Technische Universität Berlin

The Parameterized Complexity of Finding Paths with Shared Edges

Masterarbeit

von Till Fluschnik

zur Erlangung des Grades „Master of Science“ (M. Sc.) im Studiengang Wirtschaftsmathematik

Erstgutachter:	Prof. Dr. Rolf Niedermeier
Zweitgutachter:	Prof. Dr. Martin Skutella

Berlin, den 10.04.2015

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.

Die selbstständige und eigenhändige Ausfertigung versichert an Eides statt

Berlin, den ______________________ ________________________________

Datum Unterschrift

Zusammenfassung

In dieser Arbeit studieren wir das sogenannte Minimum Shared Edges-Problem auf ungerichteten Graphen: Gegeben ist ein ungerichteter Graph $G = (V, E)$, zwei spezielle Knoten $s, t \in V$ und zwei natürliche Zahlen $p \geq 1$, $k \geq 0$. Die Frage ist, ob es p s-t Pfade in G gibt, die höchstens k Kanten teilen, wobei eine Kante geteilt heißt, wenn sie in mindestens zwei s-t Pfaden vorkommt. Das Problem findet zum Beispiel eine Anwendung in dem folgenden Kontext. Wir wollen eine wichtige Person, auch VIP genannt, von einem Anfangsort zu einem Zielort befördern. Dabei besteht die Gefahr, dass ein Attentat auf den VIP ausgeübt werden könnte. Daher entschließen wir uns dafür, mehrere Konvois vom Anfangsort zum Zielort zu entsenden, wobei nur einer der Konvois den VIP befördert, um die Erfolgswahrscheinlichkeit eines Anschlags zu reduzieren. Bei dem Koordinieren der Konvois beachten wir, dass keine Straße oder Ähnliches von mehr als einem Konvoi genutzt wird, da diese Straßen für einen Anschlag bevorzugt werden könnten. Ist es jedoch nicht zu vermeiden, dass eine Straße von mindestens zwei Konvois genutzt werden muss, so müssen wir erhöhte Sicherheitsmaßnahmen für diese Straße durchführen. Wir fragen uns demnach, gegeben eine Anzahl von Konvois, was ist die kleinstmögliche Anzahl von Straßen, die von mindestens zwei Konvois gemeinsam genutzt werden müssen.

Wir studieren in dieser Arbeit die Komplexität des Minimum Shared Edges-Problems. Wir zeigen, dass die Entscheidungsvariante des Problems NP-vollständig ist, auch dann, wenn der Maximalgrad des zugrundeliegenden Graphen durch fünf beschränkt ist. Darüber hinaus zeigen wir, dass das Problem $W[2]$-schwer bezüglich der Anzahl der geteilten Kanten ist. Wir zeigen zudem, dass das Problem *fixed-parameter tractable* bezüglich der Anzahl der Pfade ist. Für diesen Zweck verwenden wir die sogenannte Treewidth Reduction Technique um den initialen Graphen zu modifizieren und führen anschließend ein Dynamisches Programm aus, das das Problem bei gegebener Baumzerlegung des Graphen löst. Wir formulieren ein solches Dynamisches Programm und beweisen seine Korrektheit. Darüber hinaus präsentieren wir einen Algorithmus, der das Minimum Shared Edges-Problem effizient löst für kleine Werte für die Anzahl von Kanten, die geteilt werden dürfen.

Zudem stellen wir eine Variation des Problems vor, die zusätzlich eine obere Schranke für die Länge der Pfade fordert. Wir nennen das Problem das Short Minimum Shared Edges-Problem. Wir zeigen auf, dass das Problem $W[2]$-schwer bezüglich der Anzahl der geteilten Kanten und der oberen Schranke für die Länge der Pfade ist. Weiter geben wir eine Modifikation unseres Dynamischen Programms an, die das Short Minimum Shared Edges-Problem löst. Mithilfe des modifizierten Dynamischen Programms zeigen wir, dass das Problem auf planaren Graphen *fixed-parameter tractable* bezüglich der Anzahl der Pfade und der oberen Schranke für die Länge der Pfade ist.

Summary

In this work, we study the so-called Minimum Shared Edges problem on undirected graphs. Given an undirected graph $G = (V, E)$, two vertices $s, t \in V$, and two integers $p \geq 1$ and $k \geq 0$, the question is whether there are p s-t paths in G that share at most k edges, where an edge is shared if it appears in at least two s-t paths. We show that the problem is NP-complete, and that it remains NP-complete on graphs of maximum degree five. Moreover, we show that the problem is $W[2]$-hard when parameterized by the number k of shared edges and that it is fixed-parameter tractable when parameterized by the number p of paths. We provide an FPT algorithm with respect to the number k of shared edges and the number p of paths that solves the Minimum Shared Edges problem in $(p-1)^k \cdot O(|G|^2)$ time. Moreover, we provide a dynamic program that, given a tree decomposition of the input graph, solves the problem in FPT-time with respect to the number p of paths and the width of the tree decomposition. We introduce and study a variation of the Minimum Shared Edges problem, where the length of the p s-t paths is upper-bounded by an integer λ. We denote this problem by the Short Minimum Shared Edges problem. We show that this problem is $W[2]$-hard with respect to the number k of shared edges and the upper bound λ. Further, we show that our dynamic program can be adapted to solve the Short Minimum Shared Edges problem in FPT-time with respect to the number p of paths, the upper bound λ, and the width of the given tree decomposition of the input graph. Upon this, we show that the problem is fixed-parameter tractable on planar graphs when parameterized by the number p of paths and the upper bound λ.

Contents

1 Introduction

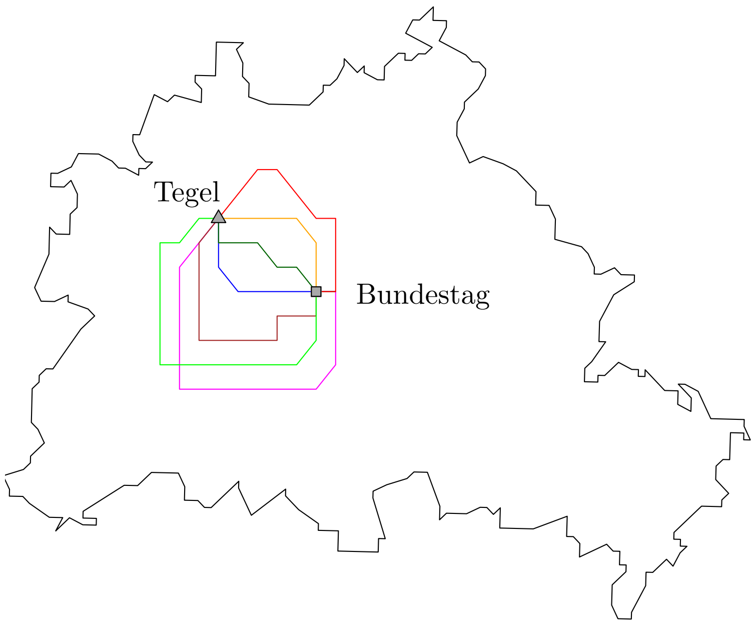

Figure 1.1: A sketch of the routes of the convoys. The shapefile of Berlin is derived from a shapefile provided by `http://www.gadm.org/`.

Berlin. The president of the United States, Barack Obama, will come to Berlin to visit our federal chancellor Angela Merkel. We are the head of a security agency, and we are asked by the United States Secret Service (USSS) to route Mr. Barack Obama from the airport Berlin-Tegel (TXL) to the German Bundestag, close to Berlin's Brandenburger Tor, where he and Angela Merkel planned to meet. The U.S. president is under threat of attack and, therefore, in need of a high security level. We decide to send some convoys through the city as sketched in Figure 1.1, where one of the convoys carries the president. No attacker can know in which convoy the president will be present. When we route the convoys through the city, we want to avoid that any two or more convoys share a part of their routes, for example using the same street, since any part that is shared by at least two convoys could imply a higher potential of any attack. If we cannot avoid that two or more convoys share a part of their routes, we need to install special units to increase the security level of that part. These special units provoke costs and we want to avoid any additional costs. Therefore, our main task is to route a number of convoys in such a way that the number of shared parts is minimized.

In this work, we consider the following decision problem:

> **Problem**: MINIMUM SHARED EDGES (MSE)
> **Input**: Graph $G = (V, E)$, $s, t \in V$, $p \in \mathbb{N}$, and $k \in \mathbb{N}_0$.
> **Parameter**: p, k.
> **Question**: Are there p s-t paths in G that share at most k edges?

We say that an edge is *shared*, if the edge appears in at least two s-t paths. The problem was first introduced by Omran et al. [16] on directed graphs. Although graph G can be undirected as well as directed for MINIMUM SHARED EDGES, throughout this work, we consider MINIMUM SHARED EDGES on simple, undirected graphs, where simple means without multiple edges or loops. We say that p is the number of paths, and that k is the number of shared edges or edges that are allowed to be shared.

According to our introductory example in Berlin, the graph G represents the street network of Berlin, vertex s is the Tegel Airport and vertex t is the German Bundestag, p is the number of convoys we want to send and k is the number of parts the convoys are allowed to share, where the cost of installing special units for all the parts is one everywhere. We say that vertex s is the source (or the source vertex) and that vertex t is the sink (or the sink vertex). We remark that we call this problem also the VIP-routing problem.

If k is equal to zero, then the problem is equivalent to the Disjoint Paths problem with parameter p. Given a graph G, two vertices $s, t \in V(G)$ and $p \in \mathbb{N}$, the Disjoint Paths problem asks whether there are p pairwise edge-disjoint s-t paths in G. We remark that this problem can be solved in polynomial time using flow techniques [11].

A generalization of MINIMUM SHARED EDGES is the following minimization problem:

> **Problem**: MINIMUM VULNERABILITY (MV)
> **Input**: Graph $G = (V, E, c, u)$, $s, t \in V$, edge costs $c : E \to \mathbb{R}_{\geq 0}$, edge capacities $u : E \to \mathbb{R}_{\geq 0}$, and $r, p \in \mathbb{N}$.
> **Parameter**: r, p.
> **Task**: Find p s-t paths in G in such a way that the total cost of edges that are used in more than r of the p s-t paths is minimized?

We remark that for $r = 1$, edge costs equal to one, and edge capacities equal to the number p of paths, MINIMUM VULNERABILITY is equivalent to the minimization version of MINIMUM SHARED EDGES, where the minimization version of MINIMUM SHARED EDGES asks for a set of p s-t paths in a graph G with $s, t \in V(G)$ such that the number of shared edges is minimized. Thus, MINIMUM VULNERABILITY generalizes MINIMUM SHARED EDGES in this sense.

Related Work. Omran et al. [16] introduced and studied the minimization version of MINIMUM SHARED EDGES on directed graphs. That is, given a directed graph D, two vertices $s, t \in V(D)$ and a number p of paths, the problem asks for a set of p s-t paths such that the number of shared edges is minimized. We point out that in our work, we consider Minimum Shared Edges as decision version on undirected graphs. Omran et al. showed that the minimization version of Minimum Shared Edges is NP-hard, by proving that Minimum Shared Edges on directed graphs is NP-complete, using a reduction from the Set Cover problem. Their reduction is a parameterized reduction with respect to the number k of shared edges, and thus, they implicitly showed that MSE(k) on directed graphs is $W[2]$-hard, where MSE(k) denotes MSE parameterized by the number k of shared edges. According to the number p of paths, they presented a $(p-1)$-approximation algorithm for the minimization version of Minimum Shared Edges on directed graphs. In addition, they proved an inapproxibility result for the minimization version of Minimum Shared Edges on directed graphs within a factor of $2^{\log^{1-\epsilon}(n)}$, for any constant $\epsilon > 0$, where n is the number of vertices in the given graph. Finally, they discussed some heuristics for Minimum Shared Edges on directed graphs and presented some experimental results, where one of their examples is the road network of Rome.

Ye et al. [19] studied the minimization version of Minimum Shared Edges on simple, undirected graphs. They showed that Minimum Shared Edges can be solved in polynomial time on graphs with bounded treewidth. They showed that given a graph G, a tree decomposition of G of width at most ω, two vertices $s, t \in V(G)$ and $p \in \mathbb{N}$, the minimum number of shared edges by p s-t paths can be computed in $O(|V(G)| \cdot (p+1)^{2^{\omega\cdot(\omega+1)/2}} + |V(G)| \cdot (p+1)^{(\omega+4)^{2\cdot\omega+8}})$ time. As a consequence, they showed on the one hand that MSE(p, ω) is fixed-parameter tractable when parameterized by the number p of paths and an upper bound ω on the treewidth of the input graph, and on the other hand that MSE(ω) is in XP.

Assadi et al. [2] introduced and studied Minimum Vulnerability on directed graphs as a generalization of Minimum Shared Edges on directed graphs. They provided a $\lfloor \frac{p}{r+1} \rfloor$-approximation algorithm for Minimum Vulnerability on directed graphs using a primal-dual approach, which implicates a $\lfloor p/2 \rfloor$-approximation algorithm for Minimum Shared Edges on directed graphs. This result improves the $(p-1)$-approximation algorithm due to Omran et al. [16]. In addition, they presented an approximation algorithm for Minimum Shared Edges on directed graphs with an approximation guarantee of $O(|V(G)|^{3/4})$, where G is the given directed graph. Further, they showed that Minimum Vulnerability is in XP when parameterized by the number p of paths.

Aoki et al. [1] studied Minimum Vulnerability on undirected graphs. They showed that Minimum Vulnerability on undirected graphs is NP-hard, and even NP-hard on undirected bipartite series-parallel graphs and undirected threshold graphs. They showed that Minimum Vulnerability on undirected graphs can be solved in polynomial time on graphs with bounded treewidth. In comparison to the algorithm provided by Ye et al. [19], they provided an algorithm that, given an undirected graph G, two vertices $s, t \in V(G)$, an integer $p \in \mathbb{N}$ and a bound ω on the treewidth of graph G, computes the minimum number of shared edges for p s-t paths in $(p+1)^{O(\omega^{\omega+1})} \cdot |V(G)|$ time. In addition, they showed that $\mathrm{MV}(p)$ is fixed-parameter tractable on chordal graphs when parameterized by the number p of paths.

Considering the introductory example, we may want to restrict the length of each path of the convoys to an upper bound, for example, if there is a per meter cost of each convoy. This motivates us to consider the following decision problem.

Problem: Short Minimum Shared Edges (SMSE)
Input: Graph $G = (V, E)$, $s, t \in V$, $p \in \mathbb{N}$, $k \in \mathbb{N}_0$, and $\lambda \in \mathbb{N}$.
Parameter: p, k, λ.
Question: Are there p s-t-paths of length at most λ in G that share at most k edges?

We remark that Short Minimum Shared Edges reduces to Minimum Shared Edges in the case that λ is at least the number of edges in the graph, since then every s-t path has length at most λ and the question remains, whether there are p s-t paths that share at most k edges. We study Short Minimum Shared Edges beside Minimum Shared Edges, but our main focus in this work is on MSE.

