. The class is defined as follows.

Definition 20 (Complete Graph)**.** A graph $G = (V, E)$ is a *complete graph* if and only if $E = \binom{V}{2}$.

3.1 Hardness results

Instead of proving hardness for complete graphs specifically, we show two stronger claims that pass over to other graph classes as well: Both L-SP-MVE and C-SP-MVE are NP-complete on all graph classes that *contain* the class of complete graphs.

Theorem 3.1. L-SP-MVE *is* NP*-complete on superclasses of the complete graphs.*

The claim can be shown through a reduction from L-SP-MVE on general graphs, which is NP-complete as a generalization of the NP-complete U-SP-MVE [4]. The reduction „fills in" all edges necessary to turn the general graph into a complete graph, assigning to each of the new edges an edge length that is greater than ℓ. These edges are not relevant in the context of the most vital edges scenario as no shortest s-t-path of length at most ℓ contains them, so the reduced problem instance is equivalent to the original one.

Proof. Let $\mathcal{C}$ be a graph class that contains the class of complete graphs and let $\mathcal{I} := (G = (V, E), k, \ell, s, t, \tau)$ be an L-SP-MVE problem instance. We construct an equivalent instance $\mathcal{I}' := (G' = (V', E'), k, \ell, s, t, \tau')$ of L-SP-MVE with $G' \in \mathcal{C}$ as follows. Let $V' := V$, $E' := \binom{V}{2}$ and $\tau' : E' \to \mathbb{N}^+$ with

$$\tau'(e) := \begin{cases} \ell + 1, & e \in E^C, \\ \tau(e), & \text{otherwise.} \end{cases}$$

It holds that $\mathcal{I}'$ is a L-SP-MVE problem instance and can be computed in polynomial time given $\mathcal{I}$. Furthermore, it holds that $G' \in \mathcal{C}$ as G' is a complete graph and $\mathcal{C}$ contains the complete graphs.

We show that $\mathcal{I}$ is a *yes*-instance if and only if $\mathcal{I}'$ is a *yes*-instance.

Forward implication: If $\mathcal{I}$ is a yes-instance, then $\mathcal{I}'$ is a yes-instance. An edge of length $\ell + 1$ cannot be part of an s-t-path of length at most ℓ, hence an edge deletion set X with $|X| \leqslant k$ that destroys all s-t-paths of length at most ℓ in G also does so in G'. Since $\mathcal{I}$ is a *yes*-instance such a set exists and hence $\mathcal{I}'$ is a *yes*-instance. □

Reverse implication: If $\mathcal{I}'$ is a yes-instance, then $\mathcal{I}$ is a yes-instance. Since $\mathcal{I}'$ is a *yes*-instance, k edge deletions are sufficient to destroy all s-t-paths of length at most ℓ in G'. Since $V = V'$ and $E \subseteq E'$, this amount of edge deletions is also sufficient to destroy all s-t-paths of length at most ℓ in G. ■

An analogous claim can be made for the problem variant C-SP-MVE.

Theorem 3.2. C-SP-MVE *is* NP*-complete on superclasses of the complete graphs.*

The claim can be shown through a reduction from U-SP-MVE on general graphs, which is known to be NP-complete [4]. Again the reduction „fills in" all edges necessary

to turn the general graph into a complete graph. This time the edge deletion cost and the budget are set such that all newly introduced edges can be deleted without influencing the subsets of all other edges that the new budget affords to delete. The new deletion costs for the edges that existed in the original graph are still uniform, so that the same number and thus the same subsets of them can be deleted. This means that the new instance is equivalent to the original one.

Proof. Let $\mathcal{C}$ be a graph class that contains the class of complete graphs and let $\mathcal{I} := (G = (V, E), k, \ell, s, t)$ be an U-SP-MVE problem instance. We construct an equivalent instance $\mathcal{I}' := (G' = (V', E'), k', \ell, s, t, \sigma)$ of C-SP-MVE with $G' \in \mathcal{C}$ as follows. Let $V' := V$ and $E' := \binom{V}{2}$. Let $k' := kn^2 + n(n-1)$ and $\sigma : E' \to \mathbb{N}^+$ with

$$\sigma(e) := \begin{cases} 1, & e \in E^C, \\ n^2, & \text{otherwise.} \end{cases}$$

It holds that $\mathcal{I}'$ is a C-SP-MVE problem instance and can be computed in polynomial time given $\mathcal{I}$. Furthermore, $G' \in \mathcal{C}$ holds as G' is a complete graph and $\mathcal{C}$ contains the complete graphs.

We show that $\mathcal{I}$ is a *yes*-instance if and only if $\mathcal{I}'$ is a *yes*-instance.

Forward implication: If $\mathcal{I}$ is a yes-instance, then $\mathcal{I}'$ is a yes-instance. Let $\mathcal{I}$ be a *yes*-instance and let $X \subseteq E$ be a solution for $\mathcal{I}$, that is $\text{dist}_{G-X}(s,t) \geqslant \ell$ and $|X| \leqslant k$. We construct a solution $X' \subseteq E'$ for $\mathcal{I}'$ as follows. Let $X' := \{e \in E' \mid e \in X \vee e \in E^C\} = X \cup E^C$. First we show that $\sigma(X') \leqslant k'$. From $X \subseteq E$ it follows that $X \cap E^C = \emptyset$ and thus

$$\sigma(X') = \sigma(X) + \sigma(E^C) = n^2|X| + 1|E^C| \leqslant kn^2 + n(n-1) = k'.$$

Next we show that $\text{dist}_{G'-X'}(s,t) \geqslant \ell$. Recall that $G' = (V, E \cup E^C)$ and $X' = X \cup E^C$ hold. Hence we have

$$G' - X' = G' - (X \cup E^C) = (G' - E^C) - X = G - X$$

and $\text{dist}_{G'-X'}(s,t) = \text{dist}_{G-X}(s,t) \leqslant \ell$ follows. □

Reverse implication: If $\mathcal{I}'$ is a yes-instance, then $\mathcal{I}$ is a yes-instance. Let $\mathcal{I}'$ be a *yes*-instance and let $X' \subseteq E'$ be a solution for $\mathcal{I}'$, that is $\text{dist}_{G'-X'}(s,t) \geqslant \ell$ and $\sigma(X') \leqslant k'$. We construct a solution $X \subseteq E$ for $\mathcal{I}$ as follows. Let $X := \{e \in E \mid e \in X' \wedge e \notin E^C\} = X' \setminus E^C$. First we show that $|X| \leqslant k$. Recall that $k' = kn^2 + n(n-1)$. As $\sigma(e) = n^2 > n(n-1)$ holds for all $e \in E' \setminus E^C$, we know that the $n(n-1)$ part of the budget k' could only be spent on edges in E^C. As $\sigma(E^C) = 1|E^C| \leqslant n(n-1)$ holds, we can assume without loss of generality that $E^C \subseteq X'$. Hence, $\sigma(X) = \sigma(X' \setminus E^C) \leqslant kn^2$ holds. From $\sigma(e) = n^2$ for all $e \in E' \setminus E^C$ and $X = X' \setminus E^C \subseteq E' \setminus E^C$ it follows that $\sigma(e) = n^2$ for all $e \in X$ and thus

$$|X| = \frac{\sigma(X)}{n^2} \leqslant \frac{kn^2}{n^2} = k.$$

Next we show that $\mathrm{dist}_{G-X}(s,t) \geqslant \ell$. From $G' = (V, E \cup E^C)$ and $G = (V, E)$ it follows that $G = G' - E^C$. Hence we have

$$G - X = (G' - E^C) - (X' \setminus E^C) = G' - (E^C \cup X') \overset{E^C \subseteq X'}{=} G' - X'$$

and $\mathrm{dist}_{G-X}(s,t) = \mathrm{dist}_{G'-X'}(s,t) \leqslant \ell$ follows. ■

Note that these hardness results hold in the general case where edge weights, that are the function values of σ and τ, are not upper-bounded. It is easy to see that the results also apply if the edge weights as well as the values of all other numeric parameters of the input instance are polynomially upper-bounded in the input size, a property known as *strong* NP-hardness. It remains open, however, whether efficient algorithms that decide L-SP-MVE and C-SP-MVE on the class of complete graphs and its superclasses exist when there is an even stricter upper bound on the function values of σ and τ, that is in the case where $\sigma(e) \leqslant q$ or $\tau(e) \leqslant q$ holds for all $e \in E$ and some $q \in \mathbb{N}^+$ with $q \leqslant \ell$.

3.2 Tractability results

When we consider complete graphs with unit-length edges, our main observation is that in a highly connected graph structure like this, the number of edge deletions necessary for increasing the length of a shortest path to a certain length, in this case to a length of at least three, is also sufficient to produce an s-t-cut. Once we find an s-t-cut in a graph G, it holds that the corresponding edge deletion set X solves any U-SP-MVE instance on G with a budget k that is at least as big as the size of X, regardless of the desired shortest path length ℓ. In the case of complete graphs, we show that an optimum solution for $\ell = 3$ is at least as big as a minimum s-t-cut, which is easy to find. Hence, the case of $\ell \geqslant 4$ can be reduced to the case of $\ell = 3$; in both cases it is sufficient to look for a minimum s-t-cut. The remaining case of $\ell \leqslant 2$ boils down to the question whether our budget supports the deletion of the single edge connecting s and t. These observations allow for an efficient algorithm that decides U-SP-MVE on complete graphs.

Theorem 3.3. U-SP-MVE *on complete graphs is linear-time solvable.*

Proof. Let $\mathcal{I} := (G, k, \ell, s, t)$ be a problem instance of U-SP-MVE such that $G = (V, E)$ is a complete graph, that is $E = \binom{V}{2}$. We look at the values of n, b, and k to decide $\mathcal{I}$.

Decision. First we rule out trivial instances. If $b = 0$, then we have that $\mathcal{I}$ is a *yes*-instance. If $b \neq 0$ and $n = 2$, then we have that $\mathcal{I}$ is a *yes*-instance if and only if $k \geqslant 1$ as $\mathrm{dist}_{G-\{\{s,t\}\}}(s,t) = \infty > \ell$ and $|\{\{s,t\}\}| = 1$. Last, if $b \neq 0$, $n \neq 2$ and $k = 0$, then we have that $\mathcal{I}$ is a *no*-instance. Hence, we assume in the following that $b \geqslant 1$, $k \geqslant 1$, and $n \geqslant 3$. Under these assumptions, the case of $b = 1$ also becomes trivial: If $b = 1$, then it holds that $X := \{\{s,t\}\}$ is a solution with $\mathrm{dist}_{G-X}(s,t) = 2 = \ell$ and $\mathcal{I}$ is a *yes*-instance. Hence, we also assume in the following that $b \geqslant 2$.

Next, let $X \subseteq E$ be minimum such that $\text{dist}_{G-X}(s,t) \geqslant \ell$. Since $b \geqslant 2$ it holds that $\{s,t\} \in X$. Since for all $v \in V \setminus \{s,t\}$ we have that $s \to v \to t$ is an s-t-path of length 2 in G, it follows that at least one of the edges $\{s,v\}$ and $\{v,t\}$ is contained in X. Hence $n-1$ is a lower bound for $|X|$, formally $|X| \geqslant 1 + |V \setminus \{s,t\}| = n-1$. Let $X' := \{\{s,v\} \in E \mid v \in N[s]\}$. It holds that $\text{dist}_{G-X'}(s,t) = \infty \geqslant \ell$ and $|X'| = n-1$. As X is chosen minimum such that $\text{dist}_{G-X}(s,t) \geqslant \ell$, we can conclude that also $|X| = n-1$. It follows that $\mathcal{I}$ is a *yes*-instance if and only if $k \geqslant n-1$.

Running time. The case distinction above can be made in constant time given n, b, and k, with the latter being an input parameter. Since $\text{dist}_G(s,t) = 1$, it holds that $b = \ell - 1$ can be computed in constant time given the input parameter ℓ. It holds that n can be computed in linear time by reading the input and all input parameters can be retrieved in the process of computing n. Thus, we obtain linear running time altogether. ■

From observations made in the proof of Theorem 3.3, we can draw a number of conclusions with respect to cliques within a general graph.

Corollary 3.4. *Let $G = (V,E)$ be a graph. Let $C \subseteq V$ be a clique in G with $v_1, v_2 \in C$ and $v_1 \neq v_2$. The following holds.*

- *Deleting a single edge from E is sufficient to increase the distance between v_1 and v_2 to at least 2 (by at least 1).*
- *The distance between v_1 and v_2 can be increased to at most 2 (by at most 1) by deleting less than $|C|-1$ edges from E.*
- *By deleting $|C|-1$ edges from E, it can be achieved that any remaining v_1-v_2-path uses a vertex $v \in V$ with $v \notin C$.*

4 Threshold Graphs

The term threshold graph was first coined by Chvátal and Hammer [7] in the context of integer linear programming problems. However, Mahadev and Peled [18] survey that they have been „discovered independently [by] people working in different areas“ as they „play an important role in graph theory as well as in several applied areas.“ As a result there are plenty of equivalent definitions for the class of threshold graphs. We give only one that we deem particularly rewarding with regard to the Shortest Path Most Vital Edges scenario which is based on the Dilworth number of a graph. The formal definitions in this section closely resemble those given by Brandstädt et al. [6]; for alternative definitions of the threshold graphs refer to both Brandstädt et al. [6] and Mahadev and Peled [18].

To introduce the Dilworth number, we first need a notion of comparability of vertices. In short, two vertices are comparable if the open neighborhood of one vertex is a subset of the closed neighborhood of the other vertex.

Definition 21 (Domination and Comparability)**.** Let $G = (V, E)$ be a graph with $u, v \in V$. We say that u *dominates* v if $N(v) \subseteq N[u]$. If u dominates v or v dominates u, we call u and v *comparable*, otherwise we call them *incomparable*.

The dilworth number of a graph describes the size of the greatest subset of vertices which are pairwise incomparable. It has its roots in order theory; if the graph models a strict partial order $(U, \prec)$ such that an edge represents two elements of U that are related by $\prec$, then such a graph is called a *comparability graph* [6] and its dilworth number corresponds to the size of the biggest *antichain* of the order, which denotes a subset of U where no two elements are related in $\prec$.

Definition 22 (Dilworth number)**.** Let $G = (V, E)$ be a graph. The *Dilworth number* of a graph is defined as the size of the biggest subset of V that contains only pairwise incomparable vertices, that is

$$\operatorname{dilw}(G) := \max\left\{|V'| \;\middle|\; V' \subseteq V \wedge \forall v_1, v_2 \in V' \,.\, v_1 \neq v_2 \Rightarrow v_1, v_2 \text{ incomparable}\right\}.$$

The threshold graphs can now be defined as the class of graphs where every pair of vertices is comparable. In terms of order theory, threshold graphs are the comparability graphs of threshold orders [6]. Given a threshold value, a threshold order is a strict partial order $(U, \prec)$ where each element of U is assigned a weight and two elements are related in $\prec$ if and only if the sum of their weights reaches the threshold [6].

Definition 23 (Threshold Graph)**.** A graph G is a *threshold graph* if and only if $\operatorname{dilw}(G) = 1$.

From this definition of threshold graphs we can obtain directly the following lemma, which is central to the proof of the succeeding tractability claim.

Lemma 4.1. *Let $G = (V, E)$ be a threshold graph. Let $v_1, v_2 \in V$. Then*

$$N(v_1) \setminus \{v_2\} \subseteq N(v_2) \setminus \{v_1\}$$

or

$$N(v_2) \setminus \{v_1\} \subseteq N(v_1) \setminus \{v_2\}$$

holds.

Proof. Let $G = (V, E)$ be a threshold graph, that is $\operatorname{dilw}(G) = 1$. Then there is no set of two or more vertices in V that are pairwise incomparable. Hence, every pair of vertices in G is comparable. Let v_1, v_2 in V. As v_1 and v_2 are comparable, we have

$N(v_1) \subseteq N[v_2]$ or $N(v_2) \subseteq N[v_1]$. Assume without loss of generality the former, that is $N(v_1) \subseteq N[v_2]$. It follows that

$$\begin{aligned} & N(v_1) \subseteq N[v_2] \\ \Rightarrow\ & N(v_1) \setminus \{v_1, v_2\} \subseteq N[v_2] \setminus \{v_1, v_2\} \\ \Rightarrow\ & N(v_1) \setminus \{v_2\} \subseteq N(v_2) \setminus \{v_1\}. \end{aligned}$$

■

With this property of threshold graphs at hand, we can show the following claim.

Theorem 4.2. U-SP-MVE *is linear-time solvable on threshold graphs.*

Proof. Let $\mathcal{I} := (G, k, \ell, s, t)$ be a problem instance of U-SP-MVE such that $G = (V, E)$ is a threshold graph. By Lemma 4.1, $N(s) \setminus \{t\} \subseteq N(t) \setminus \{s\}$ or $N(t) \setminus \{s\} \subseteq N(s) \setminus \{t\}$ holds. Assume without loss of generality the former, that is $N(s) \setminus \{t\} \subseteq N(t) \setminus \{s\}$.

Decision. We consider different values of ℓ. Deciding $\mathcal{I}$ in the case of $\ell \leqslant 2$ is trivial. In the case of $\ell \geqslant 3$ we need to delete at least as many edges as there are edge-disjoint s-t-paths of length at most 2 in G. As $N(s) \setminus \{t\} \subseteq N(t) \setminus \{s\}$ holds, we have that every $v \in N(s) \setminus \{t\}$ one-to-one corresponds to an s-t-path of length 2 in G of the form $s \to v \to t$. It is easy to see that these paths are pairwise edge-disjoint. It follows that in the case of $t \notin N(s)$, there are $|N(s)|$ edge-disjoint s-t-paths of length 2 in G. If $t \in N(s)$, then there are $|N(s)| - 1$ edge-disjoint s-t-paths of length 2 as well as one s-t-path of length 1, being the unique path of the form $s \to t$. In either cases we have that there are $|N(s)|$ edge-disjoint s-t-paths of length at most 2 in G. Hence, we need a budget of $k \geqslant |N(s)|$ to destroy each of them. It holds that such k is sufficient to produce an s-t-cut. An optimum solution to $\mathcal{I}$ that is also an s-t-cut is $X := \{(v, s) \in E \mid v \in N(s)\}$.

Running time. The trivial cases of $\ell \leqslant 2$ can be decided in linear time by reading the input to see if $\{s, t\} \in E$ and to retrieve the value of ℓ. In the case of $\ell \geqslant 3$ we need to determine the degrees of s and t to decide which of $N[s] \setminus \{s, t\} \subseteq N[t] \setminus \{s, t\}$ and $N[t] \setminus \{s, t\} \subseteq N[s] \setminus \{s, t\}$ holds. This is done in $\mathcal{O}(E)$ time by counting the occurrences of s and t within E. Finding the minimum of both degrees and comparing it to k yields a decision for $\mathcal{I}$ in constant time. ■

As the threshold graphs are a superclass of the complete graphs, which follows directly from the fact that complete graphs have dilworth number 1 as every pair of vertices in a complete graph has the same closed neighborhood, we can refer to the hardness results of Section 3 for all other problem variants.

5 Split Graphs

The next graph class we look at is the class of split graphs. A split graph is a graph whose vertices can be partitioned into two sets C and S such that $G[C]$ is a complete graph and $G[S]$ is an independent set. While this is arguably a structural property of considerable weight, our results show that U-SP-MVE remains NP-hard on split graphs.