Our Contributions. In this work, we obtain the following results for Minimum Shared Edges.

- In Section 3, we present an algorithm that solves an instance (G, s, t, p, k) of Minimum Shared Edges in $(p-1)^k \cdot O(|G|^2)$ time. The algorithm implies that $\mathrm{MSE}(p, k)$ is fixed-parameter tractable (Theorem 3.10), and $\mathrm{MSE}(k)$ is in XP. Moreover, if k is a constant, then Minimum Shared Edges can be solved in polynomial time. For small values of k, relative to p and the size $|G|$ of the graph G, this algorithm is of potentially practical interest.

- In Section 4, we show that on the unbounded, undirected $\mathbb{Z} \times \mathbb{Z}$-grid graph $\mathbb{G}$ with $s, t \in V(\mathbb{G})$, any instance $(\mathbb{G}, s, t, p, k)$ of Minimum Shared Edges can be solved in constant time (Theorem 4.1). For this purpose, for every instance $(\mathbb{G}, s, t, p, k)$ we provide a construction of $p = 4 + 2 \cdot \lfloor k/2 \rfloor$ s-t paths that share at most k edges.

- In Section 5, we prove that $\mathrm{MSE}(k)$ is $W[2]$-hard (Theorem 5.1) by giving a reduction from the Set Cover problem. Further, we show that Minimum Shared Edges is NP-complete and we prove that MSE remains NP-hard even on graphs

Parameter	Complexity	Remark
p	FPT	Theorem 7.1
k	XP, $W[2]$-hard	Theorem 5.1
Δ	NP-hard for $\Delta \geq 5$	Theorem 5.2
d	XP	$k < d$, see Corollary 3.2, and Algorithm 3.1
tw	XP	Ye et al. [19], Aoki et al. [1]
(p, k)	FPT	Theorem 3.10, Algorithm 3.1
(p, d)	FPT	$k < d$, see Corollary 3.2
(k, Δ)	FPT	Corollary 3.11
(p, tw)	FPT	Theorem 6.1

Table 1.1: Overview of the results for Minimum Shared Edges on undirected graphs.

with maximum degree at least five (Theorem 5.2) by giving a reduction from the Vertex Cover problem.

- In Section 6, we present a dynamic program on a tree decomposition of width ω that solves MSE(p, ω) in FPT-time (Theorem 6.1). More precisely, if graph G is given together with a tree decomposition of width ω, our dynamic program solves MSE(p, ω) in $O(p \cdot (\omega + 4)^{3 \cdot p \cdot (\omega+3)+4} \cdot |V(G)|)$ time. Though Ye et al. [19] and Aoki et al. [1] already provided dynamic programs on a tree decomposition of width ω solving MSE(p, ω) in FPT-time, we present our dynamic program because its running time is, in contrast, not double exponentially in any of the two parameters p and ω, and it allows an adaption for Short Minimum Shared Edges.

- In Section 7, we prove the main result of our work. We prove that MSE(p) is fixed-parameter tractable (Theorem 7.1) when parameterized by the number p of paths. For this purpose, we make use of the treewidth reduction technique due to Marx et al. [14] to modify the input graph in such a way that the treewidth of the modified graph is upper-bounded by a function only depending on p. Then, we make use of the fact that MSE(p, ω) is fixed-parameter tractable.

Our main results are summarized in Table 1.1 and in Figure 1.2. We remark that in Table 1.1 and in Figure 1.2, we denote by d the diameter of the input graph. In Figure 1.2, we show a Hasse diagram of the parameter space of Minimum Shared Edges. Each node in the diagram represents Minimum Shared Edges when parameterized by the label of the node. Each node is additionally labeled by the complexity of the represented problem. Two nodes in the diagram are connected, if they include the same parameter or if any parameter in the lower node can be upper-bounded by at least one

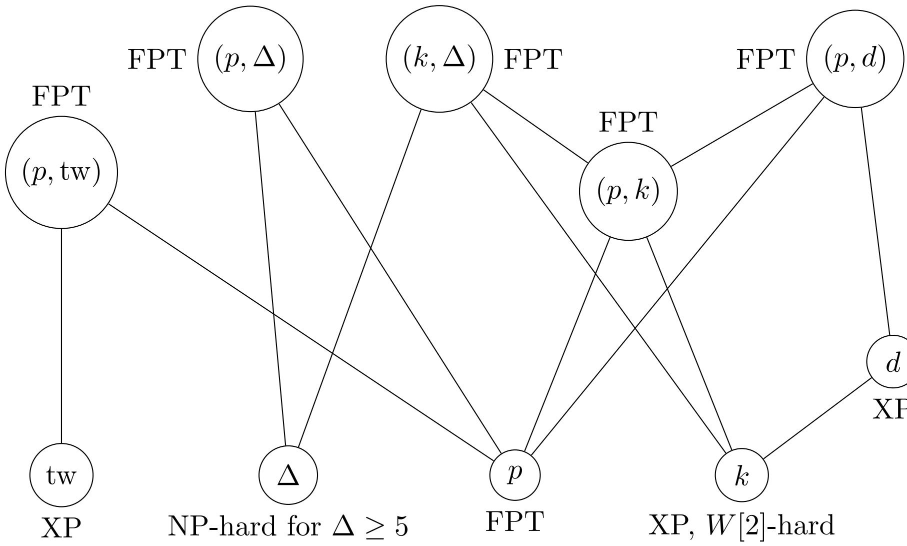

Figure 1.2: Hasse diagram of the parameter space for MINIMUM SHARED EDGES.

parameter in the upper node.

Our contributions according to SHORT MINIMUM SHARED EDGES are the following. We show in Section 5 that SMSE(k, λ) is $W[2]$-hard (Theorem 5.3) by giving a reduction from the SET COVER problem. Further, we present in Section 6 a modification of our dynamic program such that SMSE(p, λ, ω) can be solved in FPT-time (Theorem 6.2). Moreover, we show that SMSE(p, λ) is fixed-parameter tractable on planar graphs (Theorem 6.3).

2 Preliminaries

As a convention, by $\mathbb{N}$ we denote the natural numbers without zero, and by $\mathbb{N}_0$ the natural numbers containing zero, i.e. $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. For every $\ell \in \mathbb{N}$, we define $[\ell] := \{1, \ldots, \ell\} \subseteq \mathbb{N}$ as the set of positive integers at most ℓ. We remark that our definitions of s-t flows and tree decompositions differ from the standard.

Graph Theory. Let $G = (V, E)$ be an undirected graph. We write $V(G)$ for the vertex set of graph G and $E(G)$ for the edge set of graph G. We define the size of graph G as $|G| := |V(G)| + |E(G)|$. For a vertex set $W \subseteq V(G)$, we denote by $G[W]$ the subgraph of G with vertex set $\{v \in V(G) \mid v \in W\}$ and edge set $\{\{v, w\} \in E(G) \mid v, w \in W\}$. We say that $G[W]$ is the subgraph of G *induced* by the vertex set W. For an edge set $F \subseteq E(G)$, we denote by $G[F]$ the subgraph of G with vertex set $\{v \in V(G) \mid (e \in F) \wedge (v \in e)\}$ and edge set $\{e \in E(G) \mid e \in F\}$. We say that $G[F]$ is the subgraph of G *induced* by the edge set F. For an edge $e \in E$, we denote by $G/\{e\}$ the contraction of edge e in G, and we denote by $G\backslash\{e\}$ the deletion of edge e in G (we write G/e and $G\backslash e$ for short). Consequently, for a set of edges $F \subseteq E$ we write G/F and $G\backslash F$ for the contraction and the deletion of the edges in F, respectively. We remark that G/F is well-defined, since edge contraction is commutative [18]. For graph parameters like the maximum degree or the diameter we write $\Delta(G)$ to denote the maximum degree of graph G and $\mathrm{diam}(G)$ to denote the diameter of G.

Let G be an undirected, connected graph. A *cut* $C \subseteq E$ is a set of edges such that the graph $G\backslash C$ is not connected. Let $s, t \in V(G)$ be two vertices in G. An *s-t cut* C is a cut such that vertices s and t are not connected in $G\backslash C$. A *minimum s-t cut* is an s-t cut C such that $|C| = \min |C'|$, where the minimum is taken over all s-t cuts C' in G. An s-t cut C in G is *minimal* if for all edges $e \in C$ it holds that $C\backslash\{e\}$ is not an s-t cut in G.

For a vertex set $S \subseteq V(G)$, we write $G - S$ for the graph $G[V/S]$. A vertex set $S \subseteq V$ is a *separator* in G if the graph $G - S$ is not connected. An *s-t separator* in G is a vertex set $S \subseteq V\backslash\{s, t\}$ such that s and t are not connected in $G - S$. An s-t separator S is a *minimum s-t separator* if there is no s-t separator S' in G with $|S'| < |S|$. An s-t separator S is a *minimal s-t separator* if for all $v \in S$ holds that $S\backslash\{v\}$ is not an s-t separator.

A *path* is a connected graph with exactly two vertices of degree one and no vertex of degree at least three. We call the vertices with degree one the *endpoints* of the path. The *length* of a path is defined as the number of edges in the path. For two distinct vertices $s, t \in V(G)$, we call the path with endpoints s and t as subgraph of G an s-t path in G. An s-t path in G is a *shortest s-t path* in G, if there is no s-t path in G of smaller length. We denote by $\mathrm{dist}_G(s, t)$ the length of a shortest s-t path in G.

A graph G has *edge capacities* if there is a function $c : E(G) \to \mathbb{R}_{\geq 0}$ that maps each edge in G to a number in $\mathbb{R}_{\geq 0}$, where $\mathbb{R}_{\geq 0}$ denotes the non-negative real numbers. For an edge $e \in E(G)$, we say that $c(e)$ is the *capacity* of edge e in G. We say that graph G has *unit edge capacities* if $c(e) = 1$ for all $e \in E(G)$. In this work, if we consider a graph with edge capacities, then we always consider a graph with unit edge capacities.

Let D be an directed graph with edge capacities $c : E(D) \to \mathbb{R}_{\geq 0}$ and let $s, t \in V(D)$ be two vertices in D. An *s-t flow* in D is a function $f : E(D) \to \mathbb{R}_{\geq 0}$ such that

(i) $f(e) \leq c(e)$ for all $e \in E(D)$,

(ii) $\sum_{w \in V(D):(v,w) \in E(D)} f((v, w)) = \sum_{w \in V(D):(w,v) \in E(D)} f((w, v))$ for all $v \in V(D) \backslash \{s, t\}$, and

(iii) $\sum_{w \in V(D):(w,t) \in E(D)} f((w, t)) - \sum_{w \in V(D):(t,w) \in E(D)} f((t, w)) \geq 0$.

The *value* of an s-t flow f in D is defined as $|f| := \sum_{w \in V(D):(w,t) \in E(D)} f((w, t)) - \sum_{w \in V(D):(t,w) \in E(D)} f((t, w))$. An s-t flow f is a *maximum s-t flow* in D if there is no s-t flow f' in D with $|f'| > |f|$.

For an undirected graph G we call the directed graph D_G the directed version of graph G if $V(D_G) = V(G)$ and $E(D_G) = \{(u, v), (v, u) \mid \{u, v\} \in E(G)\}$. If G has edge capacities $c : E(G) \to \mathbb{R}_{\geq 0}$, then D_G has edge capacities $c' : E(D_G) \to \mathbb{R}_{\geq 0}$ with $c'((u, v)) := c'((v, u)) := c(\{u, v\})$ for all edges $\{u, v\} \in E(G)$. We say that a function $f : E(G) \to \mathbb{R}_{\geq 0}$ is an s-t flow with value $|f| := \sum_{w \in V(G):\{w,t\} \in E(G)} f(\{w, t\})$ in an undirected graph G with edge capacities $c : E(G) \to \mathbb{R}_{\geq 0}$ and $s, t \in V(G)$, if there is an s-t flow f' in D_G such that $|f'| = |f|$, and for all edges $\{u, v\} \in E(G)$ it holds that $f'((u, v)) = 0$ and $f'((v, u)) = f(\{u, v\})$, or $f'((v, u)) = 0$ and $f'((u, v)) = f(\{u, v\})$. An s-t flow f_1 is a maximum s-t flow in G if there is no s-t flow f_2 in G with $|f_2| > |f_1|$. We remark that our definition of s-t flows on undirected graphs is close to the definition given by Goldberg and Rao [10]. For more information on flows, in particular on integral flows, the max-flow min-cut theorem, and Menger's theorem, we refer to the work of Kleinberg and Tardos [11].

Let $v \in V$ be a vertex in G. The *open neighborhood* $N_G(v)$ of v in G is the set of vertices that are connected with v by an edge, i.e. $N_G(v) := \{w \in V(G) \mid \{v, w\} \in E(G)\}$. The *closed neighborhood* $N_G[v]$ of v in G is defined as $N_G(v) \cup \{v\}$. For a vertex set $W \subseteq V(G)$, we define the open neighborhood of W as $N_G(W) := \bigcup_{v \in W} (N_G(v) \backslash W)$ and the closed neighborhood of W as $N_G[W] := W \cup N_G(W)$.

Let $G = (V, E)$ be a simple, undirected graph. A *tree decomposition* of graph G is a tuple $\mathbb{T} := (T, (B_\alpha)_{\alpha \in V(T)})$ of a tree T and family $(B_\alpha)_{\alpha \in V(T)}$ of sets $B_\alpha \subseteq V(G)$ such that

(i) for every edge $e \in E(G)$ there exists an $\alpha \in V(T)$ such that $e \subseteq B_\alpha$ and

(ii) for each $v \in V(G)$, the graph induced by the node set $\{\alpha \in V(T) \mid v \in B_\alpha\}$ is a tree.

The *width* ω of a tree decomposition $\mathbb{T}$ of graph G is defined as $\omega(\mathbb{T}) := \max\{|B_\alpha| - 1 \mid \alpha \in V(T)\}$. The *treewidth* $\mathrm{tw}(G)$ of graph G is defined as the minimum width over all tree decompositions of G, i.e. $\mathrm{tw}(G) := \min\{\omega(\mathbb{T}) \mid \mathbb{T}$ is a tree decompositon of $G\}$. We remark that if an upper bound on the treewidth of graph G is given, then a tree decomposition for graph G can be computed in linear time [4]. A tree decomposition $\mathbb{T} = (T, (B_\alpha)_{\alpha \in V(T)})$ is a *nice tree decomposition* if (i) tree T is rooted and binary, and (ii) each node $\alpha \in V(T)$ is of one of the following types:

- leaf node: α is a leaf of T and $B_\alpha = \emptyset$;
- introduce vertex node: α is an inner node of T with exactly one child node $\beta \in V(T)$ such that $B_\beta \subseteq B_\alpha$ and $|B_\alpha \backslash B_\beta| = 1$;
- forget node: α is an inner node of T with exactly one child node $\beta \in V(T)$ such that $B_\alpha \subseteq B_\beta$ and $|B_\beta \backslash B_\alpha| = 1$;
- join node: α is an inner node of T with exactly two child nodes $\beta, \gamma \in V(T)$ such that $B_\alpha = B_\beta = B_\gamma$.