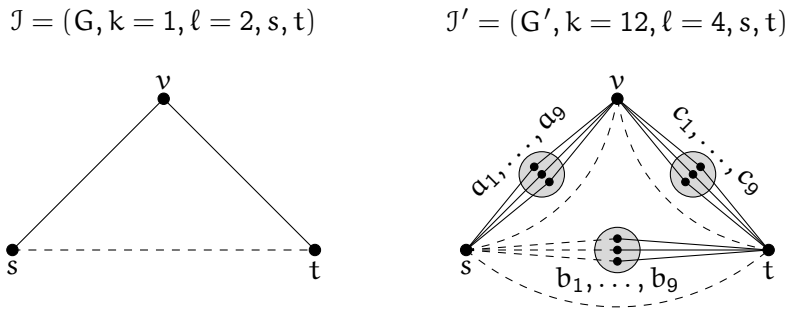

Figure 3: An example reduction from a general graph to a split graph. Some vertices are omitted for the sake of readability; the encircled groups of three vertices represent groups of nine vertices. Dashed lines represent solutions of the associated U-SP-MVE instance.

Theorem 5.1. U-SP-MVE *is* NP*-complete on split graphs.*

We show the claim by reduction from U-SP-MVE on general graphs, based on the following idea. First we apply edge subdivision and vertex cloning to create an edge gadget for every edge in the original graph. In the context of s-t-paths, these edge gadgets behave like an edge of length two with a deletion cost of n^2. We reflect this by adjusting the edge deletion budget and the desired shortest path length accordingly. Last we add regular edges between all vertices in the original graph and increase the edge deletion budget such that deleting all these additional edges is affordable but the budget cannot be spent reasonably on any of the edge gadgets. With this we ensure that the resulting graph is a split graph. Figure 3 shows an example reduction.

Proof. We reduce the NP-complete U-SP-MVE on general graphs [4] to U-SP-MVE on split graphs.

Construction. Let $\mathcal{I} := (G = (V, E), k, \ell, s, t)$ be a problem instance of U-SP-MVE. We reduce $\mathcal{I}$ to an instance $\mathcal{I}' := (G' = (V', E'), k', \ell', s, t)$ on split graphs as follows. Initially, let V' and E' be empty. First, we add all vertices in V to V' and we add all edges in E to E'. Next, we subdivide every edge in E' and for each vertex in V' obtained through subdivision, we add $n^2 - 1$ false twins of it to V'. Then, for every two vertices in V we add an edge to E' connecting the corresponding two vertices in V'. Last, we set $k' := \binom{n}{2} + kn^2$ and $\ell' = 2\ell$. In summary, we obtain

$$
\begin{aligned}
V' &:= V \cup \{\sigma^i_{\{u,v\}} \mid \{u, v\} \in E \wedge i \in \{1, \ldots, n^2\}\},\\
E' &:= \binom{V}{2} \cup \{\{u, \sigma^i_{\{u,v\}}\}, \{v, \sigma^i_{\{u,v\}}\} \mid \{u, v\} \in E \wedge i \in \{1, \ldots, n^2\}\},\\
k' &:= \binom{n}{2} + kn^2, \text{ and}\\
\ell' &:= 2\ell
\end{aligned}
$$

with $\sigma^i_{\{u,v\}} \notin V$ for all $u, v \in V$ and $i \in \mathbb{N}$. Clearly, $\mathcal{I}'$ can be computed in polynomial time given $\mathcal{I}$. Moreover, it holds that the values of the numeric parameters of $\mathcal{I}'$ are polynomially upper-bounded in the input length and the values of the numeric parameters of $\mathcal{I}$.

Correctness. We show that G' is a split graph and that $\mathcal{I}$ is a *yes*-instance if and only if $\mathcal{I}'$ is a *yes*-instance.

First we show that G' is a split graph. It holds that $V \subseteq V'$ and $\binom{V}{2} \subseteq E'$. Hence, $C := V$ induces a clique in G'. Recall that every vertex in $S := V' \setminus V$ has exactly two neighbors that are endpoints of an edge in E. Hence, $N(\sigma) \subseteq C$ for all $\sigma \in S$ and thus G' is a split graph.

Last we show that $\mathcal{I}$ is a *yes*-instance if and only if $\mathcal{I}'$ is a *yes*-instance.

Forward implication: If $\mathcal{I}$ is a yes-instance, then $\mathcal{I}'$ is a yes-instance. Let $\mathcal{I}$ be a *yes*-instance and let $X \subseteq E$ be a solution for $\mathcal{I}$, that is $\mathrm{dist}_{G-X}(s,t) \geqslant \ell$. Construct a solution X' for $\mathcal{I}'$ as follows. For every $\{u,v\} \in X$ and $i \in \{1,\dots,n^2\}$, add one of the edges incident to $\sigma^i_{\{u,v\}}$ to X'. At this point, it holds that $|X'| = |X|n^2 \leqslant kn^2$. Next, add $\binom{C}{2} = \binom{V}{2}$ to X'. It holds that $|X'| \leqslant \binom{n}{2} + kn^2 = k'$. Let P' be a shortest s-t-path in $G' - X'$. Since both $(G-X)[C]$ and $(G-X)[S] = G[S]$ contain only isolated vertices, it holds that P' alternates between vertices in C and S. By construction of G' and X', it holds that the biggest subsequence of vertices in P' that contains only vertices in C describes a path P in $G-X$ and this path has size $|P| = \frac{1}{2}|P'|$. Since $\mathcal{I}$ is a *yes*-instance, we have that $|P| \geqslant \ell$ and thus $|P'| \geqslant 2\ell$. Hence, X' is a solution and $\mathcal{I}'$ is a *yes*-instance. □

Reverse implication: If $\mathcal{I}'$ is a yes-instance, then $\mathcal{I}$ is a yes-instance. Let $\mathcal{I}'$ be a *yes*-instance and let $X' \subseteq E'$ be a minimal solution for $\mathcal{I}'$. As X' is minimal, it holds that if $\{u, \sigma^i_{\{u,v\}}\} \in X'$ for some $u, v \in C$ and $i \in \{1,\dots,n^2\}$, then either $\{u, \sigma_{(\{u,v\},j)}\} \in X'$ or $\{v, \sigma_{(\{u,v\},j)}\} \in X'$ holds for all $j \in \{1,\dots,n^2\}$, since otherwise $X' \setminus \{\{u, \sigma^i_{\{u,v\}}\}\}$ would be a solution. This leads to the following observation.

$$\forall u, v \in C \,.\, |\{e \in X' \mid \exists i \in \{1,\dots,n^2\} \,.\, \sigma^i_{\{u,v\}} \in e\}| \in \{0, n^2\} \tag{1}$$

Construct a solution X for $\mathcal{I}$ as follows. For every $e \in X'$ with $\sigma^i_{\{u,v\}} \in e$ for some $u, v \in C$ and $i \in \{1,\dots,n^2\}$, add $\{u,v\}$ to X. Recall that $|X'| \leqslant \binom{n}{2} + kn^2$. Since $\binom{n}{2} < n^2$ holds for $n \in \mathbb{N}$, we obtain $|X'| < (k+1)n^2$. As of Equation (1), we know that for every $\{u,v\} \in X$, there are exactly n^2 corresponding edges of the form $\{w, \sigma^i_{\{u,v\}}\}$ with $w \in \{u,v\}$ and $i \in \{1,\dots,n^2\}$ in X'. Hence, we obtain

$$|X| \leqslant \frac{|X'|}{n^2} < \frac{(k+1)n^2}{n^2} = k+1$$

and thus $|X| \leqslant k$ holds. Let P be a shortest s-t-path in $G-X$. By construction of G' we have that every vertex contained in P is also contained in $C \subseteq V'$. Furthermore, for every two vertices $u, v \in C$ there is a $\sigma^i_{\{u,v\}} \in S$ with $i \in \{1,\dots,n^2\}$ such that $u \to \sigma^i_{\{u,v\}} \to v$

is a path in G'. Hence, there is a s-t-path P' in G' that alternates between vertices in P and in S and has size $|P'| = 2|P|$. Assume towards a contradiction that P' is not a path in $G' - X'$. Then there are $u, v \in C$ and $\sigma^i_{\{u,v\}} \in S$ with $i \in \{1, \ldots, n^2\}$ such that $\{w, \sigma\} \in X'$ for some $w \in \{u, v\}$. By construction of X it follows that $\{u, v\} \in X$. This contradicts that P is a path in $G - X$. Hence, P' is a path in $G' - X'$. Since X' is a solution, it follows that

$$|P'| \geqslant \ell' \Leftrightarrow 2|P| \geqslant 2\ell \Leftrightarrow |P| \geqslant \ell.$$

Since P is a shortest s-t-path in $G - X$, we can conclude that X is a solution and $\mathcal{I}$ is a *yes*-instance. ■

6 Proper Interval Graphs

In this section we look at another superclass of the complete graphs which is well known in the context of resource allocation. Interval graphs model the intersection of intervals on a line. If the line represents a time line then an interval, modeled by a vertex of the interval graph, could correspond to a time frame in which exclusive access to a resource is requested by an agent. In this context overlapping edges represent conflicting requests, which are modeled by an edge in the graph. We look at the restricted class of proper interval graphs, where the intervals being modeled do not properly contain each other. Equivalent definitions go under the names of unit interval graphs, where the intervals being modeled are required to have uniform length, and indifference graphs, which model points on a number line as vertices that share an edge if the points are within certain distance to another [23].

An important structural property that proper interval graphs possess in contrast to the general interval graphs is a vertex order corresponding to the start point (or end point) of the intervals on the line. For a given graph the order is unique up to true twins, which correspond to intervals with the same intersection pattern [12]. Such an order can be found in linear time along with recognizing that a given graph is a proper interval graph [10]. In this section we conjecture the existence of a minimum solution to any U-SP-MVE *yes*-instance on proper interval graphs with certain terminals, where, after edge deletion, the distances of vertices measured from the first vertex in the order increases monotonically with respect to the vertices' index in the order. Indeed, the same statement does hold with respect to distances in the original graph, as discussed in Observation 6.2. If the conjecture holds, it is possible to decide according instances in polynomial time, as shown in Section 6.2. We first give a formal definition of the proper interval graphs.

Definition 24 (Proper Interval Model)**.** Let $\mathcal{M} \subseteq \{[x, y] \mid x < y\}$ be a finite set such that for all $I \in \mathcal{M}$ there is no $I' \in \mathcal{M}$ with $I' \subset I$. Then we call $\mathcal{M}$ a *proper interval model*.

Definition 25 (Proper Interval Graph)**.** A graph $G = (V, E)$ is a *proper interval graph* if it can be constructed from a proper interval model $\mathcal{M}$ as follows. Initially let $V := \mathcal{M}$ and let $E := \left\{\{v_1, v_2\} \in \binom{V}{2} \mid v_1 \cap v_2 \neq \emptyset\right\}$. Last rename vertices in G as required.

Definition 26 (Fitting Interval Model)**.** If a proper interval graph $G = (V, E)$ can be constructed from an interval model $\mathcal{M}$ as described in Definition 25, then we call $\mathcal{M}$ a *fitting interval model* (*of* G). We say that a vertex $v \in V$ and an interval $I \in \mathcal{M}$ *correspond* to each other if, in the renaming step of Definition 25, I was renamed to v.

Next we define the vertex order mentioned above.

Definition 27 (Interval Order)**.** Let $G = (V, E)$ be a proper interval graph and let $\mathcal{M}$ be a fitting interval model. Let $I_v := [\triangleright_v, \triangleleft_v] \in \mathcal{M}$ be the interval corresponding to v for all $v \in V$. Then a strict, total order $(V, \prec)$ such that $u \prec v \Rightarrow \triangleright_u \leqslant \triangleright_v$ holds for all $u, v \in V$ is called an *interval order* of G.

Definition 28 (Order-Fitting Interval Model)**.** Let $G = (V, E)$ be a proper interval graph and let $(V, \prec)$ be a fitting interval order. Then by Definition 27 there is a fitting interval model $\mathcal{M}$ of G such that $I_v := [\triangleright_v, \triangleleft_v] \in \mathcal{M}$ is the interval corresponding to v for all $v \in V$ and $u \prec v \Rightarrow \triangleright_u \leqslant \triangleright_v$ holds for all $u, v \in V$. We call such a proper interval model an *order-fitting interval model* (*of* G *and* $\prec$).

Last we define the central conjecture. We discuss it in the following.

Conjecture 6.1. *Let* $G = (V, E)$ *be a proper interval graph with an interval order* $(V, \prec)$ *such that* $s \in V$ *is minimum with respect to* $\prec$ *and* $t \in V$ *is maximum with respect to* $\prec$*. Let* $\mathcal{I} = (G, k, \ell, s, t)$ *be a yes-instance of* U-SP-MVE*. Then there is a minimum solution* $X \subseteq E$ *of* $\mathcal{I}$ *such that*

$$i \prec j \Rightarrow \operatorname{dist}_{G-X}(s, i) \leqslant \operatorname{dist}_{G-X}(s, j)$$

holds for all $i, j \in V$*.*

6.1 Observations

To give intuition, we show a statement related to Conjecture 6.1 where instead of distances after edge deletion, distances in the original graph are measured.

Observation 6.2. *Let* $G = (V, E)$ *be a proper interval graph with an interval order* $(V, \prec)$ *such that* $s \in V$ *is minimum with respect to* $\prec$*. Then*

$$i \prec j \Rightarrow \operatorname{dist}_{G}(s, i) \leqslant \operatorname{dist}_{G}(s, j)$$

holds for all $i, j \in V$*.*

Proof. Let $G = (V, E)$ be a proper interval graph with an interval order $(V, \prec)$ such that $s \in V$ is minimum with respect to $\prec$. Let $\mathcal{M}$ be an order-fitting interval model of G and $\prec$ and let $I_v := [\triangleright_v, \triangleleft_v] \in \mathcal{M}$ be the interval corresponding to v for all $v \in V$. Assume towards a contradiction that there are two vertices $i, j \in V$ with $i \prec j$ such that $\mathrm{dist}_G(s, j) < \mathrm{dist}_G(s, i)$. Observe that either $i = \mathrm{pred}(j)$ holds or there must be $i', j' \in V$ with $i \preccurlyeq i' = \mathrm{pred}(j') \prec j' \preccurlyeq j$ and $\mathrm{dist}_G(s, j') < \mathrm{dist}_G(s, i')$. Assume without loss of generality the former, that is $i = \mathrm{pred}(j)$, as otherwise we could just choose i' and j' instead of i and j. It holds that there is a shortest s-j-path in G of the form $s \rightsquigarrow h \to j$ for some $h \in V$ with $\{h, j\} \in E$. From $i \prec j$ it follows that

$$\triangleright_i \leqslant \triangleright_j < \triangleleft_j .$$

As $i = \mathrm{pred}(j)$, $\triangleright_i$ is the greatest value with $\triangleright_i \leqslant \triangleright_j$ among all $\triangleright_v$ for $v \in V$. Hence, from $\{h, j\} \in E$ and Definitions 24 and 25 it follows either

$$\triangleright_h \leqslant \triangleright_i \leqslant \triangleright_j \leqslant \triangleleft_h \leqslant \triangleleft_i \leqslant \triangleleft_j,$$

if $h \prec j$ or

$$\triangleright_i \leqslant \triangleright_j \leqslant \triangleright_h \leqslant \triangleleft_i \leqslant \triangleleft_j \leqslant \triangleleft_h,$$

if $j \prec h$. In either case we have

$$I_h \cap I_i = [\triangleright_h, \triangleleft_h] \cap [\triangleright_i, \triangleleft_i] \neq \emptyset$$

and thus $\{h, i\} \in E$ by Definition 25. This however implies that there is a shortest path of the form $s \rightsquigarrow h \to i$. As a shortest path of the form $s \rightsquigarrow h \to j$ has length $\mathrm{dist}_G(s, j)$, it follows that a shortest path of the form $s \rightsquigarrow h$ has length $\mathrm{dist}_G(s, j) - 1$. Thereby, a shortest path of the form $s \rightsquigarrow h \to i$ has length $\mathrm{dist}_G(s, j) - 1 + 1 = \mathrm{dist}_G(s, j)$, a contradiction to $\mathrm{dist}_G(s, j) < \mathrm{dist}_G(s, i)$. ■

Conjecture 6.1 now states that among the solutions of an U-SP-MVE *yes*-instance, there is always one solution that, when its edges are removed from the graph, leaves the property stated in Observation 6.2 intact. The proof of the latter depicts another important property of the proper interval graphs: If two vertices are adjacent, then all vertices that come between the two in any interval order are also adjacent to them.

Lemma 6.3. *Let $G = (V, E)$ be a proper interval graph with an interval order $(V, \prec)$. Let $i, v, j \in V$ with $i \prec v \prec j$ and $\{i, j\} \in E$. Then also $\{i, v\}, \{v, j\} \in E$.*

Proof. Let $G = (V, E)$ be a proper interval graph with an interval order $(V, \prec)$. Let $\mathcal{M}$ be an order-fitting interval model of G and $\prec$ and let $I_v := [\triangleright_v, \triangleleft_v] \in \mathcal{M}$ be the interval corresponding to v for all $v \in V$. Further let $i, v, j \in V$ with $i \prec v \prec j$ and $\{i, j\} \in E$. Due to $\{i, j\} \in E$, $i \prec j$ and Definition 24 we have that

$$\triangleright_i \leqslant \triangleright_j \leqslant \triangleleft_i \leqslant \triangleleft_j$$

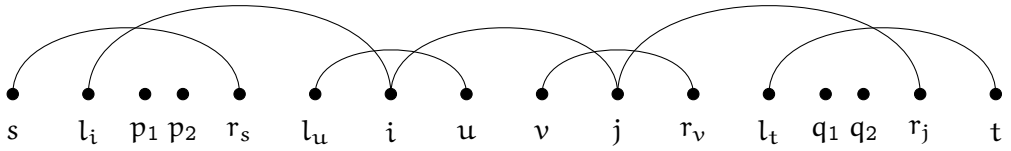

Figure 4: A proper interval graph for which an optimum solution of an associated U-SP-MVE instance increases the distance between two neighbors by more than one, without disconnecting them. Vertices are drawn in interval order; some edges are omitted and can be inferred from those that are drawn based on Lemma 6.3.

and with $i \prec v \prec j$ and Definition 24,

$$\triangleright_i \leqslant \triangleright_v \leqslant \triangleright_j \leqslant \triangleleft_i \leqslant \triangleleft_v \leqslant \triangleleft_j$$

follows. This then implies

$$I_i \cap I_v = [\triangleright_i, \triangleleft_i] \cap [\triangleright_v, \triangleleft_v] \neq \emptyset \text{ and}$$
$$I_v \cap I_j = [\triangleright_v, \triangleleft_v] \cap [\triangleright_j, \triangleleft_j] \neq \emptyset.$$

By Definition 25 we can conclude that $\{i, v\}, \{v, j\} \in E$. ■

During our attempts to prove Conjecture 6.1, we considered the idea that no minimum solution to an U-SP-MVE *yes*-instance on proper interval graphs would increase the distance between any two adjacent vertices to more than two, except for when they get disconnected. However, this is not the case. We give a counter-example.

Observation 6.4. *There is a yes-instance* $\mathcal{I} = (G = (V, E), k, \ell, s, t)$ *of* U-SP-MVE *such that*

- G *is a proper interval graph with an interval order* $(V, \prec)$ *such that* $s \in V$ *is minimum with respect to* $\prec$ *and* $t \in V$ *is maximum with respect to* $\prec$*,*
- G *admits no* s-t*-cut of size at most* k*, and*
- $\mathcal{I}$ *admits a minimum solution* $X \subseteq E$ *such that there are* $i, j \in V$ *with* $\{i, j\} \in E$ *and* $|\mathrm{dist}_{G-X}(s, i) - \mathrm{dist}_{G-X}(s, j)| > 2$*.*

Recall that, if s and t are not disconnected by a minimum solution, then neither are i and j as of Observation 2.3.