We assume that the number of nodes in a nice tree decomposition of width ω of graph G is in $O(\omega \cdot |V(G)|)$, which follows from Kloks [12]. For more about nice tree decompositions, we refer to the work of Kloks [12].

A tree decomposition $\mathbb{T}$ for graph G is a *tree decomposition with introduce edge nodes* if for all edges in $E(G)$ there is exactly one introduce edge node in $\mathbb{T}$, where an introduce edge node is a node α in the tree decomposition $\mathbb{T}$ of G labeled with an edge $\{v, w\} \in E(G)$ with $v, w \in B_\alpha$ that has exactly one child node α' such that $B_\alpha = B_{\alpha'}$. The number of introduce edge nodes is equal to $|E(G)|$. Given a tree decomposition of G of width ω, the number of edges of G is at most $\omega \cdot |V(G)|$, which follows from Kloks [12]. Thus, we can assume that the number of nodes in a tree decomposition with introduce edge nodes is in $O(\omega \cdot |V(G)|)$. For more about tree decompositions with introduce edge nodes, we refer to the work of Cygan et al. [5].

Parameterized Complexity. An algorithm with running time $f(\ell) \cdot n^{O(1)}$, where n denotes the size of the input, ℓ is part of the input and f is a computable function only depending on ℓ, is called fixed-parameter tractable algorithm, or FPT algorithm. Equivalently, for an algorithm with running time $f(\ell) \cdot n^{O(1)}$ we say that the algorithm runs in FPT-time. A *parameterized problem* is a language $P \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a fixed, finite alphabet and the second component is called the parameter of the problem. For example, we write MSE(p) for Minimum Shared Edges parameterized by the number p of paths, and MSE(p, k) for Minimum Shared Edges parameterized by the number p of paths and the number k of shared edges. We write $(X, \ell) \in P$ for an instance of a parameterized problem P with parameter ℓ. A parameterized problem P is called *fixed-parameter tractable* if there is an FPT algorithm with respect to the parameter ℓ that solves any instance $(X, \ell) \in P$ of the problem P. The complexity class FPT is the class containing all parameterized problems that are fixed-parameter tractable.

An algorithm with running time $f(\ell) \cdot n^{g(\ell)}$, where n denotes the size of the input, ℓ is part of the input and f, g are computable functions only depending on ℓ, is called XP algorithm. We say that a parameterized problem P is in XP if there is an XP algorithm with respect to the parameter ℓ that solves any instance $(X, \ell) \in P$ of the problem P.

We say that a reduction from a parameterized problem P to a parameterized problem Q is a *parameterized reduction* if given an instance $(X, \ell) \in P$ with parameter ℓ, the reduction computes an instance $(X', \ell') \in Q$ with parameter ℓ' in $f(\ell) \cdot |X|^{O(1)}$ time for some computable function f, where $|X|$ denotes the size of X, such that (i) instance (X', ℓ') is a yes-instance if and only if (X, ℓ) is a yes-instance, and (ii) $\ell' \leq g(\ell)$ for some computable function g.

A *kernelization* for a parameterized problem P is an algorithm that reduces a given instance $(X, \ell) \in P$ of the parameterized problem P in polynomial time to an equivalent instance $(X', \ell') \in P$, called the *problem kernel*, such that $|X'| \leq f(\ell)$ and $\ell' \leq f(\ell)$, where f is a computable function only depending on ℓ. The problem kernel (X', ℓ') is called polynomial if the function f is polynomial in ℓ.

The class $W[1]$ is assumed to be the basic class of parameterized intractability, that is $W[1] \neq \text{FPT}$. A parameterized problem P is in the parameterized complexity class $W[2]$ if there is a parameterized reduction from the problem P to the following problem:

Problem: WEIGHTED CNF SAT
Input: A formula ϕ in conjunctive normal form and a number $k \in \mathbb{N}_0$.
Parameter: k.
Question: Is there a satisfying assignment for ϕ with k variables set true?

Accordingly, a parameterized problem P belongs to the parameterized complexity class $W[1]$ if there is a parameterized reduction from P to WEIGHTED 3 CNF SAT, where WEIGHTED 3 CNF SAT is defined as WEIGHTED CNF SAT, but every clause consists of at most three variables. The classes $W[1]$ and $W[2]$ are the first two classes in the so-called W-hierarchy. The relation of the complexity classes presented so far is $\text{FPT} \subseteq W[1] \subseteq W[2] \subseteq \ldots \subseteq \text{XP}$.

A parameterized problem P is called $W[1]$-hard if there is a parameterized reduction from a $W[1]$-hard problem to P. A parameterized problem is called $W[1]$-complete if it is contained in $W[1]$ and $W[1]$-hard. We define a parameterized problem as $W[2]$-hard and as $W[2]$-complete in an analogous way.

For more information on parameterized complexity, and in particular on the W-hierarchy, we refer to the work of Downey and Fellows [6], Flum and Grohe [8] and Niedermeier [15].

3 Basic Observations

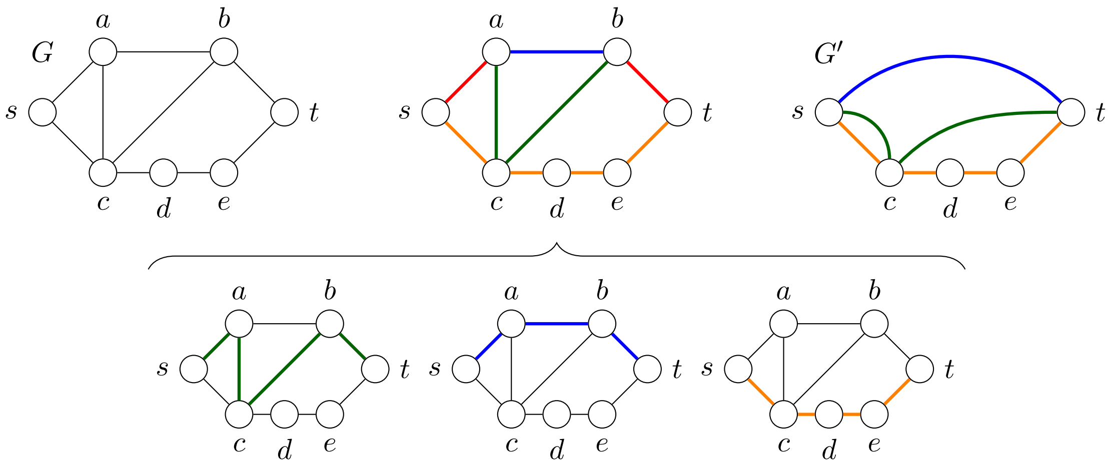

Figure 3.1: Example for a solution for an example instance $(G, s, t, 3, 2)$ of MSE.

In this section, we provide some basic observations on Minimum Shared Edges and an algorithm that solves Minimum Shared Edges in FPT-time with respect to the number p of paths and the number k of shared edges.

In Figure 3.1, we provide an example for Minimum Shared Edges on an example graph G with $s, t \in V(G)$. On the top-left, the example graph G is shown. On the top-middle, a solution for the instance (G, s, t, p, k) for MSE with $p = 3$ and $k = 2$ is shown. The three s-t paths are colored blue, orange, and darkgreen respectively, each illustrated below the top-middle plot. The red-colored edges correspond to the edges that are shared. Here, the blue-colored path and the darkgreen-colored path share the edges $\{s, a\}$ and $\{b, t\}$. On the top-right, graph G' obtained from G by contracting the edges $\{s, a\}$ and $\{b, t\}$ is shown. As a convention throughout this work, if we contract an edge incident with vertex s, then we call the obtained vertex s, and if we contract an edge incident with vertex t, then we call the obtained vertex t. Recall that the two edges $\{s, a\}$ and $\{b, t\}$ are shared by the s-t paths in G. Note that the three s-t paths that are obtained by contracting the edges $\{s, a\}$ and $\{b, t\}$ from the three s-t paths in G are edge-disjoint in G'. The latter observation motivates the following.

Let G be a graph with $s, t \in V(G)$. Let (G, s, t, p, k) be a yes-instance of MSE. Let $\mathcal{P}$ be a set of p s-t paths in G that share at most k edges. Let $F \subseteq E(G)$ be the set of shared edges. Let $\mathcal{P}'$ be the set of p s-t paths in G/F obtained from $\mathcal{P}$ by contracting all edges in F. The p s-t paths in $\mathcal{P}'$ are edge-disjoint in G/F. Hence, by Menger's theorem together with the max-flow min-cut theorem, G/F with unit edge capacities allows an s-t flow of value at least p.

Conversely, let G be a graph with $s, t \in V(G)$, $p \in \mathbb{N}$, and $k \in \mathbb{N}_0$. Let $F \subseteq E(G)$

be a set of edges in G with $|F| \leq k$ such that G/F with unit edge capacities allows an s-t flow of value at least p. Then, by the max-flow min-cut theorem together with Menger's theorem, G/F allows p edge-disjoint s-t paths. Let $\mathcal{P}$ be such a set of p edge-disjoint s-t paths in G/F. Then there is a set $\mathcal{P}'$ of p s-t paths in G such that each path in $\mathcal{P}$ is obtained from a path in $\mathcal{P}'$ by contracting the edges in F. Since the paths in $\mathcal{P}$ are edge-disjoint and $|F| \leq k$, the p s-t paths in $\mathcal{P}'$ share at most k edges.

Following these observations, we provide an equivalent formulation of Minimum Shared Edges as the following contraction problem:

Problem: Minimum Shared Edges - Contraction Equivalent (MSE-CoE)
Input: Graph $G = (V, E)$, $s, t \in V(G)$, $p \in \mathbb{N}$, and $k \in \mathbb{N}_0$.
Parameter: p, k.
Question: Is there a subset $F \subseteq E$ of edges of cardinality at most k in G such that the graph G/F with unit edge capacities allows an s-t flow of value at least p?

We make use of the equivalent problem formulation in our algorithms and proofs.

We call an instance of Minimum Shared Edges *trivial*, if there is an inequality A such that A holds on some parameters of the instance and A verifies that the instance is a yes- or no-instance. For example, if for a graph G with $s, t \in V(G)$ the inequality $k \geq E(G)$ holds, then instance (G, s, t, p, k) is a trivial yes-instance of MSE for every $p \geq 1$. Since if the number k of edges that are allowed to be shared is at least the number of edges in the graph, then we can construct infinitely many paths connecting the source with the sink without sharing more than k edges. In the following lemma we state that if for a given instance of MSE the number k of edges is at least the length of a shortest path from the source to the sink, then the instance is a trivial yes-instance of MSE.

Lemma 3.1. *Let G be a graph and $s, t \in V(G)$. If $k \in \mathbb{N}_0$ is at least the length of a shortest s-t path in G, that is, if $k \geq \operatorname{dist}_G(s, t)$, then (G, s, t, p, k) is a yes-instance of* MSE *for every $p \geq 1$.*

Proof. Let P be a shortest s-t path in G. Let $p \geq 1$ and let $P_1, \ldots, P_p$ be s-t paths in G such that each of the p paths is a copy of path P, i.e. $P_i = P$ for all $i = 1, \ldots, p$. The paths $P_1, \ldots, P_p$ share $|P| = \operatorname{dist}_G(s, t) \leq k$ edges. Therefore, the paths $P_1, \ldots, P_p$ form a solution for instance (G, s, t, p, k) of MSE. □

Given an instance (G, s, t, p, k) of MSE and the length $\operatorname{dist}_G(s, t)$ of a shortest s-t path in G, if $k \geq \operatorname{dist}_G(s, t)$, then (G, s, t, p, k) is a trivial yes-instance of Minimum Shared Edges by Lemma 3.1. The length of a shortest s-t path in a graph G with $s, t \in V(G)$ can be computed in $O(|G|)$ time by using a breadth-first search.

Recall that the length of a shortest path between any pair of vertices in a graph is at most the diameter of the graph. Therefore, if the number k of edges that are allowed to be shared is at least the diameter of graph G with $s, t \in V(G)$, then (G, s, t, p, k) is a trivial yes-instance for all $p \geq 1$.

Corollary 3.2. *Let G be a graph and $s, t \in V(G)$. If $k \geq \operatorname{diam}(G)$, then the instance (G, s, t, p, k) is a yes-instance of* MSE *for all $p \geq 1$.*

Proof. Since $k \geq \operatorname{diam}(G) \geq \operatorname{dist}_G(s, t)$, Lemma 3.1 completes the proof. □

With the next lemma, we show that an instance of MSE is a trivial yes-instance if the number p of paths is at most the value of a maximum flow between the source and the sink in the graph.

Lemma 3.3. *Let G be a graph with unit edge capacities and $s, t \in V(G)$. Let f be a maximum s-t flow in G with value $|f|$. If $p \leq |f|$, then (G, s, t, p, k) is a yes-instance of* MSE *for every $k \geq 0$.*

Proof. Let f be a maximum s-t flow in G with value $|f|$. By the max-flow min-cut theorem, the size of any minimum s-t cut in G is equal to $|f|$. Thus, by Menger's theorem, the number of edge-disjoint s-t paths in G is equal to the value $|f|$. Since $p \leq |f|$, there are at least p edge-disjoint s-t paths in G. Hence, (G, s, t, p, k) is a yes-instance of MSE for every $k \geq 0$, . □

Lemmas 3.1 and 3.3 provide inequalities that allow us to verify if an instance of Minimum Shared Edges is a trivial yes-instance. The following lemma provides an inequality to verify if an instance of Minimum Shared Edges is a trivial no-instance.

Lemma 3.4. *Let G be a graph with maximum degree $\Delta \geq 3$ and let $s, t \in V(G)$. Let $k \in \mathbb{N}_0$ with $k < \operatorname{dist}_G(s, t)$. If $p > \Delta + \lfloor k/2 \rfloor \cdot (\Delta - 2)$, then (G, s, t, p, k) is a no-instance of* MSE.

With the following two lemmas, we prepare the proof of Lemma 3.4. We remark that the size of every minimum s-t cut in a graph G with $s, t \in V(G)$ is upper-bounded by the maximum degree $\Delta(G)$. By the min-cut max-flow theorem, the value of every maximum s-t flow is equal to the size of a minimum s-t cut. Therefore, if the number p of paths is greater than the maximum degree, at least one edge has to be shared. The next lemma is motivated by the question how the maximum degree of a graph G changes if we contract a set of edges in $E(G)$.

Lemma 3.5. *Let G be a graph with $\Delta(G) \geq 2$. Let $F \subseteq E(G)$ such that $G[F]$ is connected. Let v_F be the vertex in $V(G/F) \backslash V(G)$, i.e. the vertex obtained from contracting all edges in F. Then $\deg_{G/F}(v_F) \leq \Delta(G) + |F| \cdot (\Delta(G) - 2)$.*

Proof. Let G be a graph with $\Delta(G) \geq 2$ and $F \subseteq E(G)$ such that $G[F]$ is connected. Since graph $G[F]$ is connected and $|E(G|F])| = |F|$, graph $G[F]$ has at most $|F| + 1$ vertices. Let v_F be the vertex in $V(G/F) \backslash V(G)$, i.e. the vertex obtained from contracting all edges in F. We show that the degree of v_F in G/F is upper-bounded by $\Delta(G) + |F| \cdot (\Delta(G) - 2)$.