Proof. Let $\mathcal{I} := (G, k = 3, \ell = 7, s, t)$ be a problem instance of U-SP-MVE. Let $G = (V, E)$ be the proper interval graph given in Figure 4. Formally, it holds that

$$V = \{s, l_i, p_1, p_2, r_s, l_u, i, u, v, j, r_v, l_t, q_1, q_2, r_j, t\}$$

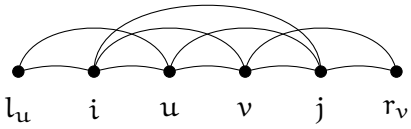

Figure 5: A subgraph of the one shown in Figure 4, with all edges drawn.

with

$$s \prec l_i \prec p_1 \prec p_2 \prec r_s \prec l_u \prec i \prec u \prec v \prec j \prec r_v \prec l_t \prec q_1 \prec q_2 \prec t$$

and E is the smallest set such that all of the following holds.

- G is a proper interval graph,
- $\{i, j\} \in E$,
- $\forall x \in V \,.\, l_x \in V \Rightarrow \{l_x, x\} \in E$ and
- $\forall x \in V \,.\, r_x \in V \Rightarrow \{x, r_x\} \in E$.

We have that $\mathrm{dist}_G(s,t) = \mathrm{dist}_G(s, l_i, i, j, r_j, t) = 5$ and thus $b = 2$. It holds that $k = 3$ cuts are not sufficient to increase the distance between s and l_u as $\{s, l_u\} \notin E$ and there are at least 4 edge-disjoint paths of length 2 from s to l_u:

$$s \to l_i \to l_u, \qquad s \to p_1 \to l_u, \qquad s \to p_2 \to l_u, \qquad s \to r_s \to l_u.$$

By symmetry we obtain that k cuts are not sufficient to increase the distance between r_v and t. Analogously, we can show that k cuts are not sufficient to increase the distances between s and i and between j and t, respectively. Observe that $\mathrm{dist}_{G[V \setminus \{l_u, i\}]}(s,t) = \infty$. Hence, any s-t-path in G contains either l_u or i or both. By symmetry it holds that any s-t-path in G contains either j or r_v or both. It follows that, in order to increase the distance between s and t, we have to delete edges connecting vertices between l_u and r_v in the interval order. Figure 5 shows the subgraph in question. Recall that for any edge deletion set $S \subseteq E$ with $|S| \leqslant k$ it holds that $\mathrm{dist}_{G-S}(s, l_u) = \mathrm{dist}_{G-S}(s, i) = 2$ and $\mathrm{dist}_{G-S}(j, t) = \mathrm{dist}_{G-S}(r_v, t) = 2$. Furthermore, it holds that in G there are exactly 5 paths of length up to 2 that start in either l_u or i and end in either t or r_v such that either l_u or i are used but not both and such that either t or r_v are used but not both:

$$l_u \to u \to j, \qquad i \to u \to j, \qquad i \to v \to j, \qquad i \to v \to r_v, \qquad i \to j.$$

In order to destroy all these paths, we need to destroy $i \to j$ in particular and hence we need to delete $\{i, j\}$. The remaining four paths persist in $G - \{\{i, j\}\}$ as they only contain the edges $E' = \{\{l_u, u\}, \{u, j\}, \{i, u\}, \{i, v\}, \{v, j\}, \{v, r_v\}\}$. By trying all edges in E' we can show that deleting a single edge is not sufficient to destroy all four paths in question. By trying all size-2 subsets of E' we can show that there is only one size-2 subset $S \subseteq \binom{E}{2}$

such that at least one of its elements is contained in every remaining path. It holds that $S = \{\{i, v\}, \{u, j\}\}$. It follows that $X := S \cup \{\{i, j\}\} = \{\{i, j\}, \{i, v\}, \{u, j\}\}$ is the unique solution of $\mathcal{I}$ with

$$\operatorname{dist}_{G-X}(s, t) = \operatorname{dist}_{G-X}(s, l_i, i, u, v, j, r_j, t) = 7 = \ell < \infty$$

and $|X| = 3 = k$. Hence, we have shown that $\mathcal{I}$ is a *yes*-instance of U-SP-MVE and G admits no s-t-cut of size at most k. It remains to observe that $\{i, j\} \in E$ and

$$|\operatorname{dist}_{G-X}(s, i) - \operatorname{dist}_{G-X}(s, j)| = |2 - 5| = 3 > 2$$

holds. ■

The principle idea behind all attempts to prove Conjecture 6.1 so far has been that, given an optimum solution to an U-SP-MVE instance on proper interval graphs with terminals as in the conjecture, we pick a certain pair of „conflicting“ vertices, being vertices whose distance to the first vertex in the order does *not* increase monotonically with respect to their own index in the order. We would then edit the solution to an equivalent one such that this pair of vertices is not in conflict anymore. If this can be continued iteratively in a fashion that terminates in a state where no conflicting pair of vertices persists in the modified solution, then this proves Conjecture 6.1.

Unfortunately, the necessary modifications of the solution became fairly complex, to the point where we failed to show that no other vertex was freshly put into conflict with another one or that the modification yields an equivalent solution in the first place. Note that there is a variety of strategies to pick a conflicting pair of vertices. For example, one could pick the smallest vertex in the order that is in conflict with another one and then pick the biggest vertex in the order that it conflicts with. Another strategy would be to always pick adjacent vertices in the order. Despite our failed attempts to prove it, we still believe Conjecture 6.1 to be correct.

6.2 Towards a polynomial-time algorithm

We show that, if Conjecture 6.1 holds, then it is possible to decide according instances of U-SP-MVE on proper interval graphs in polynomial time via dynamic programming. First we introduce the notion of distance classes.

Definition 29 (Distance Classes)**.** Let $G = (V, E)$ be a proper interval graph with an interval order $(V, \prec)$. Let $X \subseteq E$ be an edge deletion set such that $i \prec j \Rightarrow \operatorname{dist}_{G-X}(s, i) \leqslant \operatorname{dist}_{G-X}(s, j)$ holds for all $i, j \in V$. Then we call a function $P : \mathbb{N} \to \mathcal{P}(V)$ with $P(d) = \{v \in V \mid \operatorname{dist}_{G-X}(s, v) = d\}$ a *distance class partition* of $G - X$ and we call $P(d)$ a *distance class* with *distance* d for all $d \in \mathbb{N}$.

From Observation 6.2 we know that every proper interval graph admits a distance class partition. This corresponds to choosing $X = \emptyset$ in Definition 29. A nice property

of distance classes is that they persist in certain induced subgraphs, as stated by the following lemma.

Lemma 6.5. *Let* $G' = (V, E')$ *be a proper interval graph with an interval order* $(V, \prec)$*. Let* $X' \subseteq E'$ *such that* $G := (V, E) := G' - X'$ *admits a distance class partition* $P : \mathbb{N} \to \mathcal{P}(V)$*. Then* $G\big[\{v \in V \mid v \preccurlyeq u\}\big]$ *admits a distance class partition* P_u *for all* $u \in V$ *with* $P_u(d) \subseteq P(d)$ *for all* $d \in \mathbb{N}$*.*

Note that the graph G in Lemma 6.5 is not necessarily a proper interval graph; it is the result of deleting edges from a proper interval graph such that the resulting graph admits a distance class partition. The proof of the lemma follows.

Proof. Let G and P as in the lemma. Then $i \prec j \Rightarrow \operatorname{dist}_G(s, i) \leqslant \operatorname{dist}_G(s, j)$ holds for all $i, j \in V$. Let $u \in V$ and let $G_u := (V_u, E_u) := G\big[\{v \in V \mid v \preccurlyeq u\}\big]$. Without loss of generality, assume $G_u \neq G$. Think of u as an *upper boundary vertex*. Let $v_u \in V_u \subseteq V$. Assume towards a contradiction that $\operatorname{dist}_{G_u}(s, v_u) \neq \operatorname{dist}_G(s, v_u)$. In the case of $s = v_u$ the contradiction is obvious, so assume that $s \prec v_u$. As G_u is an induced subgraph of G, every shortest path in G_u is also a shortest path in G. Hence, we have $\operatorname{dist}_G(s, v_u) < \operatorname{dist}_{G_u}(s, v_u)$. This then implies that there is a shortest s-v_u-path Q in G that contains a vertex $v \in V \setminus V_u$ and has the form $s \rightsquigarrow v \rightsquigarrow v_u$. From $v_u \prec v$ and the fact that P is a distance class partition of G it follows that $\operatorname{dist}_G(s, v_u) \leqslant \operatorname{dist}_G(s, v)$. This however contradicts that Q is a shortest path in G, as there is a shorter s-v_u-path that does not contain v. Thus, $\operatorname{dist}_{G_u}(s, v_u) = \operatorname{dist}_G(s, v_u)$ holds for all $v_u \in V_u$. It follows that G_u admits a distance class partition $P_u : \mathbb{N} \to \mathcal{P}(V_u)$ with $v_u \in P_u(d) \Leftrightarrow v_u \in P(d)$ for all $v_u \in V_u$ and $d \in \mathbb{N}$. Due to $V_u \subseteq V$, we have $P_u(d) \subseteq P(d)$ for all $d \in \mathbb{N}$. ■

With the notion of distance classes at hand, we are ready to state and proof the central theorem of this section.

Theorem 6.6. *Let* $G = (V, E)$ *be a proper interval graph with an interval order* $(V, \prec)$ *such that* $s \in V$ *is minimum with respect to* $\prec$ *and* $t \in V$ *is maximum with respect to* $\prec$*. Let* $\mathcal{I} = (G, k, \ell, s, t)$ *be a problem instance of* U-SP-MVE*. If Conjecture 6.1 holds, then* $\mathcal{I}$ *can be decided in* $O(n^5)$ *time.*

Proof. Let $\mathcal{I} := (G = (V, E), k, \ell, s, t)$ be a problem instance of U-SP-MVE such that G is a proper interval graph with an interval order $(V, \prec)$ such that $s \in V$ is minimum with respect to $\prec$ and $t \in V$ is maximum with respect to $\prec$. Observe that we can decide in $\mathcal{O}(nm + n^2 \log n)$ time whether G admits an s-t-cut of size k [25]. If it does admit such a cut, then $\mathcal{I}$ is a *yes*-instance. In the following we assume without loss of generality that G does not admit an s-t-cut of size at most k. Next, let

$$G_u := (V_u, E_u) := G\big[\{v \in V \mid s \preccurlyeq v \preccurlyeq u\}\big]$$

for all $u \in V$ be the induced subgraph of G that contains all vertices up to u in the interval order. Last, let $C[\ulcorner p, \ulcorner c, c\urcorner, d]$ for all $\ulcorner p, \ulcorner c, c\urcorner \in V$ with $\ulcorner p \preccurlyeq p\urcorner := \operatorname{pred}(\ulcorner c), \ulcorner c \preccurlyeq c\urcorner$

and $d \in \{1, \dots, n-1\}$ denote the minimum size of an edge deletion set $X_{\lceil c \rceil} \subseteq E_{\lceil c \rceil}$ such that $G_{\lceil c \rceil} - X_{\lceil c \rceil}$ has a distance class partition $P_{\lceil c \rceil} : \mathbb{N} \to \mathcal{P}(V_{\lceil c \rceil})$ with

$$P_{\lceil c \rceil}(d) = \{v \in V_{\lceil c \rceil} \mid \lceil c \preccurlyeq v \preccurlyeq \bar{c} \rceil\} \text{ and}$$
$$P_{\lceil c \rceil}(d-1) = \{v \in V_{\lceil c \rceil} \mid \lceil p \preccurlyeq v \preccurlyeq \bar{p} \rceil\},$$

if such a set $X_{\lceil c \rceil}$ exists, or ∞, otherwise.[1]

Observations. We present observations on C. On the basis of these observations, we will later give a dynamic program that computes C. We distinguish two base cases and several subcases for both of them. In all cases, let $\lceil p, \bar{p}, \lceil c, \bar{c} \rceil \in V$ and $d \in \{1, \dots, n-1\}$ as in the definition of C, that is such that $\lceil p \preccurlyeq \bar{p} \rceil := \text{pred}(\lceil c)$ and $\lceil c \leqslant \bar{c} \rceil$ hold.

Case 1: $\lceil p = s \vee d = 1$. We distinguish a number of subcases.

Case 1.1: $\lceil p = s = \bar{p} \rceil$, $\bar{c} \rceil \in N(s)$, and $d = 1$. By Lemma 6.3 we have that $\bar{V} := \{v \in V \mid \lceil c \preccurlyeq v \preccurlyeq \bar{c} \rceil\} \subseteq N(s)$. Clearly, $\text{dist}_{G_{\lceil c \rceil} - \emptyset}(s, v) = 1$ holds for all $v \in \bar{V}$ and $\text{dist}_{G_{\lceil c \rceil} - \emptyset}(s, v) = 0$ holds for all $v \in \{s\} = \{v \in V \mid \lceil p \preccurlyeq v \preccurlyeq \bar{p} \rceil\}$. Hence, $|\emptyset| = 0$ is the minimum size of an edge deletion set $X_{\lceil c \rceil} \subseteq E_{\lceil c \rceil}$ such that $G_{\lceil c \rceil} - X_{\lceil c \rceil}$ has a distance class partition $P_{\lceil c \rceil} : \mathbb{N} \to \mathcal{P}(V_{\lceil c \rceil})$ with

$$P_{\lceil c \rceil}(1) = \{v \in V_{\lceil c \rceil} \mid \lceil c \preccurlyeq v \preccurlyeq \bar{c} \rceil\} \text{ and}$$
$$P_{\lceil c \rceil}(0) = \{v \in V_{\lceil c \rceil} \mid \lceil p \preccurlyeq v \preccurlyeq \text{pred}(\lceil c)\}$$

and we can conclude that $C[\lceil p, \lceil c, \bar{c} \rceil, d] = 0$ holds.

Case 1.2: $\lceil p = s = \bar{p} \rceil$, $\bar{c} \rceil \notin N(s)$, and $d = 1$. Assume towards a contradiction that $G_{\lceil c \rceil} - X_{\lceil c \rceil}$ admits a distance class partition $P_{\lceil c \rceil} : \mathbb{N} \to \mathcal{P}(V_{\lceil c \rceil})$ for some $X_{\lceil c \rceil} \subseteq E_{\lceil c \rceil}$ such that $P_{\lceil c \rceil}(1) = \{v \in V \mid \lceil c \preccurlyeq v \preccurlyeq \bar{c} \rceil\}$ holds. From $\bar{c} \rceil \notin N(s)$ it follows that $\text{dist}_{G_{\lceil c \rceil} - X_{\lceil c \rceil}}(s, \bar{c} \rceil) \neq 1$, a contradiction to $\bar{c} \rceil \in P_{\lceil c \rceil}(1)$. Hence, $C[\lceil p, \lceil c, \bar{c} \rceil, d] = \infty$ holds.

Case 1.3: $\lceil p = s \neq \bar{p} \rceil$ and $d = 1$. There can be no distance class partition such that $\lceil p = s$ and $\bar{p} \rceil \neq s$ are in the same distance class, so $C[\lceil p, \lceil c, \bar{c} \rceil, d] = \infty$ holds.

Case 1.4: $\lceil p \neq s$ and $d = 1$. This implies $s \prec \lceil p$. Assume towards a contradiction that $G_{\lceil c \rceil} - X_{\lceil c \rceil}$ admits a distance class partition $P_{\lceil c \rceil} : \mathbb{N} \to \mathcal{P}(V_{\lceil c \rceil})$ for some $X_{\lceil c \rceil} \subseteq E_{\lceil c \rceil}$ such that $P_{\lceil c \rceil}(0) = \{v \in V \mid \lceil p \preccurlyeq v \preccurlyeq \bar{p} \rceil\}$ holds. Due to $s \prec \lceil p$, this leads to $s \notin P_{\lceil c \rceil}(0)$, contradicting the fact that $P_{\lceil c \rceil}$ is a distance class partition. Hence, $C[\lceil p, \lceil c, \bar{c} \rceil, d] = \infty$ holds.

Case 1.5: $\lceil p = s$ and $d \neq 1$. There can be no distance class partition $P_{\lceil c \rceil} : \mathbb{N} \to \mathcal{P}(V_{\lceil c \rceil})$, such that $\lceil p = s \in P_{\lceil c \rceil}(d-1) \neq P_{\lceil c \rceil}(0)$, so $C[\lceil p, \lceil c, \bar{c} \rceil, d] = \infty$ holds.

Observe that the subcase distinction above is complete: It comprises all combinations of truth values of the predicates $\lceil p = s$, $s = \bar{p} \rceil$, $\bar{c} \rceil \in N(s)$, and $d = 1$ for which the base case condition of $\lceil p = s \vee d = 1$ holds. We move on to the second base case.

[1] The idea behind the variable names is that $\lceil p$ and $\bar{p} \rceil$ mark the first and the last vertex with respect to $\prec$ in the *preceding* distance class and $\lceil c$ and $\bar{c} \rceil$ mark the first and last vertex in the *current* distance class, respectively.

Case 2: $\ulcorner p \neq s \wedge d \neq 1$. This implies $s \prec \ulcorner p \preccurlyeq p\urcorner = \mathrm{pred}(\ulcorner c), \ulcorner c \preccurlyeq c\urcorner$, and $d \in \{2, \ldots, n-1\}$. We distinguish a number of subcases.

Case 2.1: $\{p\urcorner, c\urcorner\} \notin E$. Assume towards a contradiction that $C[\ulcorner p, \ulcorner c, c\urcorner, d] < \infty$ holds. Then there is an edge deletion set $X_{c\urcorner} \subseteq E_{c\urcorner}$ such that $G_{c\urcorner} - X_{c\urcorner}$ admits a distance class partition $P_{c\urcorner} : \mathbb{N} \to \mathcal{P}(V_{c\urcorner})$ with

$$P_{c\urcorner}(d) = \{v \in V_{c\urcorner} \mid \ulcorner c \preccurlyeq v \preccurlyeq c\urcorner\} \text{ and}$$
$$P_{c\urcorner}(d-1) = \{v \in V_{c\urcorner} \mid \ulcorner p \preccurlyeq v \preccurlyeq p\urcorner\}.$$

It follows that there is a shortest s-$c\urcorner$-path in $G_{c\urcorner} - X_{c\urcorner}$ of the form $s \rightsquigarrow p \to c\urcorner$ for some $p \in P_{c\urcorner}(d-1)$. For such p we have $\{p, c\urcorner\} \in E_{c\urcorner} \setminus X_{c\urcorner} \subseteq E$. By Lemma 6.3 and $p \preccurlyeq p\urcorner \preccurlyeq c\urcorner$ it follows that also $\{p\urcorner, c\urcorner\} \in E$ must hold, which contradicts the subcase condition of $\{p\urcorner, c\urcorner\} \notin E$. Hence, we have $C[\ulcorner p, \ulcorner c, c\urcorner, d] = \infty$.

Case 2.2: $\{p\urcorner, c\urcorner\} \in E$. We show that

$$C[\ulcorner p, \ulcorner c, c\urcorner, d] = \min_{\ulcorner o \prec \ulcorner p} \left(C[\ulcorner o, \ulcorner p, p\urcorner, d-1] + \widehat{C}(\ulcorner p, \ulcorner c, c\urcorner)\right)$$

with

$$\widehat{C}(\ulcorner p, \ulcorner c, c\urcorner) := \left|\left\{\{v_1, v_2\} \in E \;\middle|\; v_1 \prec \ulcorner p \wedge \ulcorner c \preccurlyeq v_2 \preccurlyeq c\urcorner\right\}\right|.$$

To prove equality of the given terms, we show that they are both an upper and a lower bound to each other.

Upper bound. Let $C[\ulcorner p, \ulcorner c, c\urcorner, d] = \varkappa_{c\urcorner}$. We distinguish two cases: $\varkappa_{c\urcorner} < \infty$ and $\varkappa_{c\urcorner} = \infty$. In the case of $\varkappa_{c\urcorner} < \infty$, there is a minimum edge deletion set $X_{c\urcorner} \subseteq E_{c\urcorner}$ with $|X_{c\urcorner}| = \varkappa_{c\urcorner}$ such that $G_{c\urcorner} - X_{c\urcorner}$ has a distance class partition $P_{c\urcorner} : \mathbb{N} \to \mathcal{P}(V_{c\urcorner})$ with

$$P_{c\urcorner}(d) = \{v \in V_{c\urcorner} \mid \ulcorner c \preccurlyeq v \preccurlyeq c\urcorner\} \text{ and}$$
$$P_{c\urcorner}(d-1) = \{v \in V_{c\urcorner} \mid \ulcorner p \preccurlyeq v \preccurlyeq p\urcorner\}.$$

We construct an edge deletion set $X_{p\urcorner} \subseteq E_{p\urcorner}$ as follows. Let $X_{p\urcorner} := \{e \in X_{c\urcorner} \mid e \in E_{p\urcorner}\}$. As $P_{c\urcorner}(d) = V_{c\urcorner} \setminus V_{p\urcorner}$ holds, we have $X_{c\urcorner} \setminus X_{p\urcorner} = \left\{\{v_1, v_2\} \in E \;\middle|\; v_1 \prec \ulcorner p \wedge \ulcorner c \preccurlyeq v_2 \preccurlyeq c\urcorner\right\}$. By definition of $\widehat{C}$ it follows that

$$C[\ulcorner p, \ulcorner c, c\urcorner, d] = |X_{c\urcorner}| = |X_{p\urcorner}| + \widehat{C}(\ulcorner p, \ulcorner c, c\urcorner). \tag{2}$$

By Lemma 6.5, $G_{p\urcorner} - X_{p\urcorner} = G_{p\urcorner} - X_{c\urcorner} = (G_{c\urcorner} - X_{c\urcorner})[V_{p\urcorner}]$ has a distance class partition $P_{p\urcorner} : \mathbb{N} \to \mathcal{P}(V_{p\urcorner})$ with

$$P_{p\urcorner}(d-1) = \{v \in V_{p\urcorner} \mid \ulcorner p \preccurlyeq v \preccurlyeq p\urcorner\} \text{ and}$$
$$P_{p\urcorner}(d-2) = \{v \in V_{p\urcorner} \mid \ulcorner o' \preccurlyeq v \preccurlyeq \mathrm{pred}(\ulcorner p)\}.$$

for some $\ulcorner o' \in V$ with $\ulcorner o' \prec \ulcorner p$. We know that such $\ulcorner o'$ exists by the condition of $s \prec \ulcorner p$. By Equation (2) and the definition of C it follows that

$$\begin{aligned} C[\ulcorner p, \ulcorner c, c\urcorner, d] &= C[\ulcorner o', \ulcorner p, p\urcorner, d-1] + \widehat{C}(\ulcorner p, \ulcorner c, c\urcorner) \\ &\geqslant \min_{\ulcorner o \prec \ulcorner p} \big(C[\ulcorner o, \ulcorner p, p\urcorner, d-1] + \widehat{C}(\ulcorner p, \ulcorner c, c\urcorner)\big). \end{aligned}$$

In the case of $x_{c\urcorner} = \infty$,

$$C[\ulcorner p, \ulcorner c, c\urcorner, d] = \infty \geqslant \min_{\ulcorner o \prec \ulcorner p} \big(C[\ulcorner o, \ulcorner p, p\urcorner, d-1] + \widehat{C}(\ulcorner p, \ulcorner c, c\urcorner)\big) \in \mathbb{N} \cup \{\infty\}$$

holds as well. □

Lower bound. Let $\ulcorner o \in V$ with $\ulcorner o \prec \ulcorner p$ such that

$$x_{c\urcorner} := C[\ulcorner o, \ulcorner p, p\urcorner, d-1] + \widehat{C}(\ulcorner p, \ulcorner c, c\urcorner)$$

is minimum. Recall that $\{p\urcorner, c\urcorner\} \in E$. We distinguish two cases: $x_{c\urcorner} < \infty$ and $x_{c\urcorner} = \infty$. In the case of $x_{c\urcorner} < \infty$, there is a minimum edge deletion set $X_{p\urcorner} \subseteq E_{p\urcorner}$ with $|X_{p\urcorner}| = x_{p\urcorner} := C[\ulcorner o, \ulcorner p, p\urcorner, d-1]$ such that $G_{p\urcorner} - X_{p\urcorner}$ has a distance class partition $P_{p\urcorner} : \mathbb{N} \to \mathcal{P}(V_{p\urcorner})$ with

$$\begin{aligned} P_{p\urcorner}(d-1) &= \{v \in V_{p\urcorner} \mid \ulcorner p \preccurlyeq v \preccurlyeq p\urcorner\} \text{ and} \\ P_{p\urcorner}(d-2) &= \{v \in V_{p\urcorner} \mid \ulcorner o \preccurlyeq v \preccurlyeq \mathrm{pred}(\ulcorner p)\}. \end{aligned}$$

We construct an edge deletion set $X_{c\urcorner} \subseteq E_{c\urcorner}$ as follows. Let

$$X_{c\urcorner} := X_{p\urcorner} \cup \big\{\{v_1, v_2\} \in E \;\big|\; v_1 \prec \ulcorner p \wedge \ulcorner c \preccurlyeq v_2 \preccurlyeq c\urcorner\big\}.$$

We show that $G_{c\urcorner} - X_{c\urcorner}$ admits a distance class partition $P_{c\urcorner} : \mathbb{N} \to \mathcal{P}(V_{c\urcorner})$. Let

$$\bar{V} := V_{c\urcorner} \setminus V_{p\urcorner} = \{v \in V_{c\urcorner} \mid \ulcorner c \preccurlyeq v \preccurlyeq c\urcorner\}.$$

By definition of $X_{c\urcorner}$, all edges in $X_{c\urcorner} \setminus X_{p\urcorner}$ are incident to vertices in $\bar{V}$ and further all edges incident to vertices in $\bar{V}$ in $G_{c\urcorner} - X_{c\urcorner}$ are incident to vertices in $P_{p\urcorner}(d-1)$. The latter implies that there is no shortest path of the form $s \rightsquigarrow \bar{v} \rightsquigarrow v$ in $G_{c\urcorner}$ with $\bar{v} \in \bar{V}$ and $v \in V_{c\urcorner} \setminus \bar{V}$ as such a path would also have the form $s \rightsquigarrow v_1 \to \bar{v} \to v_2 \rightsquigarrow v$ with $v_1, v_2 \in P_{p\urcorner}(d-1)$, contradicting the fact that $P_{p\urcorner}(d-1)$ is a distance class. The former, that is the fact that all edges in $X_{c\urcorner} \setminus X_{p\urcorner}$ are incident to vertices in $\bar{V}$, then implies that every shortest path in $G_{c\urcorner} - X_{c\urcorner}$ with terminals in $V_{p\urcorner} \subset V_{c\urcorner}$ is also a shortest path in $G_{p\urcorner} - X_{p\urcorner}$. Thus $P_{c\urcorner}(d-i) = P_{p\urcorner}(d-i)$ holds for all $i \in \{1, \ldots, d\}$. It remains to be shown that $P_{c\urcorner}(d) = \bar{V}$. This however follows from Lemma 6.3 and the fact that $p\urcorner \in P_{c\urcorner}(d-1)$ and $\{p\urcorner, \bar{v}\} \in E_{c\urcorner} \setminus X_{c\urcorner}$ for all $\bar{v} \in \bar{V}$. It follows that $P_{c\urcorner}$ is indeed a distance class partition of $G_{c\urcorner} - X_{c\urcorner}$. We can conclude that

$$\begin{aligned} &\min_{\ulcorner o \prec \ulcorner p} \big(C[\ulcorner o, \ulcorner p, p\urcorner, d-1] + \widehat{C}(\ulcorner p, \ulcorner c, c\urcorner)\big) \\ &\quad = C[\ulcorner o', \ulcorner p, p\urcorner, d-1] + \widehat{C}(\ulcorner p, \ulcorner c, c\urcorner) \\ &\quad = |X_{p\urcorner}| + \widehat{C}(\ulcorner p, \ulcorner c, c\urcorner) = |X_{c\urcorner}| \geqslant C[\ulcorner p, \ulcorner c, c\urcorner, d]. \end{aligned}$$

In the case of $x_{\ulcorner c \urcorner} = \infty$, we also obtain

$$\min_{\ulcorner o \urcorner \prec \ulcorner p \urcorner} \left(C[\ulcorner o \urcorner, \ulcorner p \urcorner, p\urcorner, d-1] + \widehat{C}(\ulcorner p \urcorner, \ulcorner c \urcorner, c\urcorner) \right)$$
$$= \infty \geqslant C[\ulcorner p \urcorner, \ulcorner c \urcorner, c\urcorner, d] \in \mathbb{N} \cup \{\infty\}.$$

This concludes the equality claim made in Case 2.2. □

Again, observe that the subcase distinction above is complete as either $\{p\urcorner, c\urcorner\} \in E$ or $\{p\urcorner, c\urcorner\} \notin E$ holds. The same is true for the base case distinction as either $\ulcorner p \urcorner = s \vee d = 1$ or $\ulcorner p \urcorner \neq s \wedge d \neq 1$ holds. Hence, we have given a specification of C that matches its definition. We will next give a dynamic program that computes C efficiently.

Algorithm. We give Algorithm 1 in pseudocode. It computes C and looks up values in the table representation of C to decide $\mathcal{J}$. The structure of the algorithm follows closely the case distinction made above. In addition to C, the algorithm maintains a helper table that stores precomputed results for $\widehat{C}$.

Correctness of the table. We show that the algorithm computes C correctly; by this we mean that if the return statement in line 29 is ignored, then upon reaching line 30 the table C computed by the algorithm matches its formal definition given above. Observe that the algorithm implements both the calculation specification for $\widehat{C}$, that is

$$\widehat{C}(\ulcorner p \urcorner, \ulcorner c \urcorner, c\urcorner) = \left| \{ \{v_1, v_2\} \in E \mid v_1 \prec \ulcorner p \urcorner \wedge \ulcorner c \urcorner \preccurlyeq v_2 \preccurlyeq c\urcorner \} \right|$$

with $\ulcorner p \urcorner, \ulcorner c \urcorner, c\urcorner \in V$, as well as the case distinction for C, as discussed above. In the case of $\widehat{C}$, it holds that every cell that was written at least once before line 12 contains the correct value of $\widehat{C}$ from line 12 onward. We omit a detailed proof of this statement. Observe further that, for every combination of parameters $\ulcorner p \urcorner, \ulcorner c \urcorner, c\urcorner \in V$, and $d \in \{1, \ldots, n-1\}$ in the domain of the the formal entity C, that is such that $\ulcorner p \urcorner \prec \ulcorner c \urcorner \preccurlyeq c\urcorner$ holds, the algorithm writes to the corresponding table cell exactly once. Hence, to show that the algorithm correctly computes C, it is sufficient to show that, within the computation of C that starts after its allocation in line 12, table cells of both $\widehat{C}$ and C are only read after they have been written to at least once. It holds that from line 12 onward, $\widehat{C}$ and C are only read in line 25.

In the case of $\widehat{C}$, the value of $\widehat{C}[\ulcorner p \urcorner, \ulcorner c \urcorner, c\urcorner]$ is read. Due to $\ulcorner p \urcorner, \ulcorner c \urcorner, c\urcorner \in V$ with $\ulcorner p \urcorner \prec \ulcorner c \urcorner \preccurlyeq c\urcorner$ it holds that this value was written in line 7.

In the case of C, the value of $C[\ulcorner o \urcorner, \ulcorner p \urcorner, p\urcorner, d-1]$ is read. From the condition in line 23 we obtain $d \neq 1$. In line 13, the algorithm iterates over d from 1 to $n-1$. Since $d \in \{2, \ldots, n-1\}$ and $d-1 < d$ hold there was a preceding iteration step of the loop in line 13 where the values $C[\ulcorner o' \urcorner, \ulcorner p' \urcorner, p\urcorner', d-1]$ for all $\ulcorner o' \urcorner, \ulcorner p' \urcorner, p\urcorner' \in V$ with $\ulcorner o' \urcorner \prec \ulcorner p' \urcorner \preccurlyeq p\urcorner'$ were written. Hence, it remains to be shown that $\ulcorner o \urcorner \prec \ulcorner p \urcorner \preccurlyeq p\urcorner$ holds. From line 15 we obtain $\ulcorner p \urcorner \prec \ulcorner c \urcorner$ and from line 16 we get $p\urcorner = \mathrm{pred}(\ulcorner c \urcorner)$ and thus $\ulcorner p \urcorner \preccurlyeq p\urcorner$. It follows from the conditions in line 25 that $\ulcorner o \urcorner \prec \ulcorner p \urcorner$ and thus $\ulcorner o \urcorner \prec \ulcorner p \urcorner \preccurlyeq p\urcorner$ holds.

Algorithm 1 Decide U-SP-MVE on proper interval graphs with certain terminals.

Input: An instance $\mathcal{I} = (G = (V, E), k, \ell, s, t)$ of U-SP-MVE such that G is a proper interval graph that admits no s-t-cut of size at most k and has an interval order $(V, \prec)$ such that $s \in V$ is minimum and $t \in V$ is maximum with respect to $\prec$.

Ourput: *yes* if $\mathcal{I}$ is a *yes*-instance, *no* otherwise.

Sort V according to $\prec$.
Allocate a table $\widehat{C} : V \times V \times V \to \{0, \ldots, |E|\}$.
for $\ulcorner p \in V$ **from** $\mathrm{succ}(s)$ **to** $\mathrm{pred}(t)$ **do**
  for $\ulcorner c \in V$ **from** $\mathrm{succ}(\ulcorner p)$ **to** t **do**
    $\urcorner p \leftarrow \mathrm{pred}(\ulcorner c)$
    for $\urcorner c \in V$ **from** $\ulcorner c$ **to** t **do**
      $\widehat{C}[\ulcorner p, \ulcorner c, \urcorner c] \leftarrow 0$
      for $v_1 \in V$ **from** s **to** $\mathrm{pred}(\ulcorner p)$ **do**
        for $v_2 \in V$ **from** $\ulcorner c$ **to** $\urcorner c$ **do**
          if $\{v_1, v_2\} \in E$ **then**
            $\widehat{C}[\ulcorner p, \ulcorner c, \urcorner c] \leftarrow \widehat{C}[\ulcorner p, \ulcorner c, \urcorner c] + 1$
Allocate a table $C : V \times V \times V \times \{1, \ldots, n-1\} \to \{0, \ldots, |E|\} \cup \{\infty\}$.
for $d \in \mathbb{N}$ **from** 1 **to** $n-1$ **do**
  for $\ulcorner p \in V$ **from** s **to** $\mathrm{pred}(t)$ **do**
    for $\ulcorner c \in V$ **from** $\mathrm{succ}(\ulcorner p)$ **to** t **do**
      $\urcorner p \leftarrow \mathrm{pred}(\ulcorner c)$
      for $\urcorner c \in V$ **from** $\ulcorner c$ **to** t **do**
        if $s = \ulcorner p \vee d = 1$ **then**
          if $\ulcorner p = s = \urcorner p \wedge \urcorner c \in N(s) \wedge d = 1$ **then** ▷ Case 1.1
            $C[\ulcorner p, \ulcorner c, \urcorner c, d] \leftarrow 0$
          else ▷ Cases 1.2–1.5
            $C[\ulcorner p, \ulcorner c, \urcorner c, d] \leftarrow \infty$
        else if $s \neq \ulcorner p \wedge d \neq 1$ **then**
          if $\{\urcorner p, \urcorner c\} \in E$ **then** ▷ Case 2.2
            $C[\ulcorner p, \ulcorner c, \urcorner c, d] \leftarrow \min_{\ulcorner o \prec \ulcorner p} \left(C[\ulcorner o, \ulcorner p, \urcorner p, d-1] + \widehat{C}(\ulcorner p, \ulcorner c, \urcorner c) \right)$
          else ▷ Case 2.1
            $C[\ulcorner p, \ulcorner c, \urcorner c, d] \leftarrow \infty$
        if $\urcorner c = t \wedge d \geqslant \ell \wedge C[\ulcorner p, \ulcorner c, \urcorner c, d] \leqslant k$ **then**
          return *yes*
return *no*

Correctness of the decision. We show that the algorithm outputs *yes* if and only if $\mathcal{I}$ is a *yes*-instance.

Forward implication: If the algorithm outputs yes*, then $\mathcal{I}$ is a yes-instance.* If the algorithm outputs *yes*, then by definition of C and the conditions of $C[\ulcorner p, \ulcorner c, \urcorner c, d] \leqslant k < \infty$ and $\urcorner c = t$ in line 28 it holds that there is an edge deletion set X with $|X| \leqslant k$ such that $G - X$ admits a distance class partition $P : \mathbb{N} \to \mathcal{P}(V)$ with $t \in P(d)$, which implies $\mathrm{dist}_{G-X}(s,t) = d$. Due to the condition of $d \geqslant \ell$ in line 28, it follows that $\mathcal{I}$ is a *yes*-instance. □

Reverse implication: If $\mathcal{I}$ is a yes-instance, then the algorithm outputs yes*.* Let $\mathcal{I}$ be a *yes*-instance. Recall that $\mathcal{I}$ does not admit an s-t-cut of size at most k. From Conjecture 6.1 it follows that $\mathcal{I}$ admits a minimum solution $X \subseteq E$ such that $i \prec j \Rightarrow \mathrm{dist}_{G-X}(s,i) \leqslant \mathrm{dist}_{G-X}(s,j) < \infty$ holds for all $i, j \in V$. By Definition 29 it follows that $G - X$ admits a distance class partition with

$$P(d) = \{v \in V \mid \ulcorner c \preccurlyeq v \preccurlyeq t\} \text{ and}$$
$$P(d-1) = \{v \in V \mid \ulcorner p \preccurlyeq v \preccurlyeq \mathrm{pred}(\ulcorner c)\}.$$

for some $\ulcorner p, \ulcorner c \in V$ with $\ulcorner p \prec \ulcorner c$ and some $d \in \mathbb{N}$ with $d \geqslant \ell$ for which $C[\ulcorner p, \ulcorner c, t, d] = |X|$ holds (by definition of C and the fact that X is minimal). Recall that the algorithm computes $C[\ulcorner p', \ulcorner c', \urcorner c', d']$ for all $\ulcorner p', \ulcorner c', \urcorner c' \in V$ with $\ulcorner p' \prec \ulcorner c' \preccurlyeq \urcorner c'$ and all $d' \in \{1, \ldots, n-1\}$. Assume towards a contradiction that $d \notin \{1, \ldots, n-1\}$. If $d \leqslant 0$, then this contradicts $t \in P(d)$ with $s \neq t$. If $d \geqslant n$, then either $\mathcal{I}$ is a *no*-instance or there is an s-t-cut in G of size at most k. Both are contradictions. Hence, we have $d \in \{1, \ldots, n-1\}$. It follows that the algorithm computes $C[\ulcorner p, \ulcorner c, t, d] = |X|$ in particular. As $\mathcal{I}$ is a *yes*-instance, $C[\ulcorner p, \ulcorner c, t, d] \leqslant k$ holds. We have that the algorithm, as it checks in line 28 for all $C[\ulcorner p', \ulcorner c', t, d']$ with $\ulcorner p' \prec \ulcorner c' \preccurlyeq t$ and all $d' \in \{1, \ldots, n-1\}$ whether $C[\ulcorner p', \ulcorner c', t, d'] \leqslant k$ holds, will find that $C[\ulcorner p, \ulcorner c, t, d] \leqslant k$ holds and output *yes*. □

Running time. We show that the algorithm runs in $\mathcal{O}(n^5)$ time. We assume without loss of generality that $\prec$ is given as part of the input. Otherwise we can compute an interval order of G in linear time [10] and, as interval orders are unique up to true twins [12], modify it such that s and t are its minimum and maximum, respectively. The latter can be done in linear time by searching s and t and swapping them in place. Note that, by storing the position of each vertex in the interval order in a table, we can later decide $v_1 \prec v_2$ for all $v_1, v_2 \in V$ in constant time. We assume further that the input of the algorithm represents E in such a way that the algorithm can decide $\{v_1, v_2\} \in E$ for $v_1, v_2 \in V$ in constant time. Otherwise we can transform the given representation of E into a table representation that assigns to all $v_1, v_2 \in V$ whether $\{v_1, v_2\} \in E$ holds. Such a table requires $\mathcal{O}(n^2 + m)$ time to compute.