All edges incident with vertex v_F in graph G/F have exactly one endpoint in $G[F]$. The sum of the degrees of the vertices in $V(G[F])$ in graph G is at most $\Delta(G)\cdot(|F|+1)$. The graph $G[F]$ has $|F|$ edges, and we counted each edge twice in the sum of the degrees. Therefore, there are at most $\Delta(G)\cdot(|F|+1)-|F|$ edges incident with the vertices in $V(G[F])$ in graph G. Subtracting the edges in F, there are $\Delta(G)\cdot(|F|+1)-|F|-|F| = \Delta(G)+|F|\cdot(\Delta(G)-2)$ edges incident with vertex v_F in graph G/F. □

The statement of the following lemma is that a minimal s-t cut in a graph G with $s,t \in V(G)$ is preserved under the contraction of edges disjoint to the cut.

Lemma 3.6. *Let $G=(V,E)$ be a connected graph with $s,t \in V$ and let $F \subseteq E$ such that there exists a minimal s-t cut C in G with $F \cap C = \emptyset$. Then C is a minimal s-t cut in G with $F \cap C = \emptyset$ if and only if C is a minimal s-t cut in G/F.*

Proof. "$\Leftarrow$": Let $G' := G/F$ and let C be a minimal s-t cut in G'. We suppose that C is not a minimal s-t cut in G.

Case 1: C is not an s-t cut in G. Then there exists an s-t path P in $G\backslash C$. Since $F \cap C = \emptyset$, it follows that $F \subseteq E(G\backslash C)$. Since edge contraction does not disconnect the path P, the path P' after contracting all edges in $E(P)\cap F$ is an s-t path in $G'\backslash C$. This is a contradiction to the fact that C is a minimal s-t cut in G', and hence, C is an s-t cut in G.

Case 2: C is an s-t cut in G, but C is not a minimal s-t cut in G. Then there exists an edge $e \in C$ such that $C' := C\backslash\{e\}$ is an s-t cut in G. Let G_s and G_t be the two connected components in $G\backslash C'$ with $s \in V(G_s)$ and $t \in V(G_t)$. Since $F \cap C = \emptyset$, contracting all edges in F in G_s and G_t yields the disjoint subgraphs G'_s and G'_t of the graph $G'\backslash C'$. This means that C' is an s-t cut in G' with $|C'| < |C|$, contradicting the fact that C is a minimal s-t cut in G', and hence, C is a minimal s-t cut in G.

"$\Rightarrow$": Let C be a minimal s-t cut in G with $F \cap C = \emptyset$. Let $G' := G/F$. We suppose that C is not a minimal s-t cut in G'.

Case 1: C is not an s-t cut in G'. Then there exists an s-t path P in $G'\backslash C$. Since $F \cap C = \emptyset$, there exists an s-t path P' in G such that P' results in P after contracting all edges in $E(P') \cap F$. This is a contradiction to the fact that C is a minimal s-t cut in G, and hence, C is an s-t cut in G'.

Case 2: C is an s-t cut in G', but C is not a minimal s-t cut in G'. Then there exists an edge $e \in C$ such that $C' := C\backslash\{e\}$ is an s-t cut in G'. Let G'_s and G'_t be the two connected components of $G'\backslash C'$ with $s \in V(G'_s)$ and $t \in V(G'_t)$. Since $F \cap C = \emptyset$, the graphs G'_s and G'_t are obtained from $G\backslash C'$ by contracting the edges in F. Since the contraction of edges does not disconnect a connected graph, $C' \subset C$ is an s-t cut in G. This is a contradiction to the fact that C is a minimal s-t cut in G, and hence, C is a minimal s-t cut in G'. □

We are ready to prove Lemma 3.4.

Proof of Lemma 3.4. Let (G, s, t, p, k) be a yes-instance of MSE. Let $F \subseteq E(G)$ with $|F| = k$ such that G/F with unit edge capacities allows an s-t flow of value p. We remark that both edge sets incident with s and t form an s-t cut in G/F. We show that after at most k edge contractions the minimum of the degrees of s and t in G/F is upper-bounded by $\Delta + \lfloor k/2 \rfloor \cdot (\Delta - 2)$.

Case 1: It holds that $s \notin V(G[F])$ or $t \notin V(G[F])$. If $s \notin V(G[F])$, then $\deg_{G/F}(s) \leq \Delta$. If $t \notin V(G[F])$, then $\deg_{G/F}(t) \leq \Delta$. In both cases, it follows that

$$\min\{\deg_{G/F}(s), \deg_{G/F}(t)\} \leq \Delta \leq \Delta + \lfloor k/2 \rfloor \cdot (\Delta - 2).$$

Case 2: It holds that $s, t \in V(G[F])$. Let G_s be the maximally connected subgraph of $G[F]$ with $s \in V(G_s)$ and let G_t be the maximally connected subgraph of $G[F]$ with $t \in V(G_t)$. Since $k < \text{dist}_G(s, t)$, the graphs G_s and G_t are disjoint, i.e. $V(G_t) \cap V(G_s) = \emptyset$. Let $F_s := E(G_s)$ and $F_t := E(G_t)$. Note that $F_s \subseteq F$, $F_t \subseteq F$ and $F_s \cap F_t = \emptyset$. By Lemma 3.6, the edges incident with s in G/F_s form a minimal s-t cut in G/F, since the edges form a minimal s-t cut in G/F_s. Analogously, the edges incident with t in G/F_t form an minimal s-t cut in G/F. By Lemma 3.5, it holds that

$$\deg_{G/F}(s) \leq \Delta + |F_s| \cdot (\Delta - 2),$$
$$\deg_{G/F}(t) \leq \Delta + |F_t| \cdot (\Delta - 2).$$

Since $|F_s| + |F_t| \leq |F|$, it follows that $\min\{|F_s|, |F_t|\} \leq |F|/2 = k/2$. Since $|F_s|$ and $|F_t|$ are integers, it follows that $\min\{|F_s|, |F_t|\} \leq \lfloor k/2 \rfloor$. Hence, we get

$$\min\{\deg_{G/F}(s), \deg_{G/F}(t)\} \leq \Delta + \lfloor k/2 \rfloor \cdot (\Delta - 2).$$

At least one of the edge sets incident with s and t in G/F is upper-bounded by $\Delta + \lfloor k/2 \rfloor \cdot (\Delta - 2)$. Since each of these edge sets forms a minimal s-t cut in G/F, the size of a minimum s-t cut in G/F is at most $\Delta + \lfloor k/2 \rfloor \cdot (\Delta - 2)$. Since the value of any s-t flow in G/F is upper-bounded by the size of a minimum s-t cut in G/F, we get $p \leq \Delta + \lfloor k/2 \rfloor \cdot (\Delta - 2)$. □

We summarize. Let G be a graph with $s, t \in V(G)$. Instance (G, s, t, p, k) is a trivial instance of Minimum Shared Edges, if one of the following inequalities holds:

- $k \geq \text{dist}_G(s, t)$ (Lemma 3.1),
- $p \leq |f|$, where $|f|$ is the value of a maximum s-t flow in G with unit edge capacities (Lemma 3.3),
- $p > \Delta(G) + \lfloor k/2 \rfloor \cdot (\Delta(G) - 2)$ and $k < \text{dist}_G(s, t)$ (Lemma 3.4).

Input: Graph G, $s, t \in V(G)$, $p \in \mathbb{N}$, and $k \in \mathbb{N}_0 \cup \{-1\}$.
Output: TRUE if there are at most k edge contractions in graph G such that there is an s-t flow of value at least p in G with unit edge capacities, and FALSE otherwise.

if $k < 0$ **then**
    return *FALSE*;
end
if $k \geq \mathrm{dist}_G(s,t)$ **then**
    return *TRUE*;
end
$C \leftarrow$ any minimum s-t cut in G with unit edge capacities;
if $|C| \geq p$ **then**
    return *TRUE*;
end
solvable $\leftarrow$ FALSE;
for *each* $e \in C$ **do**
    solvable $\leftarrow$ (solvable $\vee$ MSE$(G/e, s, t, p, k-1)$);
end
return solvable;

Algorithm 3.1: MSE(G, s, t, p, k)

If none of these inequalities holds, then Minimum Shared Edges is hard to solve in general. In Section 5, Theorem 5.1, we show that MSE(k) is W[2]-hard, that is, when parameterized only by the number k of edges. Now, we present Algorithm 3.1 that solves MSE in $(p-1)^k \cdot O(|G|^2)$ time on a graph G. As a consequence, we show that MSE(p, k) is fixed-parameter tractable, that is, when parameterized by the number p of paths and the number k of shared edges. We remark that in Section 7, Theorem 7.1, we show the stronger result that MSE(p) is fixed-parameter tractable, that is, when parameterized only by the number p of paths. However, for small values of k, Algorithm 3.1 performs well compared to the FPT algorithms with respect to parameter p that we will present later in this work.

The idea of the algorithm is that in every minimum s-t cut of size smaller than p in G, at least one edge has to be shared by at least two paths. First, we show that Algorithm 3.1 is correct. To this end, we make use of the following lemma.

Lemma 3.7. *Let G be a graph with $s, t \in V(G)$ and at least one minimum s-t cut of size smaller than p. Then, instance (G, s, t, p, k) is a yes-instance of* MSE *if and only if for all minimum s-t cuts C in G there exists an edge $e \in C$ such that instance $(G/e, s, t, p, k-1)$ is a yes-instance of* MSE*.*

Proof. "$\Rightarrow$": Let (G, s, t, p, k) be a yes-instance of MSE. Let $\mathcal{P}$ be a set of p s-t paths in G that share at most k edges. Let C be an arbitrary minimum s-t cut in G of size smaller than p, i.e. $|C| < p$. Since C is a minimum s-t cut of G, each of the p paths in $\mathcal{P}$ contains an edge in C. Since $|\mathcal{P}| = p$ and $|C| < p$, there is an edge $e \in C$ that appears in at least two paths in $\mathcal{P}$, or in other words, that is shared by at least two paths in $\mathcal{P}$. Let $\mathcal{P}'$ be the set of paths obtained from $\mathcal{P}$ by contracting edge e in each path in $\mathcal{P}$ that contains edge e. Note that every path in $\mathcal{P}'$ is an s-t path in G/e. The p s-t paths in G/e in set $\mathcal{P}'$ share at most $k-1$ edges and verifies that $(G/e, s, t, p, k-1)$ is a yes-instance of MSE.

"$\Leftarrow$": Let C be an arbitrary minimum s-t cut in G and let $e = \{v, w\} \in C$ such that $(G/e, s, t, p, k-1)$ is a yes instance of MSE. Let $\mathcal{P}$ be a set of p s-t paths in G/e sharing at most $k-1$ edges. Let vw be the vertex obtained by the contraction of edge e. Let $P \in \mathcal{P}$ be an s-t path in the set $\mathcal{P}$ of s-t paths containing the vertex vw. Let e_1' and e_2' be the edges incident with vertex vw in path P corresponding to edges e_1 and e_2 in G. We replace edge e_1' by edge e_1, edge e_2' by edge e_2 and vertex vw in P in one of the following ways.

Case 1: $v \in e_1$ and $v \in e_2$. Then we replace vertex vw by vertex v.

Case 2: $w \in e_1$ and $w \in e_2$. Then we replace vertex vw by vertex w.

Case 3: $v \in e_1$ and $w \in e_2$, or $w \in e_2$ and $v \in e_1$. Then we replace vertex vw by vertices v and w, and we add edge e to P.

Let $\mathcal{P}'$ be the set of p paths that results by applying the modifications to each of the paths in $\mathcal{P}$. Then $\mathcal{P}'$ is a set of p s-t paths in G that share at most k edges in G. These are, on the one hand, the at most $k-1$ edges that are shared by the paths in set $\mathcal{P}$, and, on the other hand, edge e, if there are more than two paths in $\mathcal{P}'$ modified due to Case 3. Thus, path set $\mathcal{P}'$ verifies that (G, s, t, p, k) is a yes-instance of MSE. □

We show that Algorithm 3.1 correctly determines whether a given instance of Minimum Shared Edges is a yes-instance or a no-instance.

Lemma 3.8. *Let* (G, s, t, p, k) *be an instance of* MSE. *Algorithm 3.1 returns TRUE if and only if* (G, s, t, p, k) *is a yes-instance of* MSE.

Proof. "$\Leftarrow$": We prove this direction by induction on k in the input for Algorithm 3.1.

Base case. Let $(G, s, t, p, 0)$ be a yes-instance of MSE and let $(G, s, t, p, 0)$ be the input for Algorithm 3.1. If $s = t$, then Algorithm 3.1 returns TRUE (lines 4-6). If $s \neq t$, then the algorithm computes a minimum s-t cut C in G with unit edge capacities. Since $(G, s, t, p, 0)$ is a yes-instance of MSE, there are p edge-disjoint s-t paths in G. By Menger's theorem, the size of any minimum s-t cut in G is at least p. Thus, it holds that $|C| \geq p$ and Algorithm 3.1 returns TRUE (lines 8-10).

Inductive step. Assume that if (G, s, t, p, k) is a yes-instance of MSE, then Algorithm 3.1 returns TRUE on input (G, s, t, p, k). We show that if $(G, s, t, p, k+1)$ is a yes-instance of MSE, then Algorithm 3.1 returns TRUE on input $(G, s, t, p, k+1)$. Let $(G, s, t, p, k+1)$ be a yes-instance of MSE. If $k+1 \geq \text{dist}_G(s,t)$, then Algorithm 3.1 returns TRUE (lines 4-6). If $k+1 < \text{dist}_G(s,t)$, then the algorithm computes a minimum s-t cut C. If $|C| \geq p$, then Algorithm 3.1 returns TRUE (lines 8-10). Otherwise, the algorithm executes a recursive call for each edge in C. By Lemma 3.7, there is an edge $e \in C$ such that $(G/e, s, t, p, k)$ is a yes-instance of MSE. By the induction hypothesis, Algorithm 3.1 returns TRUE on input $(G/e, s, t, p, k)$. Thus, the algorithm returns TRUE on input $(G, s, t, p, k+1)$ (lines 11-15).