In line 1, the algorithm sorts V, which can be done in $\mathcal{O}(n \log n)$ time.

For the computation of $\widehat{C}$ that starts in line 2, there are 5 nested loops, each of

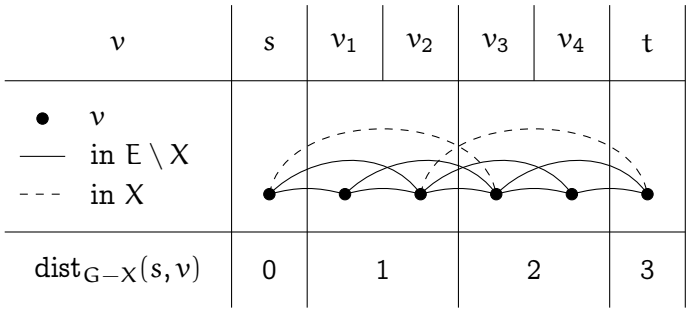

Figure 6: An optimum solution $X \subseteq E$ of an U-SP-MVE *yes*-instance on a proper interval graph $G = (V, E)$. Vertices are drawn in interval order.

which iterates over a subset of V. Within these loops, only constant-time operations are performed. Hence, the algorithm fills $\widehat{C}$ in $\mathcal{O}(n^5)$ time.

Last, the algorithm computes C, starting in line 12. Recall that C has $\mathcal{O}(n^4)$ cells. For each cell, the algorithm either performs constant-time operations or iterates over at most $n-2$ neighbors of $\ulcorner p$ and performs constant-time operations for each of them. Hence, C can be filled in $\mathcal{O}(n^5)$ time as well. ■

Note that the algorithm can be modified to find a solution for either optimization variant of U-SP-MVE; it is sufficient to remove the *return* statements and, once C is filled, look among all $C[\ulcorner p, \ulcorner c, t, d] = x$ with $\ulcorner p, \ulcorner c \in V$ and $\ulcorner p \prec \ulcorner c$ for the smallest $x < \infty$ such that $d \geqslant \ell$ holds (MinCost-U-SP-MVE) or for the greatest d such that $x \leqslant k$ holds (MaxLen-U-SP-MVE).

In the following we give an example execution of the algorithm presented in the proof.

Example. Consider a problem instance $\mathcal{I} = (G, k = 2, \ell = 3, s, t)$ of U-SP-MVE where G is the proper interval graph given in Figure 6. In the first iteration, that is for $d = 1$, the algorithm sets

$$C[s, v_1, v_1, 1] = 0, \qquad C[s, v_1, v_2, 1] = 0, \qquad C[s, v_1, v_3, 1] = 0.$$

All other $C[\ulcorner p, \ulcorner c, \ulcorner c\urcorner, d]$ with $\ulcorner p, \ulcorner c, \ulcorner c\urcorner \in V$, $\ulcorner p \prec \ulcorner c \preccurlyeq \ulcorner c\urcorner$, and $d = 1$ are set to ∞ as they represent an impossible distance class partition. In the second iteration, that is for $d = 2$, the algorithm sets, among other cells of C,

$$\begin{aligned}
C[s, v_3, v_4, 2] &= \infty, \\
C[v_1, v_3, v_4, 2] &= C[s, v_1, v_2, 1] + |\{\{s, v_3\}\}| = 1, \text{ and} \\
C[v_2, v_3, v_4, 2] &= \min_{\ulcorner o \in \{s, v_1\}} \{C[\ulcorner o, v_2, v_2, 1] + \widehat{C}(v_2, v_3, v_4)\} \\
&= \min \left\{ \begin{array}{lll} C[s, v_2, v_2, 1] & + & |\{\{s, v_3\}, \{v_1, v_3\}\}|, \\ C[v_1, v_2, v_2, 1] & + & |\{\{s, v_3\}, \{v_1, v_3\}\}| \end{array} \right\} \\
&= \min\{\infty + 2, \infty + 2\} = \infty.
\end{aligned}$$

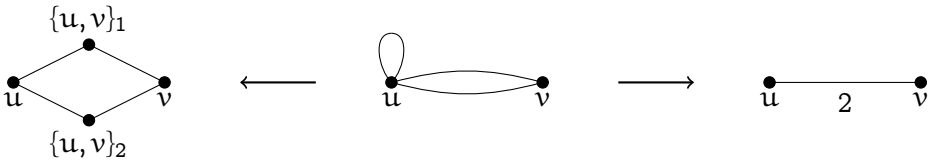

Figure 7: The two reduction strategies applied in Section 7. In the middle is a multigraph. The left- and right hand side depict the results of applying both reduction techniques to the graph in the middle. An edge label represents edge deletion cost.

In the next iteration the algorithm computes the table cells with $d = 3$, among them one that matches the return condition in line 28:

$$\begin{aligned} C[v_3, t, t, 3] &= \min_{\ulcorner o \in \{s, v_1, v_2\}} (C[\ulcorner o, v_3, v_4, 2] + \widehat{C}(v_3, t, t)) \\ &= \min \left\{ \begin{array}{lcl} C[s, v_3, v_4, 2] & + & |\{\{v_2, t\}\}|, \\ C[v_1, v_3, v_4, 2] & + & |\{\{v_2, t\}\}|, \\ C[v_2, v_3, v_4, 2] & + & |\{\{v_2, t\}\}| \end{array} \right\} \\ &= \min\{\infty + 1,\ 1 + 1,\ \infty + 1\} = 2. \end{aligned}$$

This represents the optimum solution given in Figure 6; the distance from s to t can be increased to $3 \geqslant \ell$ by deleting $2 \leqslant k$ edges.

7 Handling multigraphs

So far we have been looking at simple graphs with no loops and with at most one edge between any two distinct vertices. Multigraphs, in their most permitting variant, omit both restrictions. We formalize them in Definition 8 on page 10. An example of multigraphs with parallel edges but no loops are the series-parallel graphs discussed in Section 8.

While the algorithms given in Section 8 are designed to take multigraphs as input, this may not always be a rewarding strategy when designing programs that search most vital edges. For this reason we give two intuitive reductions that allow one to tackle multigraphs in the context of U-SP-MVE as simple graphs in the context of either U-SP-MVE or C-SP-MVE. Subject to the condition that allocating space without initializing it is a constant-time operation, both reductions can be computed in linear time. Otherwise they run in $O(n^2 m)$ time. Both reductions are depicted in Figure 7.

We first show the reduction from multigraph-aware U-SP-MVE to the ordinary version of the same problem variant. The idea is to remove loops, as they never show up in paths of any sort, and subdivide all edges afterwards. By this all distances in the graph are effectively doubled, which is reflected by multiplying the desired shortest s-t-path length ℓ by two.

Theorem 7.1. *Let $\mathcal{I}$ be a problem instance of a variant of* U-SP-MVE *that permits multigraphs. Then $\mathcal{I}$ can be reduced in polynomial time to an equivalent instance of* U-SP-MVE*.*

Proof. We give such a reduction and show its correctness.

Construction. Let $\mathcal{I} := (G = (V, E), k, \ell, s, t)$ be a problem instance of a variant of U-SP-MVE such that G is a multigraph. We reduce $\mathcal{I}$ to an instance $\mathcal{I}' := (G' = (V', E'), k, \ell', s, t)$ of U-SP-MVE as follows. Initially, set $V' := V$ and $E' := \emptyset$. For every distinct $e = \{v_1, v_2\} \in E$ with $v_1 \neq v_2$, add a family of new vertices $\{v_1, v_2\}_i$ for all $i \in \{1, \ldots, 1_E(e)\}$ to V' (assuming without loss of generality that no such vertex is contained in V) and add new edges $\{v_1, \{v_1, v_2\}_i\}$ and $\{\{v_1, v_2\}_i, v_2\}$ for all $i \in \{1, \ldots, 1_E(e)\}$ to E'. Last set $\ell' := 2\ell$.

Correctness. It holds that $\mathcal{I}'$ is a problem instance of U-SP-MVE. We show that $\mathcal{I}$ is a *yes*-instance if and only if $\mathcal{I}'$ is a *yes*-instance.

Forward implication: If $\mathcal{I}$ is a yes-instance, then $\mathcal{I}'$ is a yes-instance. Let $\mathcal{I}$ be a *yes*-instance and let the multiset $X \subseteq E$ be a minimal solution for $\mathcal{I}$, that is $\mathrm{dist}_{G-X}(s, t) \geqslant \ell$ and $|X| \leqslant k$. We construct a solution $X' \subseteq E'$ for $\mathcal{I}'$ as follows. Initially, let $X' := \emptyset$. For all $e \in X$ and for all $i \in \{1, \ldots, 1_E(e)\}$, add $\{v_1, \{v_1, v_2\}_i\}$ to X'.

First we show that $|X'| \leqslant k$. Let

$$x := \sum_{\text{distinct } e \in X} 1_E(e)$$

be the sum of multiplicities with respect to E of all distinct edges that are contained in X. By construction of X' we have that $|X'| = x$. Assume towards a contradiction that $|X| \neq x$ holds. Since $X \subseteq E$ this implies $|X| < x$. It follows that there is an $e := \{u, v\} \in X$ with $1_X(e) < 1_E(e)$. Let the multiset $\bar{X}$ be the proper subset of X where all occurrences of e are removed. Let $\bar{P}$ be a shortest s-t-path in $G - \bar{X}$. If $\bar{P}$ does not contain e, then $\bar{P}$ is also a path in $G - X$ as X and $\bar{X}$ only differ with respect to containment of e. If $\bar{P}$ does contain e, then $\bar{P}$ is of the form $s \rightsquigarrow u \to v \rightsquigarrow t$. Due to $1_X(e) < 1_E(e)$ it holds that there is also a path P of the form $s \rightsquigarrow u \to v \rightsquigarrow t$ in $G - X$. As X is a solution to $\mathcal{I}$, $\bar{X}$ is also a solution to $\mathcal{I}$. Since $\bar{X} \subset X$ this contradicts that X is minimal. Hence $|X| = x$ and thus $|X'| = x = |X| \leqslant k$ holds.

Next we show that $\mathrm{dist}_{G'-X'}(s, t) \geqslant \ell'$. Assume towards a contradiction that there is an s-t-path P' in $G' - X'$ of length $|P'| < \ell'$. By construction of G' we have that P' alternates between vertices in $V \subseteq V'$ and vertices in $V' \setminus V$ such that, if $\{v_1, v_2\}_i \in V' \setminus V$ is contained in P' for some $i \in \{1, \ldots, 1_E(\{v_1, v_2\})\}$, then also $v_1, v_2 \in V$ are contained in P'. Furthermore, it holds that P' does not contain a vertex $e_i \in V' \setminus V$ for any $e \in X$ and $i \in \{1, \ldots, 1_E(e)\}$ as these vertices have degree one in $G' - X'$ by construction of G' and X'. By contraposition we obtain that, if $\{v_1, v_2\}_i \in V' \setminus V$ is contained in P' for some $i \in \{1, \ldots, 1_E(\{v_1, v_2\})\}$, then $\{v_1, v_2\} \notin X$ holds. Hence, there is an s-t-path P

in $G - X$ that contains exactly the vertices in P' that are also contained in V. As P' alternates between vertices in $V \subseteq V'$ and in $V' \setminus V$ and has the terminals $s, t \in V$, we can conclude that $|P'|$ is even and

$$|P| = \frac{|P'|}{2} < \frac{\ell'}{2} = \ell$$

holds, a contradiction to $\mathrm{dist}_{G-X}(s,t) \geqslant \ell$. □

Reverse implication: If $\mathcal{I}'$ is a yes-instance, then $\mathcal{I}$ is a yes-instance. Let $\mathcal{I}'$ be a *yes*-instance and let $X' \subseteq E'$ be a minimal solution for $\mathcal{I}'$, that is $\mathrm{dist}_{G'-X'}(s,t) \geqslant \ell'$ and $|X'| \leqslant k$. As X' is minimal, we have that, for all $\{v_1, v_2\} \in E$ and $i \in \{1, \ldots, 1_E(\{v_1, v_2\})\}$, it is $e_1 := \{v_1, \{v_1, v_2\}_i\} \in X'$ or $e_2 := \{\{v_1, v_2\}_i, v_2\} \in X'$ or $e_1, e_2 \notin X'$ but not both $e_1 \in X'$ and $e_2 \in X'$, as otherwise $\{v_1, v_2\}_i$ would be isolated in $G' - X'$ and $X' \setminus \{e_1\}$ would still be a solution. Furthermore, if $\{v_1, \{v_1, v_2\}_i\} \in X'$ holds for some $\{v_1, v_2\} \in E$ and $i \in \{1, \ldots, 1_E(\{v_1, v_2\})\}$, then $\{v_1, \{v_1, v_2\}_i\} \in X'$ or $\{\{v_1, v_2\}_j, v_2\} \in X'$ holds for all $j \in \{1, \ldots, 1_E(\{v_1, v_2\})\}$, as otherwise

$$\mathrm{dist}_{G'-X'}(v_1, v_2) = \mathrm{dist}_{G'-X'}(v_1, \{v_1, v_2\}_j, v_2) = 2 = \mathrm{dist}_{G'}(v_1, v_2)$$

follows for some $j \in \{1, \ldots, 1_E(\{v_1, v_2\})\}$ and thus $X' \setminus \{v_1, \{v_1, v_2\}_i\}$ would still be a solution.

We construct a new solution $X \subseteq E$ for $\mathcal{I}$ as follows. Initially, let $X := \emptyset$. For every $e \in X'$ with $e = \{v, \{v, v'\}_i\}$ for some $v, v' \in V$ and $i \in \{1, \ldots, 1_E(e)\}$, add $\{v, v'\}$ to X, such that $1_X(\{v, v'\})$ increments. Note that $\{v, v'\} \in E$ holds by construction of G'. We verify that $|X'| = |X| \leqslant k$.

We show that $\mathrm{dist}_{G-X}(s,t) \geqslant \ell$. Assume towards a contradiction that there is a shortest s-t-path P in $G - X$ of length $|P| < \ell$. As P is a path, it does not contain a loop. Thus, all edges e in P are ordinary edges, that is such that $|e| = 2$. Let $\{v_1, v_2\} \in E$ be contained in P. By construction of G' we have that $\{v_1, \{v_1, v_2\}_i\}, \{\{v_1, v_2\}_i, v_2\} \in E'$ for some $i \in \{1, \ldots, 1_E(\{v_1, v_2\})\}$. By construction of X, however, it follows that neither $\{v_1, \{v_1, v_2\}_i\} \in X'$ nor $\{\{v_1, v_2\}_i, v_2\} \in X'$ hold for any $i \in \{1, \ldots, 1_E(\{v_1, v_2\})\}$. Hence, there is an s-t-path P' in $G' - X'$ that contains the vertices $v_1, \{v_1, v_2\}_i, v_2 \in V'$, for some $i \in \{1, \ldots, 1_E(\{v_1, v_2\})\}$, subsequently and in the given order. As we have chosen an arbitrary edge $\{v_1, v_2\} \in E \setminus X$ and have shown that it corresponds to a path of the form $v_1 \to v \to v_2$ in $G' - X'$ for some $v \in V'$ with $N(v) = \{v_1, v_2\}$ (by construction of G'), we can conclude that there is also a path P' in $G' - X'$ of length $|P'| = 2|P| < 2\ell = \ell'$ that alternates between vertices in P and vertices in $V' \setminus V$. This, however, contradicts $\mathrm{dist}_{G'-X'}(s,t) \geqslant \ell'$. □

Running time. Fist we copy V to V' in $\mathcal{O}(n)$ time. Then we scan E in $\mathcal{O}(m)$ time. By using a table of size $\mathcal{O}(n^2)$, we can keep track of the umber of occurrences for all edges seen so far. This then allows us to construct E' while scanning E. As we can initialize all relevant cells of the table with a pass over m, it follows that the reduction

can be computed in linear time. Note that this assumes that allocating space for the table without initializing any cells is a constant-time operation. ■

The idea behind the second reduction is to merge parallel edges into one edge while keeping track of their former multiplicity via the edge deletion cost function σ. While in the multigraph we would need to delete all edges connecting two vertices in order to increase the distance between them, we now only need to delete one but at the same cost.

While this reduction seems less powerful than the one presented in Theorem 7.1 as it elevates the instance to a more general problem variant, there may still be applications that favor this approach as it does not increase the amount of vertices or edges. Furthermore, either reduction may preserve structural properties of the input graph that the other destroys.

Theorem 7.2. *Let $\mathcal{I} := ((V, E), k, \ell, s, t)$ be a problem instance of a variant of* U-SP-MVE *that permits multigraphs. Then one can reduce $\mathcal{I}$ in polynomial time to an equivalent instance $\mathcal{I}' := ((V, E'), k, \ell, s, t, \sigma)$ of* C-SP-MVE *such that $|E'| \leqslant |E|$.*

Proof. Again we give such a reduction and show its correctness.

Construction. Let $\mathcal{I} := (G = (V, E), k, \ell, s, t)$ be a problem instance of a variant of U-SP-MVE such that G is a multigraph. We reduce $\mathcal{I}$ to an instance $\mathcal{I}' := (G' = (V, E'), k, \ell, s, t, \sigma)$ of C-SP-MVE as follows. Set $E' := \{\{u, v\} \in E \mid u \neq v\}$ such that E' is an ordinary set and set $\sigma : E' \to \mathbb{N}^+$ with $\sigma(e) := 1_E(e)$.

Correctness. It holds that $\mathcal{I}'$ is a problem instance of C-SP-MVE. We show that $\mathcal{I}$ is a *yes*-instance if and only if $\mathcal{I}'$ is a *yes*-instance.

Forward implication: If $\mathcal{I}$ is a yes-instance, then $\mathcal{I}'$ is a yes-instance. Let $\mathcal{I}$ be a *yes*-instance and let the multiset $X \subseteq E$ be a minimal solution for $\mathcal{I}$, that is $\operatorname{dist}_{G-X}(s, t) \geqslant \ell$ and $|X| \leqslant k$. We construct a solution for $\mathcal{I}'$ as follows. Let $X' := \{\{u, v\} \in X \mid u \neq v\}$ be an ordinary set. First we show that $\sum_{e \in X'} \sigma(e) \leqslant k$ holds. So far we have that

$$\sum_{e \in X'} \sigma(e) = \sum_{e \in X'} 1_E(e) \geqslant |X|.$$

Assume towards a contradiction that $|X| < \sum_{e \in X'} 1_E(e)$. Then at least one of the following must hold.

- There is a loop $e \in X$.
- There is an edge $e \in X$ such that $e \in E \setminus X$.