"$\Rightarrow$": Let (G, s, t, p, k) be a no-instance of MSE(p, k). Then, for all $F \subseteq E(G)$ with $|F| \leq k$ holds that the value of any maximum s-t flow in G/F is smaller than p. Let $F \subseteq E(G)$ with $|F| = k$ and $(G/F, s, t, p, 0)$ be the input for Algorithm 3.1. Then, there is a minimum s-t cut C of size smaller than p in G/F and the algorithm executes for each $e \in C$ the recursive call $((G/F)/e, s, t, p, -1)$ (lines 9-13). On each of these inputs, the algorithm returns FALSE since $k < 0$ (lines 1-3). Since this holds for all sets $F \subseteq E(G)$ with $|F| = k$, Algorithm 3.1 returns FALSE on input (G, s, t, p, k). □

Given an instance of Minimum Shared Edges, by Lemma 3.8, we can use Algorithm 3.1 to determine whether the instance is a yes-instance or a no-instance of Minimum Shared Edges. Next, we discuss the running time of Algorithm 3.1 and we show that the algorithm runs in FPT-time with respect to the number of paths and the number of shared edges.

Lemma 3.9. *Let G be a connected graph with $s, t \in V(G)$. Let $p \in \mathbb{N}$ and $k \in \mathbb{N}_0$ two integers. Then, Algorithm 3.1 with input (G, s, t, p, k) runs in $(p-1)^k \cdot O(|G|^2)$ time.*

Proof. We define $T[G, s, t, p, k]$ as the running time of Algorithm 3.1 with respect to the input (G, s, t, p, k).

If the algorithm is called for $k \geq 0$, then the length of a shortest s-t path in G is computed. This can be done in $O(|G|)$ time. If $0 \leq k < \text{dist}_G(s,t)$, then a minimum s-t cut is computed for the input graph G and $s, t \in V(G)$. A minimum s-t cut in graph G with unit edge capacities can be computed in $O(|G|^2)$ time [11]. Hence, both computations in any call of the algorithm for graph G can be done in $O(|G|^2)$ time.

If the size of a minimum s-t cut C in G is smaller than p, at most $p-1$ edges are considered in the for-loop. In each recursive call of the algorithm we decrease k by one and contract an edge in G, until k is equal to zero. Therefore, it holds that

$$\begin{aligned} T[G, s, t, p, k] &= O(|G|^2) + \sum_{e \in C} T[G/e, s, t, p, k-1] \\ &\leq O(|G|^2) + (p-1) \cdot \max_{e \in C} T[G/e, s, t, p, k-1], \end{aligned}$$

where C is the minimum s-t cut in G found by the algorithm. Since $|V(G')| \leq |V(G)|$ and $|E(G')| \leq |E(G)|$, the recursion yields $T[G, s, t, p, k] \leq (p-1)^k \cdot O(|G|^2)$. □

By Lemma 3.9, Algorithm 3.1 runs in FPT-time with respect to the number p of paths and the number k of shared edges. Thus, with Algorithm 3.1 we can solve an instance (G, s, t, p, k) of MSE(p, k) in FPT-time with respect to the number p of paths and the number k of edges that are allowed to be shared. We conclude in the following theorem.

Theorem 3.10. MINIMUM SHARED EDGES *is fixed-parameter tractable with respect to the number p of paths and the number k of edges.*

Proof. By Lemma 3.8, Algorithm 3.1 returns TRUE if and only if the input instance is a yes-instance of MINIMUM SHARED EDGES. By Lemma 3.9, Algorithm 3.1 runs in $(p-1)^k \cdot O(|G|^2)$ time. Therefore, Algorithm 3.1 solves MSE(p, k) in FPT-time. $\square$

By Lemma 3.4, we know that the number p of paths is upper-bounded by $\Delta + \lfloor k/2 \rfloor \cdot (\Delta - 2)$. This implies the following.

Corollary 3.11. MINIMUM SHARED EDGES *is fixed-parameter tractable with respect to the number k of shared edges and the maximum degree Δ of the given graph.*

4 Grids

In Section 3, we presented some inequalities that allow us to verify whether an instance of Minimum Shared Edges is trivial. In addition, we presented an FPT algorithm that solves Minimum Shared Edges parameterized by the number of paths and the number of edges that are allowed to be shared.

In this section, we study Minimum Shared Edges on the unbounded, undirected $\mathbb{Z} \times \mathbb{Z}$-grid graph. The unbounded, undirected $\mathbb{Z} \times \mathbb{Z}$-grid graph, or, throughout this section for short, the grid graph, is the graph $\mathbb{G}$ with vertex set $V = \{(x, y) \in \mathbb{Z} \times \mathbb{Z}\}$ and edge set $E = \{\{(x_1, y_1), (x_2, y_2)\} \in \mathbb{Z}^2 \mid |x_1 - x_2| + |y_1 - y_2| = 1\}$. We call an edge $\{(x_1, y_1), (x_2, y_2)\} \in E$ with $y_1 = y_2$ a *horizontal* edge, and we call an edge $\{(x_1, y_1), (x_2, y_2)\} \in E$ with $x_1 = x_2$ a *vertical* edge. We show that any instance of MSE on the grid graph can be verified in constant time.

We remark that our main intention of this section is giving an insight to Minimum Shared Edges by presenting a construction for solutions on the grid graph, in the sense of providing an example for Minimum Shared Edges. Hence, we will not go much into details. We state our main result of this section in the following theorem.

Theorem 4.1. *Let $\mathbb{G}$ be the unbounded, undirected $\mathbb{Z} \times \mathbb{Z}$-grid graph and let $s, t \in V(\mathbb{G})$ be two vertices in $\mathbb{G}$. Let $k \in \mathbb{N}_0$ and $k < \mathrm{dist}_{\mathbb{G}}(s, t)$. Then, $(\mathbb{G}, s, t, p, k)$ is a yes-instance of* MSE *if and only if $p \leq 4 + 2 \cdot \lfloor k/2 \rfloor$.*

In the following, we prepare the proof of the direction "$\Leftarrow$" in the proof of Theorem 4.1. We provide a construction that, given $k \in \mathbb{N}_0$, allows $p = 4 + 2 \cdot \lfloor k/2 \rfloor$ s-t paths sharing at most k edges in $\mathbb{G}$, where $s, t \in V(\mathbb{G})$. Recall that by Lemma 3.1, if $k \geq \mathrm{dist}_{\mathbb{G}}(s, t)$, then we can construct infinitely many s-t paths sharing at most k edges.

Let $s = (x_s, y_s)$ and $t = (x_t, y_t)$ be two vertices in $V(\mathbb{G})$. We assume that $x_s \leq x_t$ and $y_s < y_t - 1$, that is, intuitively, vertex s is below-left of vertex t. Later, we discuss this assumption. Further we assume that the number k of edges is smaller than the length of a shortest s-t path, i.e. $k < (y_t - y_s) + (x_t - x_s)$.

We describe paths in graph $\mathbb{G}$ in the following way.

- Let $a = (x_a, y_a), b = (x_b, y_b) \in \mathbb{Z}^2$ with $x_a = x_b$. We write $a \updownarrow b$ for the path with endpoints a and b using only vertical edges between a and b. We say $a \updownarrow b$ is a *vertical* path.
- Let $a = (x_a, y_a), b = (x_b, y_b) \in \mathbb{Z}^2$ with $y_a = y_b$. We write $a \leftrightarrow b$ for the path with endpoints a and b using only horizontal edges between a and b. We say $a \leftrightarrow b$ is a *horizontal* path.

We represent s-t paths by their vertical and horizontal subpaths. For example, the path $a \updownarrow b \leftrightarrow c$, for three suitable vertices $a, b, c \in V(\mathbb{G})$, is the path with subpaths $a \updownarrow b$

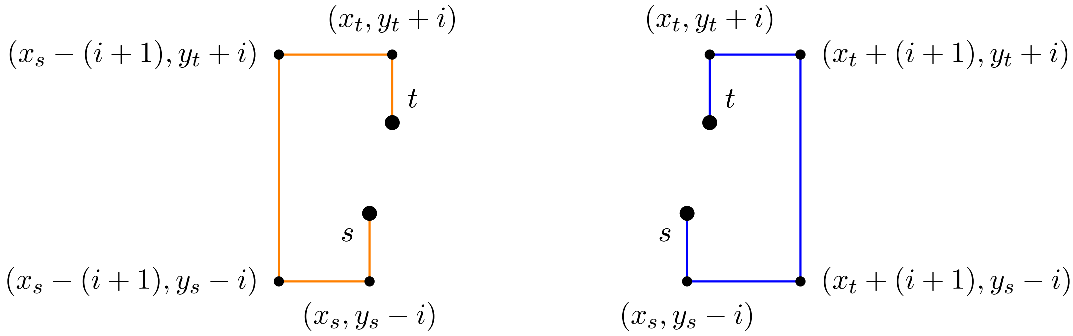

Figure 4.1: Schematic representation of the left and right s-t paths L_i and R_i for an $i \in \mathbb{N}$ in graph $\mathbb{G}$.

and $b \leftrightarrow c$. Now, we construct s-t paths of the following two types. For $i \in \mathbb{N}_0$, we define *left* s-t paths

$$\begin{aligned} L_i :=& (x_s, y_s) \updownarrow (x_s, y_s - i) \leftrightarrow (x_s - (i+1), y_s - i) \updownarrow (x_s - (i+1), y_t + i) \\ &\leftrightarrow (x_t, y_t + i) \updownarrow (x_t, y_t), \end{aligned}$$

and *right* s-t paths

$$\begin{aligned} R_i :=& (x_s, y_s) \updownarrow (x_s, y_s - i) \leftrightarrow (x_t + (i+1), y_s - i) \updownarrow (x_t + (i+1), y_t + i) \\ &\leftrightarrow (x_t, y_t + i) \updownarrow (x_t, y_t). \end{aligned}$$

In Figure 4.1, we provide a schematic representation of a left, orange-colored s-t path L_i and a right, blue-colored s-t path R_i for an $i \in \mathbb{N}$.

By construction, for the left and right s-t paths holds for all $i \in \mathbb{N}$:

(i) L_i and L_{i+1} share exactly the edges in the paths $(x_s, y_s) \updownarrow (x_s, y_s - i)$ and $(x_t, y_t) \updownarrow (x_t, y_t + i)$.

(ii) R_i and R_{i+1} share exactly the edges in the paths $(x_s, y_s) \updownarrow (x_s, y_s - i)$ and $(x_t, y_t) \updownarrow (x_t, y_t + i)$.

(iii) L_i and R_i share exactly the edges in the paths $(x_s, y_s) \updownarrow (x_s, y_s - i)$ and $(x_t, y_t) \updownarrow (x_t, y_t + i)$.

Note that for any $j \in \mathbb{N}$ it holds that the set of shared edges by the s-t paths L_{j-1} and L_j is contained in the set of shared edges by the s-t paths L_j and L_{j+1}, and the same holds for the right s-t paths. Thus, for every $i \in \mathbb{N}_0$, the s-t paths $L_0, \ldots, L_{i+1}$ as well as the s-t paths $R_0, \ldots, R_{i+1}$ share exactly the edges in the paths $(x_s, y_s) \updownarrow (x_s, y_s - i)$ and $(x_t, y_t) \updownarrow (x_t, y_t + i)$. Note that an implication of combining (i) and (iii) is that the s-t paths L_{i+1} and R_i share exactly the edges of the paths $(x_s, y_s) \updownarrow (x_s, y_s - i)$

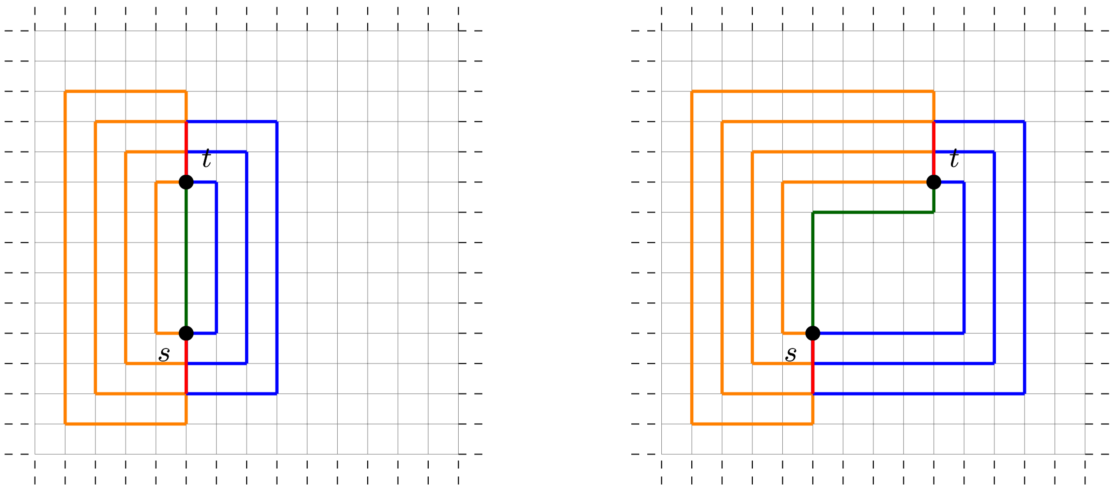

Figure 4.2: Two examples of the set $\mathcal{P}(4)$ of eight s-t paths in graph $\mathbb{G}$, according to the positions of the vertices s and t.

and $(x_t, y_t) \updownarrow (x_t, y_t + i)$. Hence, the s-t paths $L_0, \dots, L_{i+1}, R_0, \dots, R_i$ share exactly the edges in the paths $(x_s, y_s) \updownarrow (x_s, y_s - i)$ and $(x_t, y_t) \updownarrow (x_t, y_t + i)$.

In addition to the left and right s-t paths, we construct a *special* s-t path

$$P^* := (x_s, y_s) \updownarrow (x_s, y_t - 1) \leftrightarrow (x_t, y_t - 1) \updownarrow (x_t, y_t).$$

Note that P^* does not share any edge with any of the left and right s-t paths.

For every $n \in \mathbb{N}_0$, we define a set of s-t paths

$$\mathcal{P}(n) := \{P^*, L_0, \dots, L_{\lfloor n/2 \rfloor + 1}, R_0, \dots, R_{\lfloor n/2 \rfloor}\}.$$

In Figure 4.2, we present two examples for $\mathcal{P}(4)$ on graph $\mathbb{G}$. On the left-hand side, the vertices s and t are positioned in such a way that $x_s = x_t$ and $y_s = y_t - 5$. On the right-hand side, the vertices s and t are positioned in such a way that $x_s = x_t - 4$ and $y_s = y_t - 5$. The orange-colored s-t paths correspond to the left s-t paths, the blue-colored s-t paths correspond to the right s-t paths, and the two green-colored s-t paths correspond to the special s-t paths. Red-colored edges indicate the edges that are shared by at least two s-t paths.