Assume the first case, that is there is a loop $e \in X$. Let $\bar{X}$ be the multiset obtained by removing all occurrences of e from X. Assume that in $G - \bar{X}$ there is a shortest s-t-path P with $|P| < \ell$. Since X is a solution, that is $\operatorname{dist}_{G-X}(s, t) \geqslant \ell$, P must contain an edge

in $X \setminus \bar{X}$, being e. This contradicts the assumption that P is a path, as paths do not contain loops. Hence, $\bar{X}$ is a solution for $\mathcal{I}$ with $|\bar{X}| < |X|$, contradicting the fact that X is minimal.

Assume the second case, that is there is $e \in X$ such that $e \in E \setminus X$. Again, let $\bar{X}$ be the multiset obtained by removing all occurrences of e from X. Assume that in $G - \bar{X}$ there is a shortest s-t-path P with $|P| < \ell$. It holds that P is also a path in $G - X$ as X and $\bar{X}$ only differ with respect to containment of e and both $e \in E \setminus X$ and $e \in E \setminus \bar{X}$ hold. This contradicts that X is a solution. Hence, $\bar{X}$ is a solution for $\mathcal{I}$ with $\mathrm{dist}_{G-\bar{X}}(s,t) \geqslant \ell$. Since $|\bar{X}| < |X|$ holds, this contradicts that X is minimal.

As both cases are ruled out we can conclude that $\sum_{e \in X'} \sigma(e) = |X| \leqslant k$ holds.

Next we shown that $\mathrm{dist}_{G'-X'}(s,t) \geqslant \ell$. Assume towards a contradiction the opposite, that is $\mathrm{dist}_{G'-X'}(s,t) < \ell$. Then there is an s-t-path P in $G' - X'$ with $|P| < \ell$. By construction, this s-t-path is also an s-t-path in $G - X$. This contradicts that X is a solution. □

Reverse implication: If $\mathcal{I}'$ is a yes-instance, then $\mathcal{I}$ is a yes-instance. Let $\mathcal{I}'$ be a *yes*-instance and let $X' \subseteq E'$ be a minimal solution for $\mathcal{I}'$, that is $\mathrm{dist}_{G'-X'}(s,t) \geqslant \ell$ and $\sum_{e \in X'} \sigma(e) \leqslant k$. We construct a solution for $\mathcal{I}$ as follows. Let $X := \{e \in X'\}$ be a multiset with multiplicities $1_E(e) := \sigma(e)$ for all $e \in X'$. It holds that $|X| = \sum_{e \in X'} \sigma(e) \leqslant k$.

We show that $\mathrm{dist}_{G-X}(s,t) \geqslant \ell$. Assume towards a contradiction the opposite, that is $\mathrm{dist}_{G-X}(s,t) < \ell$. Then there is a path P in $G - X$ with $|P| < \ell$. By construction, this path is also a path in $G' - X'$. This contradicts that X' is a solution. □

Running time. Using a table of size $\mathcal{O}(n^2)$, it is possible to compute the edge deletion cost function σ by reading E and incrementing one entry for every edge in the input in constant time for each $e \in E$. During the same pass over E, E' can be constructed by copying those edges from E that are seen for the first time, as witnessed by the provisional σ. As we can initialize all relevant cells of the table with a pass over m, it follows that the reduction can be computed in linear time. Again this assumes that allocating space for the table without initializing any cells is a constant-time operation. ■

8 Two-Terminal Series-Parallel Graphs

Two-terminal series-parallel graphs are multigraphs that can be constructed from a collection of edges by means of parallel composition and series composition, as defined below. The classic application of this class of graphs is computing the resistance of a mixed series and parallel electrical network, where each edge represents a resistor or other electrical load. Our formal definition of the (undirected) two-terminal series-parallel graphs corresponds mostly to the one given by Brandstädt et al. [6]. For alternative definitions see e.g. de Fluiter [13] and Valdes et al. [26].

Definition 30 (Two-Terminal Labelled Graph)**.** Let $G = (V, E)$ be a multigraph with $s, t \in V$ and $s \neq t$. Then we call the triplet (G, s, t) a *two-terminal labeled graph*.

Definition 31 (Series Composition). Let $\mathcal{G}_1 = (G_1 = (V_1, E_1), s, v)$ and $\mathcal{G}_2 = (G_2 = (V_2, E_2), v, t)$ be two-terminal labeled graphs with $V_1 \cap V_2 = \{v\}$. Let $H := (V_1 \cup V_2, E_1 \cup E_2)$ be the disjoint union of G_1 and G_2 where both occurrences of v are merged. Then we call $\mathrm{SC}(\mathcal{G}_1, \mathcal{G}_2) := (H, s, t)$ the *series composition* of $\mathcal{G}_1$ and $\mathcal{G}_2$.

Definition 32 (Parallel Composition). Let $\mathcal{G}_1 = (G_1 = (V_1, E_1), s, t)$ and $\mathcal{G}_2 = (G_2 = (V_2, E_2), s, t)$ be two-terminal labeled graphs with $V_1 \cap V_2 = \{s, t\}$. Let $H := (V_1 \cup V_2, E_1 \cup E_2)$ be the disjoint union of G_1 and G_2 where both occurrences of s as well as both occurrences of t are merged. Then we call $\mathrm{PC}(\mathcal{G}_1, \mathcal{G}_2) := (H, s, t)$ the *parallel composition* of $\mathcal{G}_1$ and $\mathcal{G}_2$.

Definition 33 (Two-Terminal Series-Parallel Graph). Let SP be the smallest set (with respect to inclusion) of two-terminal labeled graphs such that the following holds.

- $((\{s, t\}, \{\{s, t\}\}), s, t) \in \mathrm{SP}$ for all terminal names s and t.
- If $\mathcal{G}, \mathcal{H} \in \mathrm{SP}$ and $\mathrm{SC}(\mathcal{G}, \mathcal{H})$ is defined, then also $\mathrm{SC}(\mathcal{G}, \mathcal{H}) \in \mathrm{SP}$.
- If $\mathcal{G}, \mathcal{H} \in \mathrm{SP}$ and $\mathrm{PC}(\mathcal{G}, \mathcal{H})$ is defined, then also $\mathrm{PC}(\mathcal{G}, \mathcal{H}) \in \mathrm{SP}$.

Lastly, if $(G, s, t) \in \mathrm{SP}$, then we call G a (*two-terminal*) *series-parallel graph* with terminals s and t. Furthermore, if such G is a simple graph, then we also call it a *simple* (*two-terminal*) *series-parallel graph*.

Convention 2 (Matching terminals). Let $\mathcal{I} = (G, k, \ell, s, t, \sigma, \tau)$ be a problem instance of W-SP-MVE (or a restricted problem variant) such that G is a series-parallel graph. Then we assume that the terminals s and t considered by the problem instance are also the two terminals of the series-parallel graph G.

8.1 Multigraph reductions

As series-parallel graphs are multigraphs, we know from Theorems 7.1 and 7.2 that, whenever we encounter them in the context of U-SP-MVE, we can also represent them as simple graphs in the context of either U-SP-MVE or C-SP-MVE via a linear-time reduction. In the case of series-parallel graphs it happens that the graph obtained via either standard reduction technique given in Section 7 is in fact a simple series-parallel graph. As we will later give a polynomial-time algorithm that decides U-SP-MVE on general series-parallel graphs, we omit a formal proof of this claim.

As a proof sketch, recall that series-parallel graphs can be constructed from a collection of edges by means of parallel and series composition. The reduction described in Theorem 7.1 removes loops and subdivides the remaining edges. It holds that there are no loops in series-parallel graphs as their smallest „building block“ is a two-terminal labeled graph and Definition 33 yields no rule to contract vertices such that loops could be created. This could be shown by structural induction on the rules of Definition 33.

As the result of an edge subdivision can be described as the series composition of two edges, one can also verify that series-parallel graphs are closed under edge subdivision. Thus, series-parallel graphs are closed under the reduction given in Theorem 7.1, too.

The reduction proposed in Theorem 7.2 again removes the loops and then contracts parallel edges. As the latter can be described as the parallel composition of edges, one can verify that series-parallel graphs are also closed under contraction of parallel edges, as implemented by the reduction.

8.2 Hardness results

Baier et al. [2] have shown that W-SP-MVE is NP-complete on series-parallel graphs by reduction from 2-Partition. As a warm-up before identifying tractable scenarios, we give an alternative reduction from the NP-complete Knapsack [17] to W-SP-MVE on simple series-parallel graphs. Note that Baier et al. [2] claim that their reduction can be extended to simple graphs, too. We define Knapsack as follows.

Knapsack

Input: A quintuple (C, w, u, W, U) defined as follows. A set of commodities C, a weight function $w : C \to \mathbb{N}^+$, a value function $u : C \to \mathbb{N}^+$, a bag size W, and a desired bag value U.

Question: Is there a subset $B \subseteq C$ of commodities to put in the bag such that $\sum_{c \in B} w(c) \leqslant W$ and $\sum_{c \in B} u(c) \geqslant U$ hold?

Theorem 8.1. W-SP-MVE *is* NP*-complete on simple series-parallel graphs.*

We reduce from Knapsack; for every commodity of the Knapsack problem instance, we build a two-terminal labeled graph gadget that admits two edge-disjoint paths from one terminal to the other. One of the paths has a length of 2 plus the value of the corresponding Knapsack commodity. The other path has a length of 2 and admits an edge that can be deleted at a cost that is as big as the weight of the corresponding commodity. No other edge deletion is affordable as every other deletion cost exceeds the total budget that we assign to the W-SP-MVE instance. Therefor, one can think of the one edge whose removal is affordable as the „weak edge“ of the gadget.

Recall that picking a commodity in the Knapsack scenario reduces the remaining bag size by the commodity's weight and increases the bag value by the commodity's value. The corresponding gadget has a dual behavior; deleting its weak edge reduces the remaining edge deletion budget by the corresponding commodity's weight and increases the length of a shortest path from one terminal to the other by the corresponding commodity's value.

To complete the reduction we „chain“ all gadgets via series composition. The path length increases achievable between the terminals of individual gadgets now add to the

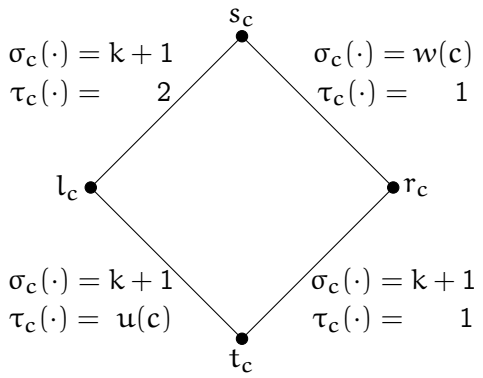

Figure 8: A gadget representing the KNAPSACK commodity c. The placeholder $\cdot$ refers to the edge that is labeled.

length of a shortest path that connects the terminals of the chain, similar to the way that the bag value increases with every picked commodity. While the W-SP-MVE edge deletion budget k is chosen equal to the KNAPSACK bag size, the desired shortest path length ℓ is set to the desired bag value plus twice the number of knapsack commodities to compensate for the gadget's „base length". The formal proof follows.

Proof. Let $\mathcal{I}_{\text{KNAPSACK}} := (C, w, u, W, U)$ be a problem instance of KNAPSACK. We construct an equivalent instance of W-SP-MVE as follows. First we define graph gadgets that represent KNAPSACK commodities within the context of a W-SP-MVE problem instance. For every $c \in C$, we create a two-terminal labeled graph gadget $\mathcal{G}_c := (G_c = (V_c, E_c), s_c, t_c)$ with

$$V_c := \{s_c, t_c, l_c, r_c\} \text{ and}$$
$$E_c := \{\{s_c, l_c\}, \{s_c, r_c\}, \{l_c, t_c\}, \{r_c, t_c\}\}.$$

Let $k := W$. For every $c \in C$ we construct edge length and edge deletion cost functions for G_c as follows.

$$\begin{aligned}
&\sigma_c : E_c \to \mathbb{N}^+, && \tau_c : E_c \to \mathbb{N}^+,\\
&\{s_c, l_c\} \mapsto k+1, && \{s_c, l_c\} \mapsto 2,\\
&\{s_c, r_c\} \mapsto w(c), && \{s_c, r_c\} \mapsto 1,\\
&\{l_c, t_c\} \mapsto k+1, && \{l_c, t_c\} \mapsto u(c),\\
&\{r_c, t_c\} \mapsto k+1. && \{r_c, t_c\} \mapsto 1.
\end{aligned}$$

Figure 8 shows a gadget with its edge weights. Next choose any order $(C, \prec)$. We want to compute the series composition of all gadgets in this order. To make this possible, we first need to rename vertices. For any two gadgets $\mathcal{G}_{c_1}$ and $\mathcal{G}_{c_2}$ with $c_1 = \text{pred}(c_2)$, we rename both t_{c_1} and s_{c_2} to m_{c_1,c_2}. The single remaining s_c for some $c \in C$ we

rename to s. The single remaining t_c for some $c \in C$ we rename to t. We compute G iteratively as follows. Let $c_1 \prec \ldots \prec c_{|C|}$ be the order of commodities chosen above. Let $\mathcal{H}_2 := \mathrm{SC}(\mathcal{G}_{c_1}, \mathcal{G}_{c_2})$ and let $\mathcal{H}_i := \mathrm{SC}(\mathcal{H}_{i-1}, \mathcal{G}_{c_i})$ for all $i \in \{3, \ldots, |C|\}$. Last, identify $\mathcal{H}_{|C|}$ with (G, s, t).

We show that G is a series-parallel graph. First we show that the graph of every gadget is a series-parallel graph. Let $\mathcal{G}_c := (G_c = (V_c, E_c), s_c, t_c)$ be the gadget for some $c \in C$ before vertices were renamed. Consider the following four two-terminal labeled graphs.

$$\begin{aligned}
\mathcal{G}_{\{s_c,l_c\}} &:= ((\{s_c, l_c\}, \{\{s_c, l_c\}\}), s_c, l_c),\\
\mathcal{G}_{\{s_c,r_c\}} &:= ((\{s_c, r_c\}, \{\{s_c, r_c\}\}), s_c, r_c),\\
\mathcal{G}_{\{l_c,t_c\}} &:= ((\{l_c, t_c\}, \{\{l_c, t_c\}\}), l_c, t_c),\\
\mathcal{G}_{\{r_c,t_c\}} &:= ((\{r_c, t_c\}, \{\{r_c, t_c\}\}), r_c, t_c).
\end{aligned}$$

By Definition 33 we have that all four are contained in SP. It it is easy to verify that

$$\mathrm{PC}(\mathrm{SC}(\mathcal{G}_{\{s_c,l_c\}}, \mathcal{G}_{\{l_c,t_c\}}), \mathrm{SC}(\mathcal{G}_{\{s_c,r_c\}}, \mathcal{G}_{\{r_c,t_c\}})) = \mathcal{G}_c$$

holds. By Definition 33 it follows that $\mathcal{G}_c \in \mathrm{SP}$ and thus G_c is a series-parallel graph with terminals s_c and t_c. By construction of G it holds that also G is a series-parallel graph with terminals s and t. Furthermore, it holds that G is a simple graph.

Let $\mathcal{I}_{\text{W-SP-MVE}} := (G, k, \ell, s, t, \sigma, \tau)$ with $G = (V, E)$, s, t, and k defined as above and ℓ, σ, and τ defined as follows.

$$\ell := 2|C| + V,$$

$$\sigma : E \to \mathbb{N}^+, \{v, w\} \mapsto \begin{cases} \sigma_c(\{s_c, w\}), & v \in \{s, m_{c',c}\} \wedge w \in \{l_c, r_c\} \text{ for some } c' \in C \\ \sigma_c(\{v, t_c\}), & v \in \{l_c, r_c\} \wedge w \in \{m_{c,c'}, t\} \text{ for some } c' \in C, \end{cases}$$

$$\tau : E \to \mathbb{N}^+, \{v, w\} \mapsto \begin{cases} \tau_c(\{s_c, w\}), & v \in \{s, m_{c',c}\} \wedge w \in \{l_c, r_c\} \text{ for some } c' \in C \\ \tau_c(\{v, t_c\}), & v \in \{l_c, r_c\} \wedge w \in \{m_{c,c'}, t\} \text{ for some } c' \in C. \end{cases}$$

By construction it holds that σ and τ are total functions and $\mathcal{I}_{\text{W-SP-MVE}}$ is a problem instance of W-SP-MVE on G that can be computed in polynomial time given $\mathcal{I}_{\text{Knapsack}}$.

We show that $\mathcal{I}_{\text{Knapsack}}$ is a *yes*-instance if and only if $\mathcal{I}_{\text{W-SP-MVE}}$ is a *yes*-instance.

Forward implication: If $\mathcal{I}_{\text{Knapsack}}$ is a yes-instance, then $\mathcal{I}_{\text{W-SP-MVE}}$ is also a yes-instance. Let $\mathcal{I}_{\text{Knapsack}}$ be a *yes*-instance and let $B \subseteq C$ be a solution for $\mathcal{I}_{\text{Knapsack}}$. Observe that by construction of $\mathcal{I}_{\text{W-SP-MVE}}$, the shortest s-t-path in G is unique and alternates between vertices in $\{s, m_{c_1,c_2}, t\}$ and r_c for some $c, c_1, c_2 \in C$, containing all r_c for all $c \in C$. Hence $\mathrm{dist}_G(s, t) = 2|C|$ and $b = U$ hold. We construct a solution $X \subseteq E$ for $\mathcal{I}_{\text{W-SP-MVE}}$ as follows. For every $c \in B$, add

$$e := \begin{cases} \{s, r_c\}, & \{s, r_c\} \in E \\ \{m_{c',c}, r_c\}, & \{m_{c',c}, r_c\} \in E \text{ for some } c' \in C \end{cases}$$

to X. By construction, the following holds.

$$\sigma(X) = \sum_{e \in X} \sigma(e) = \sum_{c \in B} w(c) \leqslant W = k$$

Consider a shortest s-t-path P in $G - X$. For $c \in C$ with $c \notin B$, P contains r_c as no edge incident to r_c is contained in X and both vertices in $N(r_c)$ must be part of any s-t-path in $G - X$ by construction of G. Hence, the length of the path that is contained in both P and the gadget representing c in G is

$$\sum_{e \in \{\{r_c, x\} \mid x \in N(r_c)\}} \tau(e) = 2.$$

For $c \in C$ with $c \in B$ and thus either $\{s, r_c\} \in X$ or $\{m_{c',c}, r_c\} \in X$ for some $c' \in C$, r_c has degree one in $G - X$ and cannot be part of any s-t-path. Thus, as both vertices in $N(r_c)$ must be part of any s-t-path in $G - X$ and l_c is the only vertex with $N(r_c) \subseteq N(x)$ among all $x \in V$, the shortest s-t-path in $G - X$ must use l_c. Hence, the length of the path that is contained in both P and the gadget representing c in G is

$$\sum_{e \in \{\{l_c, x\} \mid x \in N(r_c) = N(l_c)\}} \tau(e) = 2 + v(c).$$

It follows that the length of P is

$$\sum_{c \in C \setminus B} 2 + \sum_{c \in B} 2 + u(c) = 2|C| + \sum_{c \in B} u(c) \geqslant 2|C| + V = \ell.$$

Hence, X is a solution for $\mathcal{I}_{\text{W-SP-MVE}}$ and $\mathcal{I}_{\text{W-SP-MVE}}$ is a *yes*-instance. □

Reverse implication: If $\mathcal{I}_{\text{W-SP-MVE}}$ is a yes-instance, then $\mathcal{I}_{\text{Knapsack}}$ is also a yes-instance. Let X be a *yes*-instance and let $X \subseteq E$ be a solution for $\mathcal{I}_{\text{W-SP-MVE}}$. Let $e_c := \{x, r_c\}$ for all $c \in C$ and $x \in \{s, m_{c',c}\}$ with $c' \in C$ such that $e_c \in E$. Note that such x exists and is unique given c. (Informally, e_c is the upper right edge in G_c as drawn in Figure 8.) Let $E' := \{e_c \mid c \in C\}$. Observe that for all $e \in E \setminus E'$, $\sigma(e) = k + 1$ and thus $e \notin X$ hold. Hence, we have $X \subseteq E'$. We construct a solution $B \subseteq C$ for $\mathcal{I}_{\text{Knapsack}}$ as follows. For all $e_c \in X \subseteq E'$, add c to B. It holds that

$$\sum_{c \in B} w(c) = \sum_{e_c \in X} w(c) = \sum_{e_c \in X} \sigma(e_c) \leqslant k = W.$$

Next let $e'_c := \{l_c, x\}$ for all $c \in C$ and $x \in \{t, m_{c,c'}\}$ with $c' \in C$ such that $e'_c \in E$. Note that such x exists and is unique given c. (Informally, e'_c is the lower left edge in G_c as drawn in Figure 8.) Consider a shortest s-t-path P in $G - X$. As in the forward implication, we have that P contains either l_c or r_c for every $c \in C$. By construction it

holds that P contains l_c and thus e'_c if and only if $e_c \in X$. Let p_c be the length of the path that is contained in both P and the gadget representing c in G. It holds that

$$p_c = \begin{cases} 2, & r_c \in P \Leftrightarrow e_c \notin X, \\ 2 + \tau(e'_c), & l_c \in P \Leftrightarrow e_c \in X. \end{cases}$$

With $\tau(e_c) = u(c)$ for all $c \in C$ it follows that

$$\begin{aligned} \ell \leqslant |P| = \sum_{c \in C} p_c = \sum_{c \in C} 2 + \sum_{e_c \in X} \tau(e'_c) = 2|C| + \sum_{c \in B} u(c) \\ \Leftrightarrow \ell - 2|C| \leqslant \sum_{c \in B} u(c) \\ \Leftrightarrow V \leqslant \sum_{c \in B} u(c). \end{aligned}$$

Hence, B is a solution for $\mathcal{I}_{\text{KNAPSACK}}$ and $\mathcal{I}_{\text{KNAPSACK}}$ is a *yes*-instance. ■

As is also the case in the reduction by Baier et al. [2], the series-parallel graph produced by our reduction above is also an outerplanar graph. Hence, our reduction also yields the following corollary.

Corollary 8.2. W-SP-MVE *is* NP*-complete on outerplanar graphs.*

Furthermore, as series-parallel graphs have treewidth at most 2 [6], the following holds.

Corollary 8.3. W-SP-MVE *is* NP*-complete on graphs with treewidth at most* 2*.*

Note that Baier et al. [2] do not discuss other problem variants in the context of series-parallel graphs, or approach an algorithm. In the following we aim to do both, building on the foundation they laid.

8.3 Tractability results

Before we develop algorithms to solve Shortest Path Most Vital Edges variants on series-parallel graphs, we require the notion of sp-trees. Informally, an sp-tree of a series-parallel graph G represents a composition pattern that describes how G can be constructed from a number of edges through a sequence of series and parallel compositions. Hence, an sp-tree corresponds to one „realization“ of the recursive definition given in Definition 33.

In this work we limit sp-trees to binary trees, which can represent the same series-parallel graphs as general sp-trees [13]. Figure 9 show a series-parallel graph and an associated binary sp-tree. To help distinguish them from the vertices of the associated series-parallel graph, we refer to the vertices of an sp-tree as *nodes*. The tree structure of an sp-tree is given as an undirected graph as well as a root node. However, as the

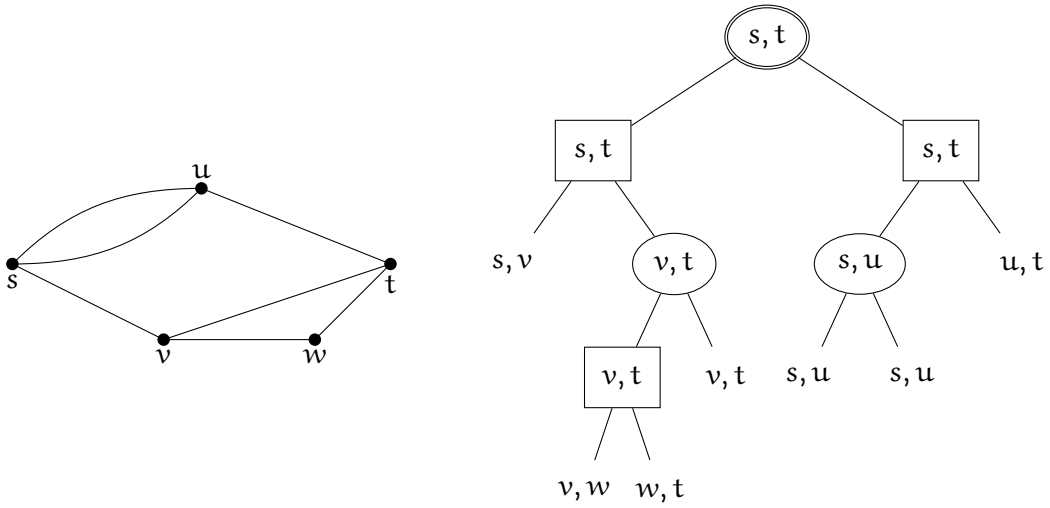

Figure 9: A series-parallel graph with terminals s and t on the left-hand side and its sp-tree representation on the right-hand side. Ellipses represent parallel composition, rectangles represent series composition, and the absence of an outline marks a leaf. Labels in the sp-tree represent node labels. The root node has double outlines.

order of arguments is important for the series composition, we do not give the edge set explicitly but represent it by pointer functions that yield the left and the right child of a node, or $\perp$ in case of a leaf. Depending on the context, we refer to both the graph and the pointer representation of a given sp-tree.

Definition 34 (SP-Tree)**.** Let V be a set of vertices and let Q be a set of nodes. A sextuple $\mathcal{T} := (T, \alpha, \beta, \delta_l, \delta_r, q_0)$ with a graph

$$T := \left(Q, \left\{\{q_1, q_2\} \in \binom{V}{2} \mid \delta_l(q_1) = q_2 \vee \delta_r(q_1) = q_2\right\}\right),$$

node types $\alpha : Q \to \{Ⓢ, Ⓟ, \text{🍃}\}$, node labels $\beta : Q \to V^2$, child pointers $\delta_l, \delta_r : Q \to Q \cup \{\perp\}$, and a root node $q_0 \in Q$ is called an *sp-tree* if the following holds.

- T is a binary tree.
- T has no half-leaves, that is, $\forall q \in Q \,.\, \delta_l(q) = \perp \Leftrightarrow \delta_r(q) = \perp$.
- Exactly the leaves of T have the leaf type, that is $\forall q \in Q \,.\, \delta_l(q) = \perp \Leftrightarrow \alpha(q) = \text{🍃}$.
- Children have matching labels, that is for all $q \in Q$ with $\beta(q) = (u, v)$, the following holds.
  - If $\alpha(q) = Ⓢ$, then $\beta(\delta_l(q)) = (u, x)$ and $\beta(\delta_r(q)) = (x, v)$ for some $x \in V$.
  - If $\alpha(q) = Ⓟ$, then $\beta(\delta_l(q)) = (u, v)$ and $\beta(\delta_r(q)) = (u, v)$.

As we are often interested in undirected edges represented by labels of leaves, we write $\widehat{\beta}(q)$ short for $\{u, v \mid \beta(q) = (u, v)\}$.

Definition 35 (SP-Tree Composition)**.** We say that a series-parallel graph $G = (V, E)$ with terminals s and t is *represented* by an sp-tree $\mathcal{T} := (T = (Q, R), \alpha, \beta, \delta_l, \delta_r, q_0)$ if $\beta(q_0) = (s, t)$ holds and G can be constructed as follows.

Traverse T in post-order, starting at the root q_0. For every $q \in Q$ with $\beta(q) = (u, v)$ in order, do the following.

- If $\alpha(q) =$  , then create a new two-terminal labeled graph
$$\mathcal{G}_q := (G_q := (V_q := \{u, v\}, E_q := \{\{u, v\}\}), u, v).$$
- If $\alpha(q) =$ Ⓢ, then replace $\mathcal{G}_{\delta_l(q)}$ and $\mathcal{G}_{\delta_r(q)}$ with their series composition
$$\mathcal{G}_q := \mathrm{SC}(\mathcal{G}_{\delta_l(q)}, \mathcal{G}_{\delta_r(q)}).$$
- If $\alpha(q) =$ Ⓟ, then replace $\mathcal{G}_{\delta_l(q)}$ and $\mathcal{G}_{\delta_r(q)}$ with their parallel composition
$$\mathcal{G}_q := \mathrm{PC}(\mathcal{G}_{\delta_l(q)}, \mathcal{G}_{\delta_r(q)}).$$

Since children have matching labels, the series and parallel compositions above are well-defined. Last, consider the unique remaining two-terminal labeled graph $\mathcal{G}_{q_0} = (G_{q_0}, s, t)$ and identify G_{q_0} as G.

Given a series-parallel graph, an sp-tree can be computed in linear time [26]. Via dynamic programming on the tree structure, we can now develop efficient algorithms for solving W-SP-MVE on series-parallel graphs. Recall Convention 2 on page 42 for the meaning of the term *matching terminals*.

Theorem 8.4. W-SP-MVE *can be decided in* $\mathcal{O}(mk^2)$ *and in* $\mathcal{O}(m\ell^2)$ *time on series-parallel graphs with matching terminals.*

Proof. We give two algorithms, show their correctness and analyze their running times.

Construction. We give Algorithms 2 and 3 as pseudocode on pages 50 and 53. Both algorithms take a W-SP-MVE problem instance $\mathcal{I}$ on a series-parallel graph G as input and first find an sp-tree $\mathcal{T}$ of G. Via dynamic programming on $\mathcal{T}$, Algorithm 2 then finds the maximum distances achievable between the two terminals of all series-parallel graphs represented by subtrees of $\mathcal{T}$, for all edge deletion budgets up to k. Likewise, Algorithm 3 finds the minimum edge deletion cost to achieve distances up to ℓ in all series-parallel graphs represented by subtrees of $\mathcal{T}$. In other words, Algorithm 2 computes the MaxLen optimization variant of W-SP-MVE while Algorithm 3 computes the MinCost variant. Both algorithms then decide $\mathcal{I}$ by comparing one of their optima for $\mathcal{T}$ (the one for the highest budget spent and the highest length achieved, respectively) with that optimum's

Algorithm 2 Solve W-SP-MVE on series-parallel graphs for small k.

Input: An instance $\mathcal{I} = (G = (V, E), k, \ell, s, t, \sigma, \tau)$ of W-SP-MVE such that G is a series-parallel graph with terminals s and t.

Ourput: *yes* if $\mathcal{I}$ is a *yes*-instance, *no* otherwise.

Find any sp-tree $\mathcal{T} := (T = (Q, R), \alpha, \beta, \delta_l, \delta_r, q_0)$ that represents G.
Let $(q_i)_{i \in \{1,\dots,|Q|\}}$ be the node sequence of a post-order traversal of T starting at q_0 and let $\overline{q} \in \{1, \dots, |Q|\}$ denote the position of q in the sequence for all $q \in Q$.
Fill a table $\mathrm{MaxLen} : \{1, \dots, |Q|\} \times \{0, \dots, k\} \to \mathbb{N}$ as follows.
for i **from** 1 **to** $|Q|$ **do**
    for all $x \in \{0, \dots, k\}$ **do**

$$\mathrm{MaxLen}(i, x) \leftarrow \begin{cases} \tau(\widehat{\beta}(q_i)), & \alpha(q_i) = \text{🍃} \wedge x < \sigma(\widehat{\beta}(q_i)) \\ \infty, & \alpha(q_i) = \text{🍃} \wedge x \geqslant \sigma(\widehat{\beta}(q_i)) \\ \mathrm{OPT}_{\text{Ⓢ}}(i, x), & \alpha(q_i) = \text{Ⓢ} \\ \mathrm{OPT}_{\text{Ⓟ}}(i, x), & \alpha(q_i) = \text{Ⓟ} \end{cases}$$

with

$$\mathrm{OPT}_{\text{Ⓢ}}(i, x) := \max_{x' \in \{0,\dots,x\}} \left(\mathrm{MaxLen}(\overline{\delta_l(q_i)}, x') + \mathrm{MaxLen}(\overline{\delta_r(q_i)}, x - x')\right)$$

and

$$\mathrm{OPT}_{\text{Ⓟ}}(i, x) := \max_{x' \in \{0,\dots,x\}} \min\left\{\mathrm{MaxLen}(\overline{\delta_l(q_i)}, x'),\ \mathrm{MaxLen}(\overline{\delta_r(q_i)}, x - x')\right\}$$

if $\mathrm{MaxLen}(|Q|, k) \geqslant \ell$ **then**
    return *yes*
else
    return *no*

respective threshold in $\mathcal{I}$. Note that both algorithms read G only once in order to compute $\mathcal{T}$. Hence, no special treatment of the multigraph properties of series-parallel graphs is necessary.

Correctness of Algorithm 2. First we show that cells of the table MaxLen are never read without prior initialization. In line 2 we build an ordered sequence $(q_i)_{i\in\{1,\dots,|Q|\}}$ by traversing T in post-order. Let $C_q := \{q_l, q_r\} \subseteq Q$ be the children of an inner node q of T. By Definition 34 this implies $C_q = \{\delta_l(q), \delta_r(q)\}$. As the post-order traversal visits both children of an inner node before their parent, it follows that both $\delta_l(q)$ and $\delta_r(q)$ appear before q in the sequence. The algorithm reads from MaxLen in lines 6 and 7. In line 6, both $\overline{\delta_l(q_i)}$ and $\overline{\delta_r(q_i)}$ yield the index i' of a child of q_i. As shown above, $i' < i$ holds. As i increases by one in every iteration of the loop that starts in line 4 and MaxLen(i, x) is written for all $x \in \{0, \dots, k\}$ in each step, we can conclude that read access in line 6 is only made on cells that have previously been written to. In line 7, MaxLen$(|Q|, k)$ is read. It is easy to see that the nested loops that start in lines 4 and 5 and stop before line 7 write to every cell MaxLen(i, x) with $i \in \{1, \dots, |Q|\}$ and $x \in \{0, \dots, k\}$ and thus to MaxLen$(|Q|, k)$ in particular.

Next we show that MaxLen$(i, x) = y$ holds if and only if the distance between the two terminals in the series-parallel graph $G_i = (V_i, E_i)$ represented by the subtree of $\mathcal{T}$ that is rooted at q_i can be increased to a maximum of y by deleting edges up to a cost of x. We do so for all $i \in \{1, \dots, |Q|\}$ and $x \in \{1, \dots, k\}$ by induction on i. Note that for readability, we deviate from the notation used in Definition 35 by writing e.g. G_i instead of G_{q_i}. Observe that for all $i \in \{1, \dots, |Q|\}$, either the simple case of $\alpha(q_i) = \text{🍃}$ or the inductive step case of $\overline{\delta_l(q_i)}, \overline{\delta_r(q_i)} < i$ holds. As a leaf comes first in the post-order, we have that the induction basis of $i = 1$ is a simple case. For the induction step, consider the following case distinction.

Case 1: $\alpha(q_i) = \text{🍃}$. To obtain $G_i = (V_i, E_i)$, we set $V_i = \{u, v\}$ and $E_i = \{e\}$ with $e := \{u, v\}$. It holds that $\operatorname{dist}_{G_i}(u, v) = \tau(e)$. If $x \geqslant \sigma(e)$ holds for some $x \in \mathbb{N}$, then we can remove e from E_i with a budget of (at least) x and obtain

$$\operatorname{dist}_{G_i - \{e\}}(u, v) = \infty = \text{MaxLen}(i, x).$$

If $x < \sigma(e)$, then we cannot afford to remove e from E_i with a budget of (at most) x. As there is no other edge in E_i, we have

$$\operatorname{dist}_{G_i - X}(u, v) = \tau(e) = \text{MaxLen}(i, x)$$

for all $X \subseteq E_i$ with $\sum \sigma(X) < x$. Hence, MaxLen(i, x) is correct for all $x \in \{1, \dots, k\}$.

Case 2: $\alpha(q_i) = \text{Ⓢ}$. Let $\beta(q_i) = (u, v)$. To obtain $G_i = (V_i, E_i)$, let $l, r \in \{1, \dots, |Q|\}$ such that $\beta(q_l) = (u, w) = \beta(\delta_l(q_i))$ and $\beta(q_r) = (w, v) = \beta(\delta_r(q_i))$ hold for some $w \in V$. Note that such w exists due to Definition 34 and $l, r < i$ holds given the order of traversal used in the algorithm. Next merge both vertices named w in the disjoint

union of $G_l = (V_l, E_l)$ and $G_r = (V_r, E_r)$. Let w also be the name of the resulting vertex in G_i. This construction of G_i corresponds to the i-th iteration of the step-wise construction of G as given in Definition 35. It holds that $E_l \cup E_r = E_i$. We consider all possibilities to distribute our edge deletion budget x between edges in E_l and E_r. As a shortest path from u to v first goes from u to w, using only edges in E_l and then from w to v, using only edges in E_r, the optimal distribution of x is the one that maximizes the sum of the lengths of the two subpaths that use edges in E_l and E_r, respectively. As $\text{MaxLen}(l, x')$ and $\text{MaxLen}(r, x - x')$ contain these lengths for all possible distributions with $x' \in \{0, \ldots, x\}$ by induction hypothesis, it holds that the algorithm computes this maximum and writes it to $\text{MaxLen}(i, x)$.

Case 3: $\alpha(q_i) = \text{Ⓟ}$. Let $\beta(q_i) = (u, v)$. Let $l, r \in \{1, \ldots, |Q|\}$ such that $\beta(q_l) = (u, v) = \beta(\delta_l(q_i))$ and $\beta(q_r) = (u, v) = \beta(\delta_r(q_i))$ hold. Recall that this is possible and implies $l, r < i$. To obtain $G_i = (V_i, E_i)$, first build the disjoint union of $G_l = (V_l, E_l)$ and $G_r = (V_r, E_r)$. Next merge both vertices named u as well as both vertices named v within the disjoint union of $G_l = (V_l, E_l)$ and $G_r = (V_r, E_r)$. Let u and v also be the names of the merged vertices. Again this construction of G_i corresponds to the i-th iteration of the step-wise construction of G as given in Definition 35. It holds that $E_l \cup E_r = E_i$. We consider all possibilities to distribute our edge deletion budget x between edges in E_l and E_r. As all edges of a shortest path from u to v are completely contained in either E_l or E_r, the optimal distribution of x is the one that maximizes the minimum length of a path from u to v that uses either edge set exclusively. As $\text{MaxLen}(l, x')$ and $\text{MaxLen}(r, x - x')$ contain these lengths for all possible distributions with $x' \in \{0, \ldots, x\}$ by induction hypothesis, it holds that the algorithm computes this maximum and writes it to $\text{MaxLen}(i, x)$.

It remains to be shown that $\mathcal{I}$ is a *yes*-instance if and only if $\text{MaxLen}(|Q|, k) \geqslant \ell$. Recall that the algorithm traverses T in post-order. Hence, the root is visited last and thus $\beta(q_{|Q|}) = \beta(q_0) = (s, t)$ holds. By Definition 35 we have that the series-parallel graph represented by the subtree of $\mathcal{T}$ that is rooted at q_0, that is $\mathcal{T}$ itself, is $G_0 = G$. Hence, $\text{MaxLen}(|Q|, k) \geqslant \ell$ is equivalent to the statement that the distance between the two terminals of G, that are s and t, can be increased to at least ℓ by deleting edges up to a cost of x.