We remark that

$$\mathcal{P}(n+1) = \begin{cases} \mathcal{P}(n), & \text{if } n \text{ is even,} \\ \mathcal{P}(n) \cup \{L_{\lfloor n/2 \rfloor + 2}, R_{\lfloor n/2 \rfloor + 1}\}, & \text{if } n \text{ is odd,} \end{cases}$$

and that $|\mathcal{P}(n)| \leq 4 + 2 \cdot \lfloor n/2 \rfloor$. Recall that P^* does not share any edge with any left or right s-t path, and that the s-t paths $L_0, \dots, L_{\lfloor n/2 \rfloor + 1}, R_0, \dots, R_{\lfloor n/2 \rfloor}$ share exactly the edges in the paths $(x_s, y_s) \updownarrow (x_s, y_s - \lfloor n/2 \rfloor)$ and $(x_t, y_t) \updownarrow (x_t, y_t + \lfloor n/2 \rfloor)$. Thus, the s-t paths in set $\mathcal{P}(n)$ share $\lfloor n/2 \rfloor + \lfloor n/2 \rfloor \leq n$ edges. We conclude that for every $n \in \mathbb{N}_0$, the set $\mathcal{P}(n)$ contains $4 + 2 \cdot \lfloor n/2 \rfloor$ s-t paths that share at most n edges.

The construction allows us to provide a sketch of a proof of Theorem 4.1, if for $s = (x_s, y_s)$ and $t = (x_t, y_t)$ it holds that $x_s \leq x_t$ and $y_s < y_t - 1$. We show next that the construction can be adjusted with small effort for any positions of vertex s and vertex t.

If $x_s \leq x_t$, $y_s \leq y_t$ and $|y_s - y_t| \leq 1$, then we consider the following two cases.

Case 1: $|x_s - x_t| \leq 1$. Since $|y_s - y_t| \leq 1$, it follows that $\mathrm{dist}_{\mathbb{G}}(s, t) \leq 2$. If $k \geq \mathrm{dist}_{\mathbb{G}}(s, t)$, then we can construct infinitely many s-t paths in $\mathbb{G}$ sharing at most k edges. Let $k < \mathrm{dist}_{\mathbb{G}}(s, t) \leq 2$. For each $k \in \{0, 1\}$ holds that $4 + 2 \cdot \lfloor k/2 \rfloor = 4$. Since the value of any maximum s-t flow in $\mathbb{G}$ is equal to four (stated here without proof), we can construct $p = 4$ edge-disjoint s-t paths in $\mathbb{G}$.

Case 2: $|x_s - x_t| > 1$. Then we obtain a feasible construction by switching the x- and y-coordinates in the constructions. Note that if $x'_s = y_s$, $y'_s = x_s$ and $x'_t = y_t$, $y'_t = x_t$, then $x'_s \leq x'_t$ and $y'_s < y'_t - 1$, consistently with the basic case yet presented.

Note that for all other cases, we can find a reflection $\phi : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z} \times \mathbb{Z}$ such that $\phi(x_s) \leq \phi(x_t)$ and $\phi(y_s) \leq \phi(y_t)$. Remark that each of the reflections is an involution. Then, we can apply the construction presented above with $s' = (\phi(x_s), \phi(y_s))$ and $t' = (\phi(x_t), \phi(y_t))$. Since graph $\mathbb{G}$ is undirected and ϕ is a reflection, this yields a construction for any positions of the vertices s and t in $\mathbb{G}$.

Sketch of a proof of Theorem 4.1. Let $k \in \mathbb{N}_0$ and $k < \mathrm{dist}_{\mathbb{G}}(s, t)$.

"$\Rightarrow$": Let $p > 4 + 2 \cdot \lfloor k/2 \rfloor$. Since the maximum degree of graph $\mathbb{G}$ is four, it follows that $p > 4 + 2 \cdot \lfloor k/2 \rfloor = \Delta(\mathbb{G}) + \lfloor k/2 \rfloor \cdot (\Delta(\mathbb{G}) - 2)$. Thus, by Lemma 3.4, the instance $(\mathbb{G}, s, t, p, k)$ is a no-instance of Minimum Shared Edges.

"$\Leftarrow$": We construct the set $\mathcal{P}(k)$ of $p = 4 + 2 \cdot \lfloor k/2 \rfloor$ s-t paths. Since the s-t paths in $\mathcal{P}(k)$ share at most k edges, the construction yields a solution for instance $(\mathbb{G}, s, t, p, k)$ of Minimum Shared Edges. $\square$

By Theorem 4.1, we know that Minimum Shared Edges can be solved in constant time on $\mathbb{G}$. However, it remains open whether Minimum Shared Edges on planar graphs can be solved in polynomial time.

5 Hardness Results

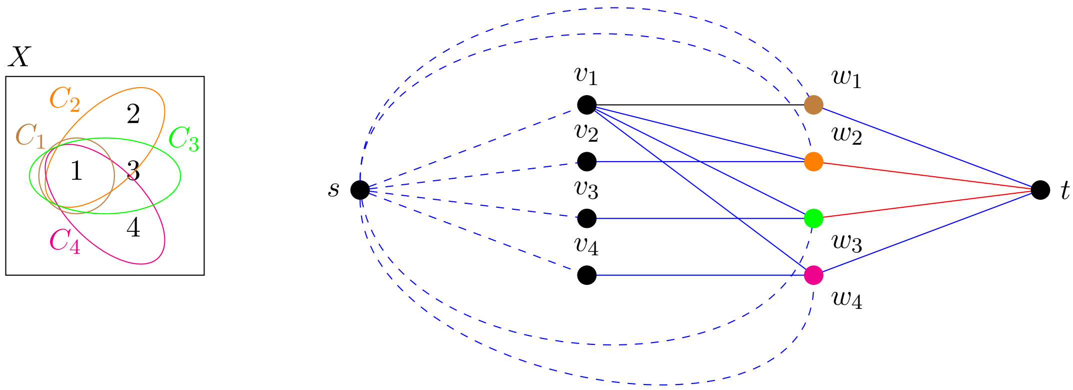

Figure 5.1: Counter example for adapting the reduction due to Omran et al. [16] for the undirected case of MINIMUM SHARED EDGES.

In Section 3, we showed that MSE(k) is in XP. In this section, we show that MINIMUM SHARED EDGES is $W[2]$-hard with respect to the number k of shared edges. To this end, we give a parameterized reduction from the SET COVER problem. Upon this reduction, we show that MINIMUM SHARED EDGES is NP-complete. Further, we show that MINIMUM SHARED EDGES remains NP-hard on graphs with maximum degree at least five, by giving a reduction from the VERTEX COVER problem.

Theorem 5.1. MINIMUM SHARED EDGES *is* $W[2]$*-hard with respect to the number* k *of shared edges.*

In the proof of Theorem 5.1 we provide a reduction from the following problem.

Problem: SET COVER (SC)
Input: A set X, a set of sets $\mathcal{C} \subseteq 2^X$, and an integer ℓ.
Parameter: ℓ.
Question: Are there sets $C_1, \ldots, C_{\ell'} \in \mathcal{C}$ with $\ell' \leq \ell$ such that $X = \bigcup_{i=1}^{\ell'} C_i$?

Omran et al. [16] showed that MINIMUM SHARED EDGES on directed graphs is NP-hard using a reduction from SET COVER. In addition, since their reduction is a parameterized reduction with respect to the number k of shared edges, they showed implicitly that MSE(k) on directed graphs is $W[2]$-hard. Illustrated as a counter example in Figure 5.1, we can not adapt their reduction for MSE on undirected graphs. Here, adapting means to apply the reduction described by Omran et al. [16] and remove the directions of the edges in the directed graph to convert it into an undirected graph. The left-hand side instance of Set Cover in Figure 5.1 does not allow a set cover of at most two sets, since the sets C_2, C_3, and C_4 are essential to cover the elements 2, 3, and 4. Adapting the reduction, the right-hand side instance of Minimum Shared Edges resulting from the adapted reduction should not allow eight s-t-paths sharing at most two edges. As illustrated, the right-hand instance allows eight s-t-paths sharing two edges, where blue lines correspond to edges used by exactly one s-t path and red lines correspond to shared edges. Dashed lines represent paths of length 3. The problem that occurs after removing the direction of the edges is, roughly speaking, that the paths are allowed to go *backwards*, where here backwards is related to if we read the graph as illustrated from left to right. We remark that the reduction we present next is closely related to their reduction.

Now, we present a parameterized reduction of each instance $(X, \mathcal{C}, \ell)$ of SC(ℓ) to an instance (G, s, t, p, k) of MSE(k). We remark that SC(ℓ) is well-known to be $W[2]$-complete [6]. In the following, we call a path of length $m \in \mathbb{N}$ an *m-chain*, consistently with Omran et al. [16].

Proof of Theorem 5.1. Let $(X, \mathcal{C}, \ell)$ be an instance of SC(ℓ). Let $\deg(x)$ be the number of sets in $\mathcal{C}$ containing element $x \in X$, that is, $\deg(x) := |\{C \in \mathcal{C} \mid x \in C\}|$ for every $x \in X$. We reduce the instance $(X, \mathcal{C}, \ell)$ to an instance (G, s, t, p, k) of MSE(k) with $p = |\mathcal{C}| + \sum_{x \in X} \deg(x)$ and $k = \ell$ as follows.

Construction. Initially, let G be an empty graph, that is $V(G) = E(G) = \emptyset$. First, we add the vertices s and t to the vertex set $V(G)$ of graph G. Next, we add to $V(G)$ the following vertex sets:

- $V_X = \{v_i \mid i \in X\}$, the set of vertices corresponding to the elements of X,
- $V_C = \{w_j \mid C_j \in \mathcal{C}\}$, the set of vertices corresponding to the sets in $\mathcal{C}$,
- $V_D = \{v_{i,j} \mid (i \in X) \wedge (C_j \in \mathcal{C}) \wedge (i \in C_j)\}$, the set of vertices corresponding to the relation of the elements in X with the sets in $\mathcal{C}$, i.e. a vertex $v_{i,j}$ is in V_D if there is an element $i \in X$ and a set $C_j \in \mathcal{C}$ such that $i \in C_j$, and
- $V_T = \{t_i \mid i \in X\}$.

We connect each $v_{i,j} \in V_D$ via an $(\ell+1)$-chain with $v_i \in V_X$, with $t_i \in V_T$ and with $w_j \in V_C$. Next, we connect vertex s with every vertex $w \in V_C$ via an $(\ell+1)$-chain and with each $v_i \in V_X$ via $\deg(i)$ $(\ell+1)$-chains. Finally, we connect vertex t with each $w \in V_C$ via a single edge each and with each $t_i \in V_T$ via $\deg(i) - 1$ $(\ell+1)$-chains. Figure 5.2 illustrates this construction on an example instance of Set Cover.

Correctness. Suppose that we have p s-t paths in G that share at most k edges. We show that we can construct a set cover $\mathcal{C}' \subseteq \mathcal{C}$ of X with $|\mathcal{C}'| \leq \ell$. First, we provide some observations.

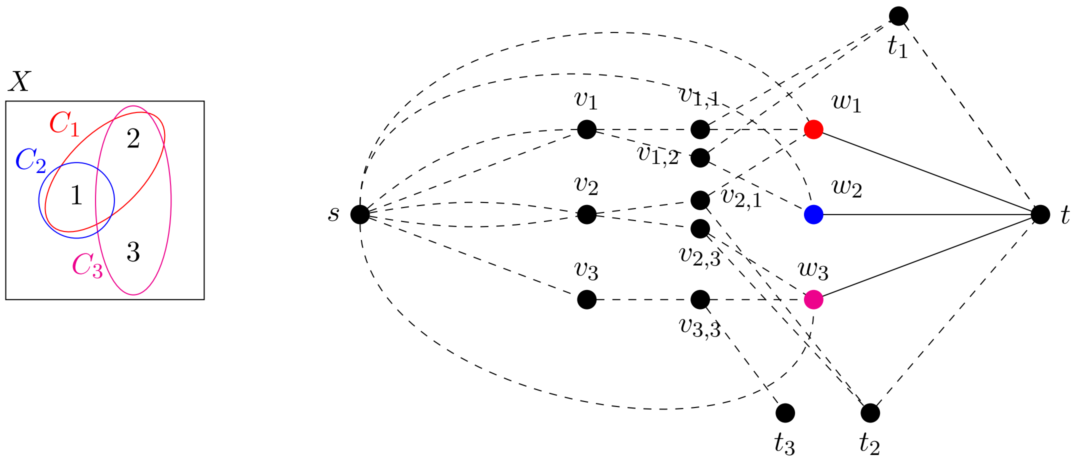

Figure 5.2: Illustration of the construction of the graph G (right-hand side) in the reduction from an instance of Set Cover on the left-hand side to an instance of Minimum Shared Edges. Dashed lines represent $(\ell+1)$-chains, where ℓ is the parameter in the instance of Set Cover.

Since every $(\ell+1)$-chain contains $\ell+1$ edges, every $(\ell+1)$-chain in G appears in at most one s-t path. Since there are p s-t paths and there are p $(\ell+1)$-chains incident with vertex s, every $(\ell+1)$-chain incident with vertex s appears in exactly one s-t path. Therefore, each $v_i \in V_X$ appears in at least $\deg(i)$ s-t paths and each $w_j \in V_C$ appears in at least one s-t path. Moreover, since each $v_i \in V_X$ is incident with $2{\cdot}\deg(i)$ $(\ell+1)$-chains, each $v_i \in V_X$ appears in exactly $\deg(i)$ s-t paths.

Each $v_{i,j} \in V_D$ has exactly degree three and is incident with three $(\ell+1)$-chains. Therefore, every $v_{i,j} \in V_D$ appears in at most one s-t path. Moreover, since each $v_i \in V_X$ appears in $\deg(i)$ s-t paths, and there are $\deg(i)$ vertices in V_D each connected with v_i via an $(\ell+1)$-chain, each $v_{i,j} \in V_D$ appears in exactly one s-t path.

Let $V' := \{w \in V_C | \{w,t\} \text{ is a shared edge}\}$. Set V' is the set of vertices in V_C that are incident with the shared edges of the p s-t paths. We claim that if $w_j \in V_C$ appears in an s-t path P containing a vertex in V_X, then $w_j \in V'$. Let $v_i \in V_X$ be the vertex that appears in path P. Suppose V' does not contain vertex w_j, and thus, edge $\{w_j, t\}$ is not shared. Since the $(\ell+1)$-chain connecting vertex s with vertex w_j appears in exactly one s-t path different from P, vertex w_j appears in at least two s-t paths. Since every vertex in V_C is incident with vertex t and vertices in V_D via $(\ell+1)$-chains, there is a vertex $v_{i',j'} \in V_D$ different from vertex $v_{i,j}$, such that one of the s-t paths containing vertex w_j contains vertex $v_{i',j'}$. Let P' be the path containing the vertices w_j and $v_{i',j'}$. We know that there is an s-t path containing vertex $v_{i',j'}$ and vertex $v_{i'}$ different from P', since there are p s-t paths. Thus, vertex $v_{i',j'}$ appears in at least two s-t paths, contradicting the fact that each vertex in V_D appears in exactly one s-t path. We conclude that set V' contains vertex w_j.