Correctness of Algorithm 3. By analogy with Algorithm 2 it holds that cells of the table MinCost are never read without prior initialization.

We show that $\text{MinCost}(i, x) = y$ holds if and only if the minimum edge deletion budget required to increase the distance between the two terminals in the series-parallel graph $G_i = (V_i, E_i)$ represented by the subtree of $\mathcal{T}$ that is rooted at q_i to at least x is y. We do so for all $i \in \{1, \ldots, |Q|\}$ and $x \in \{1, \ldots, \ell\}$ by induction on i. Recall that for all $i \in \{1, \ldots, |Q|\}$, either the base case of $\alpha(q_i) = \text{🍃}$ or the inductive step case of $\delta_l(q_i), \delta_r(q_i) < i$ holds and consider the following case distinction.

Case 1: $\alpha(q_i) = \text{🍃}$. To obtain $G_i = (V_i, E_i)$, we set $V_i = \{u, v\}$ and $E_i = \{e\}$ with

Algorithm 3 Solve W-SP-MVE on series-parallel graphs for small ℓ.

Input: An instance $\mathcal{I} = (G = (V, E), k, \ell, s, t, \sigma, \tau)$ of W-SP-MVE such that G is a series-parallel graph with terminals s and t.

Ourput: *yes* if $\mathcal{I}$ is a *yes*-instance, *no* otherwise.

Find any sp-tree $\mathcal{T} := (T = (Q, R), \alpha, \beta, \delta_l, \delta_r, q_0)$ that represents G.

Let $(q_i)_{i \in \{1,\dots,|Q|\}}$ be the node sequence of a post-order traversal of T starting at q_0 and let $\overline{q} \in \{1, \dots, |Q|\}$ denote the position of q in the sequence for all $q \in Q$.

Fill a table $\text{MinCost} : \{1, \dots, |Q|\} \times \{0, \dots, \ell\} \to \mathbb{N}$ as follows.

for i **from** 1 **to** $|Q|$ **do**

  for all $x \in \{0, \dots, \ell\}$ **do**

$$\text{MinCost}(i, l) \leftarrow \begin{cases} \sigma(\widehat{\beta}(q_i)), & \alpha(q_i) = \text{🍃} \wedge \tau(\widehat{\beta}(q_i)) < x \\ 0, & \alpha(q_i) = \text{🍃} \wedge \tau(\widehat{\beta}(q_i)) \geqslant x \\ \text{OPT}_{\text{Ⓢ}}(i, x), & \alpha(q_i) = \text{Ⓢ} \\ \text{OPT}_{\text{Ⓟ}}(i, x), & \alpha(q_i) = \text{Ⓟ} \end{cases}$$

with

$$\text{OPT}_{\text{Ⓢ}}(i, x) := \min_{x' \in \{0,\dots,x\}} \left(\text{MinCost}(\overline{\delta_l(q_i)}, x') + \text{MinCost}(\overline{\delta_r(q_i)}, x - x')\right)$$

and

$$\text{OPT}_{\text{Ⓟ}}(i, x) := \text{MinCost}(\overline{\delta_l(q_i)}, x) + \text{MinCost}(\overline{\delta_r(q_i)}, x)$$

if $\text{MinCost}(|Q|, \ell) \leqslant k$ **then**

  return *yes*

else

  return *no*

$e := \{u, v\}$. It holds that $\mathrm{dist}_{G_i}(u, v) = \tau(e)$. If $\tau(e) < x$ holds for some $x \in \mathbb{N}$, then we need to remove at least e from E_i in order to increase the distance between u and v to at least x. From $\mathrm{dist}_{G_i - \{e\}}(u, v) = \infty > x$ it follows that we do not need to remove an additional edge. Thus the minimum budget spent in order to increase the distance between u and v to at least x is $\sigma(e) = \mathrm{MinCost}(i, x)$. If $\tau(e) \geqslant x$, then we do not need to remove any edge from E_i and the minimum budget spent is $0 = \mathrm{MinCost}(i, x)$. Hence, $\mathrm{MinCost}(i, x)$ is correct for all $x \in \{1, \ldots, \ell\}$.

Case 2: $\alpha(q_i) = \textcircled{S}$. Let $l, r \in \{1, \ldots, |Q|\}$, $u, v, w \in V$, $E_l, E_r \subseteq E_i$, and $G_i = (V_i, E_i)$ as in Case 2 of the correctness proof of Algorithm 2. Recall that $E_l \cup E_r = E_i$ and $l, r < i$ hold. Observe that any path from u to v in G_i first goes from u to w, using only edges in E_l and then from w to v, using only edges in E_r. Thus, increasing the distance between u and v in G_i to at least x is equivalent to increasing the sum of the distances between u and w and between w and v to at least x. By induction hypothesis, $\mathrm{MinCost}(l, x')$ contains the minimum cost to increase the length between u and w to at least x' and $\mathrm{MinCost}(r, x - x')$ contains the minimum cost to increase the length between w and v to at least $x - x'$ for all $x' \in \{0, \ldots, x\}$. It holds that these are all possibilities to achieve a sum of distances between u and w and between w and v of at least x. Hence, the minimum budget required to increase the distance between u and v to at least x is the smallest sum $\mathrm{MinCost}(l, x') + \mathrm{MinCost}(r, x - x')$ over all these possibilities represented by $x' \in \{0, \ldots, x\}$, as computed by the algorithm.

Case 3: $\alpha(q_i) = \textcircled{P}$. Let $l, r \in \{1, \ldots, |Q|\}$, $u, v \in V$, $E_l, E_r \subseteq E_i$, and $G_i = (V_i, E_i)$ as in Case 3 of the proof of correctness of Algorithm 2. Recall that $E_l \cup E_r = E_i$ and $l, r < i$ hold. Observe that all edges of any u-v-path in G_i are completely contained in either E_l or E_r. Hence, all minimal edge deletion sets that increase the length of any shortest u-v-path using edges in E_l do not increase the length of any shortest u-v-path using edges in E_r and vice versa. It follows that the minimum cost of increasing the distance between u and v to at least x is the sum of the minimum costs of increasing the distances between u and v in both $G_l = G_i[E_l]$ and $G_r = G_i[E_r]$ to at least x. By induction hypothesis, these minimum costs are given by $\mathrm{MinCost}(l, x)$ and $\mathrm{MinCost}(r, x)$ and it holds that the algorithm computes their sum.

It remains to be shown that $\mathcal{I}$ is a *yes*-instance if and only if $\mathrm{MinCost}(|Q|, \ell) \leqslant k$. By analogy with the proof of correctness of Algorithm 2 we can argue that $\mathrm{MinCost}(|Q|, \ell) \leqslant k$ is equivalent to the statement that the minimum edge deletion budget required to increase the distance between the two terminals s and t of G to at least ℓ is at most k.

Running time. Both algorithms first compute an sp-tree $\mathcal{T}$ of the input graph G. This, along with checking whether G is indeed a series-parallel graph, can be done in $\mathcal{O}(m)$ time [26]. If δ_l and δ_r are realized as pointers within the representation of $\mathcal{T}$, then they are also an appropriate representation of the tree structure T of $\mathcal{T}$ that allows post-order traversal in $\mathcal{O}(m)$ time, as T has exactly m leaves and thus at most $2m - 1$ nodes. During traversal of T, such representation also allows the children $\delta_l(q)$ and

$\delta_r(q)$ of the current node q to be accessed in $\mathcal{O}(1)$ time.

For our analysis, we make the simplifying assumption that, as a next step, T is traversed in $\mathcal{O}(m)$ time to build a lookup table that stores $\overline{q}$ for all $q \in Q$. Note that for practical means, both algorithms need to traverse T just once as this lookup table is only used for nodes that were already visited, that are children of the current node, and thus can be built on the fly.

The algorithms differ in the next step. Algorithm 2 traverses the nodes of T and iterates from 0 to k for each node, resulting in at most $(2m-1)(k+1)$ iteration steps in total. In each step, one cell of MaxLen is written. There are four cases that can be distinguished in constant time given that α, β, and τ are realized as a pointer within the representation of $\mathcal{T}$. The two cases with $\alpha(q_i) = \text{🍃}$ can then be handled in constant time. The cases of $\alpha(q_i) \in \{\text{Ⓢ}, \text{Ⓟ}\}$ both take a maximum over up to $k+1$ values, each of which can be computed via basic constant-time operations and a constant number of lookups of values of δ_l, δ_r, $\overline{\cdot}$ and MaxLen, each of which can be made in constant time. Hence, each iteration step can be computed in $\mathcal{O}(k)$ time. As there are $\mathcal{O}(mk)$ many steps, this yields a total running time of $\mathcal{O}(mk^2)$ for the loop beginning in line 5. By analogous arguments it can be shown that the corresponding loop of Algorithm 3 can be completed in $\mathcal{O}(m\ell^2)$ time given that α, β, and σ are realized as a pointer within the representation of $\mathcal{T}$.

Both algorithms last decide on their return value in constant time. We can conclude that Algorithm 2 runs in $\mathcal{O}(mk^2)$ time and Algorithm 3 runs in $\mathcal{O}(m\ell^2)$ time. ■

Note that the numeric parameters k and ℓ of a W-SP-MVE problem instance can be represented in $\mathcal{I}$ with a size that is logarithmic in their value. Hence, both algorithms above can be said to run in *pseudo-polynomial* time, that is their running time is polynomial if the value of the respective numeric parameter, that is k for Algorithm 2 and ℓ for Algorithm 3, is polynomially upper-bounded in the size of $\mathcal{I}$. Thus, these algorithms do not contradict the NP-hardness stated in Theorem 8.1. Furthermore, we can show that in the problem variants of U-SP-MVE, C-SP-MVE, and L-SP-MVE, either k or ℓ are indeed polynomially upper-bounded in the input size of any non-trivial instance, leading to the following observation.

Corollary 8.5. U-SP-MVE, C-SP-MVE, *and* L-SP-MVE *can be decided in* $\mathcal{O}(m^3)$ *time on series-parallel graphs with matching terminals.*

Proof. Let $G = (V, E)$ be a series-parallel graph with terminals s and t. Let $\mathcal{I}_{\text{L-SP-MVE}} = (G, k, \ell, s, t, \tau)$ be a problem instance of L-SP-MVE. First consider the case of $k \geqslant m$. As all edges have deletion cost one, $X := E$ is a solution to $\mathcal{I}_{\text{L-SP-MVE}}$ and we can decide it in linear time by reading the input to compute m and retrieve k. If $k < m$, then we can decide $\mathcal{I}_{\text{L-SP-MVE}}$ in $\mathcal{O}(m^2k) = \mathcal{O}(m^3)$ time using Algorithm 2.

Let $\mathcal{I}_{\text{C-SP-MVE}} = (G, k, \ell, s, t, \sigma)$ be a problem instance of C-SP-MVE. First consider the case of $\ell > m$. As all edges have length one, any solution to $\mathcal{I}_{\text{C-SP-MVE}}$ must be a cut.

As we can find a minimum cut in a cost-weighted graph in $\mathcal{O}(mn + n^2 \log n)$ time [25] and n is upper-bounded by $m + 1$ as there are no isolated vertices in a series-parallel graph, we can decide $\mathcal{I}_{\text{C-SP-MVE}}$ in $\mathcal{O}(m^3)$ time. If $\ell \leqslant m$, then we can decide $\mathcal{I}_{\text{C-SP-MVE}}$ in $\mathcal{O}(m^2\ell) = \mathcal{O}(m^3)$ time using Algorithm 3.

By any of the two arguments above it follows directly that we can also decide U-SP-MVE on series-parallel graphs in $\mathcal{O}(m^3)$ time. ■

In conclusion, we consider both algorithms feasible for practical application. On one hand they solve not just the decision problem but also the MaxLen and MinCost optimization variants, respectively, as defined in Section 2.3. As both algorithms are, in essence, dynamic programs, it is routine to trace back to the associated edge deletion set. On the other hand the fact that the algorithm traverses $\mathcal{T}$ in post-order and reads only table cells associated with children of the current tree node allows for parallelized computation of the table in a divide-and-conquer fashion. Herein the algorithm combines the advantages of both algorithmic archetypes; that are a time-memory trade-off as well as the potential of parallelization.

9 Conclusion

In this work we analyzed, with respect to a number of selected graph classes, the computational complexity of four natural variants of finding the most vital edges for shortest paths. Table 1 on page 9 summarizes our results for all combinations of graph classes and problem variants. In particular, we found that the variants that assign a length, a deletion cost, or both to edges of the graph (that are L-SP-MVE, C-SP-MVE, and W-SP-MVE, respectively) are harder to solve than the variant where all edges are uniform (U-SP-MVE), even if the input graph has a strongly restricted structure. This observation is most apparent in the case of complete and threshold graphs, on which U-SP-MVE can be decided in linear time while the other variants remain NP-hard.

By showing that U-SP-MVE remains NP-complete on split graphs, a superclass of the threshold graphs, we could identify a complexity gap within the realm of graph classes, too. It remains an open question whether U-SP-MVE is polynomial-time decidable on cographs, since the threshold graphs are exactly the split graphs that are also cographs [6].

Another open question—the central one of this document—is marked by Conjecture 6.1. A proof of correctness would yield that U-SP-MVE is polynomial-time solvable on proper interval graphs with certain terminal vertices, as shown in Section 6.

In addition to edge weights we also looked into handling the parallel edges that appear in multigraphs. In Section 7 we give two linear-time reductions that may prove viable in scenarios where the input graph is a multigraph.

Finally, we could extend previous work by Baier et al. [2], who prove that W-SP-MVE is NP-complete on series-parallel graphs, by providing two algorithms that decide

the same combination of problem variant and graph class in pseudo-polynomial time. For all other problem variants that we examined, atleast one of the algorithms runs in polynomial time. This marks another gap in complexity among the problem variants.

Apart from the open problems mentioned above, variants of Shortest Path Most Vital Edges as well as their vertex-deletion counterparts deserve further study from a number of perspectives. While we are among the first to look into restricted graph classes in particular, there have also been few publications that focus on the parameterized complexity [5, 11, 15] or take the perspective of approximation [2, 5].

As for problem variants as such, a natural next step would be to further bridge the gap between shortest path and minimum spanning tree scenarios via a multi-terminal approach [11]. It would also be interesting to survey the civil applications of edge and vertex deletion problems, such as preventing hospital-acquired infections [1], so that also the variants with practical utility (we may, of course, refer to those as the *most vital variants*) can be identified and find particular attention in future work.

References

[1] N. Assimakopoulos. „A network interdiction model for hospital infection control“. In: *Computers in biology and medicine* 17.6 (1987), pp. 413–422.

[2] G. Baier, T. Erlebach, A. Hall, E. Köhler, P. Kolman, O. Pangrác, H. Schilling, and M. Skutella. „Length-bounded cuts and flows“. In: *ACM Transactions on Algorithms (TALG)* 7.1 (2010).

[3] M. O. Ball, B. L. Golden, and R. V. Vohra. „Finding the most vital arcs in a network“. In: *Operations Research Letters* 8.2 (1989), pp. 73–76.

[4] A. Bar-Noy, S. Khuller, and B. Schieber. *The complexity of finding most vital arcs and nodes*. Tech. rep. Maryland: University of Maryland, 1995.

[5] C. Bazgan, A. Nichterlein, and R. Niedermeier. „A Refined Complexity Analysis of Finding the Most Vital Edges for Undirected Shortest Paths“. In: *Proceedings of the 9th International Conference on Algorithms and Complexity (CIAC'15)*. Vol. 9079. LNCS. Springer, 2015, pp. 47–60.

[6] A. Brandstädt, V. B. Le, and J. P. Spinrad. *Graph classes: a survey*. Philadelphia: Society for Industrial and Applied Mathematics, 1999.

[7] V. Chvátal and P. L. Hammer. „Aggregation of inequalities in integer programming“. In: *Annals of Discrete Mathematics* 1 (1977), pp. 145–162.

[8] H. W. Corley and H. Chang. „Finding the n Most Vital Nodes in a Flow Network“. In: *Management Science* 21.3 (1974), pp. 362–364.

[9] H. W. Corley and D. Y. Sha. „Most vital links and nodes in weighted networks“. In: *Operations Research Letters* 1.4 (1982), pp. 157–160.

[10] D. G. Corneil, H. Kim, S. Natarajan, S. Olariu, and A. P. Sprague. „Simple Linear Time Recognition of Unit Interval Graphs“. In: *Information Processing Letters* 55.2 (1995), pp. 99–104.

[11] P. Dvořák and D. Knop. „Parametrized Complexity of Length-Bounded Cuts and Multi-cuts“. In: *Theory and Applications of Models of Computation*. Ed. by R. Jain, S. Jain, and F. Stephan. Springer, 2015, pp. 441–452.

[12] C. M. de Figueiredo, J. Meidanis, and C. P. de Mello. „A linear-time algorithm for proper interval graph recognition“. In: *Information Processing Letters* 56.3 (1995), pp. 179–184.

[13] B. L. E. de Fluiter. „Algorithms for Graphs of Small Treewidth“. PhD thesis. University of Utrecht, 1997.

[14] M. L. Fredman and R. E. Tarjan. „Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms“. In: *Journal of the ACM* 34.3 (1987), pp. 596–615.

[15] P. A. Golovach and D. M. Thilikos. „Paths of bounded length and their cuts: Parameterized complexity and algorithms“. In: *Discrete Optimization* 8.1 (2011), pp. 72–86.

[16] J. Guo and Y. R. Shrestha. „Parameterized complexity of edge interdiction problems“. In: *Computing and Combinatorics*. Ed. by Z. Cai, A. Zelikovsky, and A. Bourgeois. Springer, 2014, pp. 166–178.

[17] R. M. Karp. „Reducibility among Combinatorial Problems“. In: *Complexity of Computer Computations*. Springer, 1972.

[18] N. V. Mahadev and U. N. Peled. *Threshold graphs and related topics*. Vol. 56. Annals of Discrete Mathematics. Elsevier, 1995.

[19] K. Malik, A. Mittal, and S. K. Gupta. „The k most vital arcs in the shortest path problem“. In: *Operations Research Letters* 8.4 (1989), pp. 223–227.

[20] E. Nardelli, G. Proietti, and P. Widmayer. „Finding the most vital node of a shortest path“. In: *Theoretical Computer Science* 296.1 (2003), pp. 167–177.

[21] M. T. Omran, J.-R. Sack, and H. Zarrabi-Zadeh. „Finding paths with minimum shared edges“. In: *Journal of Combinatorial Optimization* 26.4 (2013), pp. 709–722.

[22] H. D. Ratliff, G. T. Sicilia, and S. H. Lubore. „Finding the n Most Vital Links in Flow Networks“. In: *Management Science* 21.5 (1975), pp. 531–539.

[23] F. S. Roberts. „Indifference Graphs“. In: *Proof techniques in graph theory*. Ed. by F. Harary. New York: Academic Press, 1969, pp. 139–146.

[24] H. Shen. „Finding the k Most Vital Edges with Respect to Minimum Spanning Tree“. In: *Acta Informatica* 36.5 (1999), pp. 405–424.

[25] M. Stoer and F. Wagner. „A Simple Min-Cut Algorithm“. In: *Journal of the ACM* 44.4 (1997), pp. 585–591.

[26] J. Valdes, R. E. Tarjan, and E. L. Lawler. „The recognition of Series Parallel digraphs“. In: *Proceedings of the eleventh annual ACM symposium on Theory of computing*. Ed. by M. J. Fischer, R. A. DeMillo, N. A. Lynch, W. A. Burkhard, and A. V. Aho. ACM, 1979, pp. 1–12.

[27] A. Washburn and K. Wood. „Two-Person Zero-Sum Games for Network Interdiction“. In: *Operations Research* 43.2 (1995), pp. 243–251.