We claim that the subset $\mathcal{C}' \subseteq \mathcal{C}$ corresponding to vertices in V', that is $\mathcal{C}' := \{C_j \in \mathcal{C} \mid w_j \in V'\}$, is a set cover of X of size at most ℓ. Each $t_i \in V_T$ is connected with vertex t via $\deg(i) - 1$ $(\ell + 1)$-chains, and connected with $\deg(i)$ vertices in V_D. Therefore, for each $i \in X$, there exists at least one $j \in [|\mathcal{C}|]$, such that v_i, $v_{i,j}$, and w_j appear in an s-t path. As shown before, it follows that $w_j \in V'$. Thus, for each element $i \in X$ there exists a set $C_j \in \mathcal{C}'$ such that $i \in C_j$, and hence, $\mathcal{C}'$ is a set cover of X of size at most ℓ.

Conversely, suppose that we have a set $\mathcal{C}' \subseteq \mathcal{C}$ with $|\mathcal{C}'| \leq \ell$, such that $\mathcal{C}'$ is a set cover of X. We show that we can construct p s-t paths in G that share at most $k = \ell$ edges.

First, we construct $|\mathcal{C}|$ s-t paths in the following way. For each vertex $w \in V_C$, we construct the s-t path containing only the $(\ell + 1)$-chain connecting s and w and the edge $\{w, t\}$. It follows that each of the $|\mathcal{C}|$ edges connecting a vertex in V_C with vertex t appears in exactly one s-t path.

Next, we construct $|X|$ s-t paths in the following way. We remark that since $\mathcal{C}'$ is a set cover of X, for each $i \in X$ there exists a $C_j \in \mathcal{C}'$ such that $i \in C_j$. For each $v_i \in V_X$, we construct an s-t path containing only the $(\ell+1)$-chains connecting s with v_i, v_i with $v_{i,j}$, $v_{i,j}$ with w_j, and the edge $\{w_j, t\}$, where vertex $w_j \in V_C$ corresponds to a $C_j \in \mathcal{C}'$ with $i \in C_j$. Since $|\mathcal{C}'| \leq \ell$, there are at most ℓ edges connecting the vertices in V_C with t that are shared by the s-t paths constructed so far.

Finally, we construct $\sum_{x \in X} \deg(x) - |X|$ s-t paths in the following way. Note that for each $v_i \in V_X$, there are $\deg(i) - 1$ $(\ell + 1)$-chains connecting s and v_i not covered by an s-t path and there are $\deg(i) - 1$ vertices in V_D connected with v_i via an $(\ell + 1)$-chain not covered by an s-t path. Moreover, no vertex in V_T is covered by an s-t path, and thus, $t_i \in V_T$ is not covered by an s-t path. Recall that $t_i \in V_T$ is connected with vertex t by $\deg(i) - 1$ $(\ell + 1)$-chains. Thus, for each $v_i \in V_X$, we can lead $\deg(i) - 1$ s-t paths from s over v_i, vertices in V_D and t_i to t without sharing any edge.

In total, we constructed

$$|\mathcal{C}| + |X| + \sum_{x \in X} \deg(x) - |X| = |\mathcal{C}| + \sum_{x \in X} \deg(x) = p$$

s-t paths sharing at most $k = \ell$ edges. □

By Theorem 5.1, we know that MSE(k) is $W[2]$-hard. We remark that in the reduction in the proof of Theorem 5.1, the size of the instance of MSE is polynomial in the size of the instance of SC, and thus, we showed that MSE is NP-hard. MSE is also in NP, since we can verify in polynomial time any certificate, that is, whether a given set of at least p s-t paths share at most k edges. Hence, we conclude that Minimum Shared Edges is NP-complete.

Next, we show that Minimum Shared Edges remains NP-hard even on graphs with maximum degree five.

Theorem 5.2. Minimum Shared Edges *is NP-hard even on graphs with maximum degree at least five.*

We prove Theorem 5.2 by giving a reduction from the following problem:

Problem: Vertex Cover (VC)
Input: An undirected graph $G = (V, E)$ and an integer $\ell \in \mathbb{N}$.
Parameter: ℓ.
Question: Is there a subset $V' \subseteq V$ with $|V'| \leq \ell$ such that each edge in E is incident with at least one vertex in V'?

Garey et al. [9] showed that Vertex Cover remains NP-complete on graphs with maximum degree three.

Proof of Theorem 5.2. We provide a polynomial time reduction from Vertex Cover to Minimum Shared Edges. Given an instance (G, ℓ) of VC with $\Delta(G) \geq 3$, we construct an instance (G', s, t, p, k) of MSE with $\Delta(G') = \Delta(G) + 2 \geq 5$ as follows.

Construction. Let $G = (V, E)$ be the graph given in the instance (G, ℓ) of VC with maximum degree $\Delta := \Delta(G) \geq 3$, vertex set $V = \{x_i \mid 1 \leq i \leq |V|\}$ and edge set $E = \{e_i \mid 1 \leq i \leq |E|\}$. We construct a graph $G' = (V', E')$ as follows.

Let $G' = (V', E')$ be initially the empty graph, i.e. $V' = E' = \emptyset$. We add the vertex sets $V_e = \{w_i \mid e_i \in E\}$ and $V_v = \{v_i \mid x_i \in V\}$ to V'. The vertices in V_e correspond to the edges in G and the vertices in V_v correspond to the vertices in G. Let T_s be a binary tree rooted at vertex s with $\lfloor (|V| + |E|)/2 \rfloor$ leaves, where the depths of any two leaves differ by at most one. Let T_t be a binary tree rooted at vertex t with $\lfloor |V|/2 \rfloor$ leaves, where the depths of any two leaves differ by at most one. We add the trees T_s and T_t to graph G'. We define

$$L := |E(T_s)| + |E(T_t)| + \ell + 1.$$

We connect the vertices in $V_v \cup V_e$ with the leaves of tree T_s via L-chains in such a way that every vertex in $V_v \cup V_e$ is connected with exactly one leaf of tree T_s, and every leaf of tree T_s is connected with at least two and at most three vertices in $V_v \cup V_e$. This construction of L-chains is possible since tree T_s contains $\lfloor (|V| + |E|)/2 \rfloor$ leaves and $|V_v \cup V_e| = |V| + |E|$. We add edges to E' connecting the vertices in V_v with the leaves of tree T_t in such a way that every vertex in V_v is connected by a single edge with exactly one leaf of T_t and every leaf of T_t is connected with at least two and at most three vertices in V_v by a single edge each. This construction of edges is possible, since tree T_t contains $\lfloor |V|/2 \rfloor$ leaves, and $|V_v| = |V|$. Finally, for each $w_i \in V_e$ and each $v_j \in V_v$, we connect vertex w_i with vertex v_j via an L-chain if and only if vertex x_j corresponding

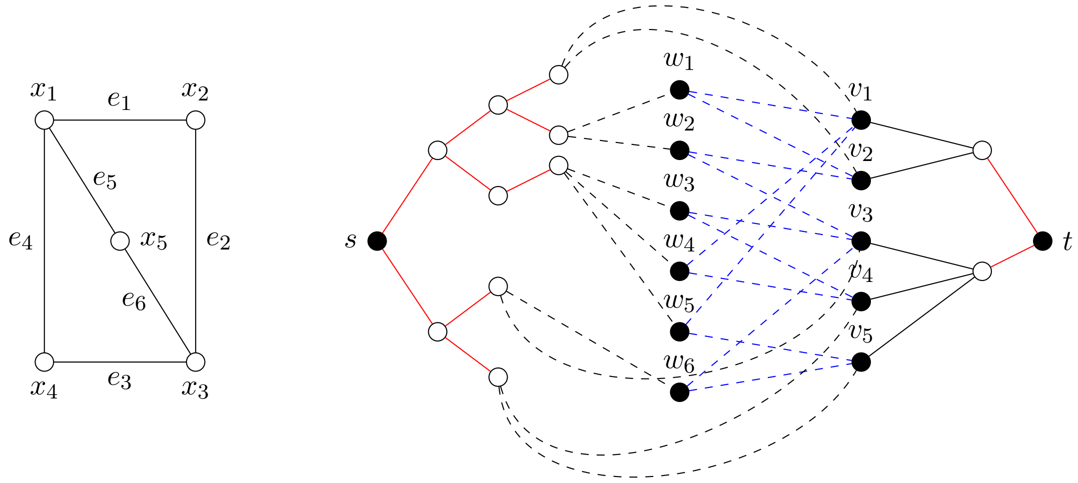

Figure 5.3: Example for the construction in the reduction from VERTEX COVER to MINIMUM SHARED EDGES. On the left-hand side, the graph G for VERTEX COVER is shown. On the right-hand side, the constructed graph G' for MINIMUM SHARED EDGES is shown. Red lines correspond to edges in the two binary trees T_s and T_t. Blue lines correspond to the edge set derived from the incidence relation of the vertices and edges of G. All dashed lines correspond to L-chains. If ℓ is the parameter in the instance of VERTEX COVER, then $L = 11 + \ell + 1$.

to v_j is an endpoint of edge e_i corresponding to w_i, i.e. $x_j \cap e_i \neq \emptyset$. In Figure 5.3, we provide an example for the construction.

We remark that every vertex in $V(T_s)$ and in $V(T_t)$ has degree at most four, each of the vertices s and t has degree two, every vertex in V_e has degree three, and every vertex in V_v has degree at most $\Delta + 2$. Thus, the maximum degree of G' is $\Delta + 2$.

With $p = |V| + |E|$, $k = L - 1$, and $\Delta' = \Delta + 2$, this construction yields an instance (G', s, t, p, k) of MSE, where the maximum degree of graph G' is at least five. We show that (G, ℓ) is a yes-instance of VC if and only if (G', s, t, p, k) is a yes-instance of MSE.

Correctness. Suppose that we have p s-t paths in G' that share at most k edges. We state the following observations.

(i) Every L-chain appears in at most one s-t path.

To see this, suppose that there is an L-chain that appears in at least two s-t paths. Then there are $L = k + 1 > k$ shared edges, contradicting the fact that we have p s-t paths in G' sharing at most k edges.

(ii) Every L-chain connecting a leaf of T_s with a vertex of $V_e \cup V_v$ appears in exactly one s-t path, and every s-t path contains exactly one of the L-chains connecting a leaf of T_s with a vertex in $V_e \cup V_v$.

We show that (ii) holds. We remark that there are $p = |V| + |E|$ L-chains connecting the leaves of T_s with the vertices in $V_e \cup V_v$ and p s-t paths. Since every s-t path leads over a leaf of tree T_s, and every L-chain appears in at most one s-t path by (i), there is a one-to-one correspondence between the p s-t paths and the p L-chains connecting the leaves of T_s with the vertices in $V_e \cup V_v$. Thus, every vertex in V_v appears in at least one s-t path, and every vertex in V_e appears in exactly one s-t path since every vertex in V_e has degree three and is incident with exactly three L-chains.

(iii) All edges in the trees T_s and T_t are shared (these are $L - \ell - 1$ many).

In the following, we prove observation (iii). We consider tree T_s. Tree T_s has $\lfloor (|V| + |E|)/2 \rfloor$ leaves. By construction, there are $|V| + |E|$ L-chains incident with the leaves of T_s. Recall that there are $p = |V| + |E|$ s-t paths and each of the p L-chains incident with the leaves of T_s appears in exactly one s-t path by (ii). Since each of the leaves of T_s is incident with at least two and at most three L-chains, for every leaf of T_s there exist at least two s-t paths connecting s with the leaf. Hence, every edge in tree T_s is shared by the p s-t paths.

We consider tree T_t. Tree T_t has $\lfloor |V|/2 \rfloor$ leaves. We claim that each of the edges connecting V_v with the leaves of T_t appears in at least one s-t path. Suppose that there is an edge f that connects a vertex in V_v with a leaf of tree T_t and that does not appear in any s-t path. Let $v_i \in V_v$ with $v_i \in f$. Vertex v_i appears in at least one s-t path P, since there is an L-chain connecting v_i with a leaf of tree T_s which is used by exactly one s-t path by (ii). Since P does not contain the edge f, path P uses an L-chain connecting vertex v_i with a vertex $w_j \in V_e$. Vertex w_j appears in an s-t path P' using the L-chain connecting w_j with a leaf of T_s by (ii). Since $P \neq P'$, vertex w_j appears in two s-t paths, which contradicts the fact that every vertex in V_e appears in exactly one s-t path. Hence, each of the edges connecting the vertices in V_v with the leaves of T_t appears in at least one s-t path. Since, by construction, every leaf of tree T_t is connected with at least two and at most three vertices in V_v by an edge, and every s-t path leads over the leaves of T_t to vertex t, every edge in tree T_t is shared by the p s-t paths.

We remark that, as a consequence of (iii), the edges connecting the vertices in V_v with the leaves of T_t are the only edges that can be shared by the s-t paths beside the edges in the trees T_s and T_t.

(iv) For every vertex $w_j \in V_e$, there is exactly one s-t path that contains w_j and a vertex in V_v.

To see (iv), we know that each vertex in V_e appears in exactly one s-t path, and each L-chain connecting the leaves of tree T_s with the vertices in V_e appears in exactly one

s-t path by (ii). Since each vertex in V_e is connected with a leaf of T_s and with two vertices in V_v via L-chains, for each vertex $w_j \in V_e$ there exists an s-t path containing w_j and a vertex in V_v.

Let $Z := \{y \in V_v \mid \{y, t'\}$ is a shared edge with $t' \in V(T_t)\}$. Set Z is the set of vertices in V_v such that each vertex is an endpoint of an edge that is connecting the vertices in V_v with the leaves of T_t and that is shared by the p s-t paths. We claim that the set $C := \{x_i \in V \mid v_i \in Z\}$ is a vertex cover of G of size at most ℓ. We consider an s-t path P containing a vertex $w_j \in V_e$ and a vertex $v_i \in V_v$. We know by (iv) that such an s-t path exists for every vertex in V_e. We show that vertex v_i is in the set Z.

Suppose that vertex v_i is not contained in set Z, then the edge connecting v_i with a leaf of tree T_t is covered by at most one s-t path. Since vertex v_i appears in an s-t path $P' \neq P$ that contains the L-chain connecting v_i with a leaf of tree T_s, vertex v_i appears in at least two s-t paths. Let Q be one of the two s-t paths P and P' such that Q does not contain the edge connecting vertex v_i with a leaf of tree T_t. Since vertex v_i is connected with one leaf of tree T_s, one leaf of tree T_t and vertices in V_e, path Q contains an L-chain connecting v_i with a vertex $w_{j'} \in V_e$ different from w_j, i.e. $w_{j'} \neq w_j$. Since the L-chain connecting vertex $w_{j'}$ with a leaf of tree T_s appears in exactly one s-t path different from Q, vertex $w_{j'}$ appears in at least two s-t paths. This is a contradiction to the fact that each vertex in V_e appears in exactly one s-t path, and hence, set Z contains vertex v_i.

Since the p s-t paths share at most $k = L - 1$ edges and, by (iii), all the edges in the trees T_s and T_t are shared, set Z has size at most ℓ, i.e. $|Z| \leq \ell$. Thus, by the one-to-one correspondence between the vertices in V_v and the vertices in $V(G)$, the set C has size at most ℓ, i.e. $|C| \leq \ell$. We show that C is a vertex cover of G. Let $e_j \in E(G)$ be an arbitrary edge of G and let $w_j \in V_e$ be the vertex in G' corresponding to edge e_j. We know by (iv) that there is exactly one s-t path that contains w_j and a vertex $v_i \in V_v$. We know that vertex v_i is contained in set Z. Let $x_i \in V(G)$ be the vertex in G corresponding to vertex v_i. By construction, $x_i \cap e_j \neq \emptyset$, and since $v_i \in Z$, it holds that $x_i \in C$. This means that vertex x_i covers edge e_j. Since edge e_j was chosen arbitrarily, every edge in G is incident with a vertex in C and thus, C is a vertex cover of G of size at most ℓ.

Conversely, let C be a vertex cover of size at most ℓ of G. Let $Z := \{v_i \in V_v \mid x_i \in C\}$ be the vertices in G' corresponding to the vertices in the vertex cover C of G. We show that we can construct p s-t paths in G that share at most $k = L - 1$ edges.

First, we construct $|V|$ s-t paths in such a way that no vertex in V_e appears in any of these $|V|$ s-t paths and each s-t path contains exactly one vertex in V_v. We remark that by that construction, every edge connecting the vertices in V_v with the leaves of tree T_t appears in exactly one s-t path, and every edge in tree T_t is shared.

Next, we construct the remaining $|E|$ s-t paths in the following way. Each of the $|E|$ s-t paths contains exactly one vertex in V_e and one vertex in V_v, and no L-chain connecting the leaves of T_s with vertices in V_v appears in any of the $|E|$ s-t paths. Since C is a vertex cover of G, we can construct the $|E|$ s-t paths in such a way that each of the $|E|$ s-t paths contains exactly one vertex in V_e and one vertex in Z. We remark that by the construction of these $|E|$ s-t paths additionally to the $|V|$ s-t paths constructed before, every edge in tree T_s is a shared edge. Since $|Z| \leq \ell$, there are at most ℓ edges connecting the vertices in V_v with the leaves of tree T_t that are shared by the $|V| + |E|$ s-t paths. Together with the $|E(T_s)|+|E(T_t)|$ shared edges in T_s and T_t, the constructed p s-t paths share at most $k = |E(T_s)| + |E(T_t)| + \ell = L - 1$ edges in G'. □

We showed that Minimum Shared Edges remains NP-hard on graphs with bounded maximum degree at least five. We remark that on graphs with maximum degree at most two, we can solve Minimum Shared Edges in polynomial time. Let G be a graph with $\Delta(G) \leq 2$ and $s, t \in V(G)$ two vertices in G. If the number p of paths is greater than the value of a maximum s-t flow in G with unit edge capacities, then the number of edges that have to be shared is at least the length of a shortest s-t path. Thus, an instance (G, s, t, p, k) of MSE with $\Delta(G) \leq 2$ is a yes-instance if and only if $k \geq \mathrm{dist}_G(s, t)$ or $p \leq |f|$, where $|f|$ is the value of a maximum s-t flow in G with unit edge capacities. However, it remains an open problem whether Minimum Shared Edges can be solved in polynomial time on graphs with maximum degree three or four.

In the remainder of this section, we show that Short Minimum Shared Edges (SMSE) is $W[2]$-hard when parameterized by the number k of shared edges and the upper bound λ on the length of the paths.

Theorem 5.3. SMSE(k, λ) *is* $W[2]$*-hard.*

We give a shortened proof of Theorem 5.3, since many of the arguments used in the proof of Theorem 5.1 can be transferred.

Proof. Let $(X, \mathcal{C}, \ell)$ be an instance of Set Cover. We reduce instance $(X, \mathcal{C}, \ell)$ of SC(ℓ) to an instance (G, s, t, p, k, λ) of SMSE(k, λ) with $p = |X| + |\mathcal{C}|$, $k = \ell$ and $\lambda = \ell + 3$ as follows.

Construction. Initially, let G be an empty graph, that is, $V(G) = E(G) = \emptyset$. First, we add the vertcies s and t to $V(G)$. Next, we add the following vertex sets to $V(G)$:

- $V_X := \{v_i \mid i \in X\}$, the set of vertices corresponding to the elements of X,
- $V_C := \{w_j \mid C_j \in \mathcal{C}\}$ and $V_s := \{u_j \mid C_j \in \mathcal{C}\}$, two sets of vertices corresponding to the sets in $\mathcal{C}$.

We connect each vertex in V_s and V_X with vertex s via an $(\ell + 1)$-chain. Finally, we add the following edge sets to $E(G)$:

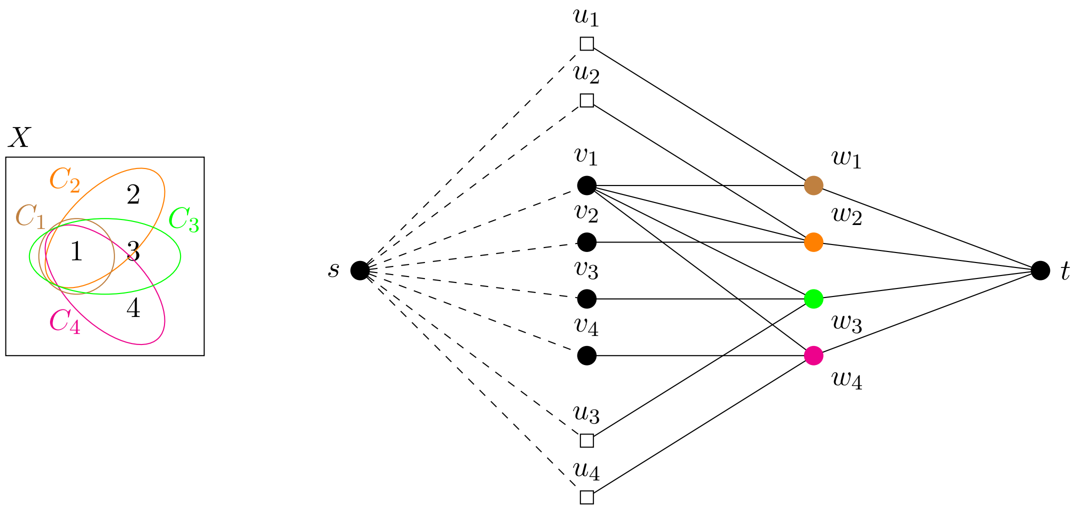

Figure 5.4: Example of the reduction from Set Cover to Short Minimum Shared Edges. On the left-hand side, an instance $(X, \mathcal{C}, \ell)$ of Set Cover is shown. On the right-hand side, the constructed graph in the reduced instance (G, s, t, p, k, λ) of Short Minimum Shared Edges is shown with $p = 8$, $k = \ell$, and $\lambda = \ell + 3$. Dashed lines represent $(\ell + 1)$-chains.

- $E_1 := \{\{v_i, w_j\} \mid (i \in X) \wedge (C_j \in \mathcal{C}) \wedge (i \in C_j)\}$,
- $E_2 := \{\{u_j, w_j\} \mid C_j \in \mathcal{C}\}$,
- $E_3 := \{\{w_j, t\} \mid C_j \in \mathcal{C}\}$.

Note that by construction, every shortest s-t path in G has length $\ell + 3$. In Figure 5.4, we provide an example for the described construction on an example instance of Set Cover.

Correctness. Suppose that we have $p = |X| + |\mathcal{C}|$ s-t paths in G of length at most $\lambda = \ell + 3$ that share at most $k = \ell$ edges. We show that we can construct a set cover $\mathcal{C}' \subseteq \mathcal{C}$ of X with $|\mathcal{C}'| \leq \ell$.

Since every $(\ell + 1)$-chain contains $(\ell + 1)$ edges, every $(\ell + 1)$-chain in G appears in at most one s-t path. Since vertex s is incident with exactly p $(\ell + 1)$-chains, each of the $(\ell + 1)$-chains incident with vertex s appears in exactly one s-t path. As a consequence, each of the vertices in V_s and V_X appears in at least one s-t path. Since each vertex in V_X is connected with s via an $(\ell + 1)$-chain and vertices in V_C, each of the s-t paths containing a vertex in V_X contains a vertex in V_C. Since each vertex in V_C is connected with exactly one vertex in V_s, each of the vertices in V_C appears in at least one s-t path. Note that the length of each of the p s-t paths is at most $\lambda + 3$, and each shortest s-t path in G has length $\lambda + 3$. Thus, each of the p s-t paths is a shortest s-t path in G. We show that every s-t path contains exactly one vertex in $V_X \cup V_s$. For every vertex $v \in V_X \cup V_s$ holds that $\mathrm{dist}_G(v, t) = 2$ and $\mathrm{dist}_G(s, v) = (\ell + 1)$. For every two vertices $v, v' \in V_X \cup V_s$ holds that $\mathrm{dist}_G(v, v') \geq 2$. Suppose that there is an s-t path P that contains two vertices $v, v' \in V_X \cup V_s$. Since the endpoints of P are the vertices s and t, there exists a subpath P_1 in P connecting s with one of the vertices v or v', a subpath P_2 in P connecting v with v' and a subpath P_3 connecting one of the vertices v or v' with t. It holds that

$$|E(P)| \geq |E(P_1)| + |E(P_2)| + |E(P_3)| \geq (\ell + 1) + 2 + 2 = \ell + 5.$$

This is a contradiction to the fact that P has length at most $\ell + 3$. Hence, each vertex in V_s and V_X appears in exactly one s-t path. Moreover, no edge connecting the vertices in $V_s \cup V_X$ with the vertices in V_C is shared.

Let $V' := \{w \in V_C \mid \{w, t\} \text{ is a shared edge}\}$. We claim that if vertex $v_i \in V_X$ and vertex $w_j \in V_C$ appear in an s-t path, then $w_j \in V'$. Consider an s-t path containing the vertices v_i and w_j. Since w_j appears in an s-t path containing a vertex in V_s, and every s-t path contains exactly one vertex in $V_s \cup V_X$, vertex w_j appears in at least two s-t paths. Since vertex w_j is only connected with vertices in $V_s \cup V_X \cup \{t\}$, and every s-t path contains exactly one vertex in $V_s \cup V_X$, each of the at least two s-t paths containing vertex w_j uses edge $\{w_j, t\}$. It follows that $w_j \in V'$.

We claim that $\mathcal{C}' := \{C_j \in \mathcal{C} \mid w_j \in V'\}$ is a set cover of X of size at most ℓ. Since the p s-t paths share at most $k = \ell$ edges, it follows that $|V'| \leq k$. We know that for each vertex in V_X, there is exactly one vertex in V_C such that both vertices appear in one of the s-t paths. In addition, if a vertex in V_X and a vertex in V_C appear on an s-t path, then the vertex in V_C is contained in V'. Therefore, for each $i \in X$, the corresponding vertex $v_i \in V_X$ appears in an s-t path containing a vertex in V'. Thus, each $i \in X$ is covered by a set in $\mathcal{C}'$, and hence, $\mathcal{C}'$ is a set cover of X of size at most ℓ.

Conversely, suppose that we have a set cover $\mathcal{C}'$ of X of size at most ℓ. We show that we can construct p s-t paths in G of length at most $\lambda = \ell + 3$ that share at most $k = \ell$ edges.

First, we construct $|\mathcal{C}|$ shortest s-t paths as follows. For each vertex $u_j \in V_s$, we construct an s-t path that contains the $(\ell + 1)$-chain connecting s with u_j and the edges $\{u_j, w_j\}$ and $\{w_j, t\}$. Each of these $|\mathcal{C}|$ s-t paths has length $\ell + 3$ and they do not share any edge.

Next, we construct $|X|$ shortest s-t paths as follows. For each $i \in X$, we construct an s-t path that contains the $(\ell + 1)$-chain connecting s with v_i and the edges $\{v_i, w_j\}$ and $\{w_j, t\}$, where w_j corresponds to a set $C_j \in \mathcal{C}'$ with $i \in C_j$. Since $\mathcal{C}'$ is a set cover of X, such a vertex w_j exists for vertex v_i. Each of these $|X|$ s-t paths has length $\ell + 3$. Since there are at most $|\mathcal{C}'| \leq k$ sets in the set cover $\mathcal{C}'$, at most k edges connecting the vertices in V_C with vertex t are shared by the p constructed s-t paths of length $\lambda = \ell + 3$. $\square$

6 An Efficient Algorithm for Small Treewidth

In the previous section, we showed that MSE(k) is $W[2]$-hard, and that Minimum Shared Edges remains NP-hard on graphs with maximum degree at least five. By Section 3, we know that MSE(p, k) is fixed-parameter tractable. This motivates a closer study of the parameter p. In this section, we provide a dynamic program that solves MSE(p, ω) in FPT-time, where ω is the width of a given tree decomposition of the input graph. We make use of our dynamic program in Section 7 to prove that MSE(p) is fixed-parameter tractable.

Many problems can be solved efficiently on graphs with bounded treewidth assuming that a tree decomposition of a graph is given. Ye et al. [19] showed that Minimum Shared Edges is fixed-parameter tractable with respect to the number p of paths and an upper bound on the treewidth of the graph. Ye et al. [19] provided an algorithm that solves MSE(p, ω) in FPT-time, that is, with respect to the number p of paths and a upper bound $\omega \geq \mathrm{tw}(G)$ on the treewidth of graph G. Aoki et al. [1] provided an algorithm that solves Minimum Vulnerability parameterized by the number p of paths and a upper bound ω on the treewidth of the input graph in FPT-time, which can be used to solve MSE(p, ω) in FPT-time, since MV generalizes MSE. Both algorithms are dynamic programs running on a nice tree decomposition. In contrast, the dynamic program that we present next runs on a nice tree decomposition with introduce edge nodes. The running times of the algorithms provided by Ye et al. [19] and Aoki et al. [1] are both linear in p and double exponential in ω, while the running time of our algorithm is exponential in p and exponential in ω. Hence, for high values of ω, our algorithm is preferable. The main result of this section is the following.

Theorem 6.1. *Let G be a graph with $s, t \in V(G)$ given together with a tree decomposition of width ω. Let $p \in \mathbb{N}$ be an integer. Then the minimum number of shared edges for p s-t paths can be computed in $O(p \cdot (\omega + 4)^{3 \cdot p \cdot (\omega+3)+4} \cdot |V(G)|)$ time.*

In the following, we prepare the proof of Theorem 6.1. We modify the given tree decomposition to a nice tree decomposition with introduce edge nodes such that each bag contains the sink and the source node. Cygan et al. [5] showed that given a tree decomposition, a nice tree decomposition with introduce edge nodes of equal width can be computed in polynomial time. We remark that adding the vertices s and t to each bag of the nice tree decomposition with introduce edge nodes can be done in linear time in the number of nodes in the tree decomposition and increases the width of the tree decomposition by at most two. Next, we define a graph for every node in the tree decomposition, as well as a partial solution for a node in the tree decomposition. Finally, we describe the dynamic program on the tree decomposition, prove its correctness and discuss its running time.

Let G be graph and let $s, t \in V(G)$ be two vertices in G. Let $\mathbb{T}'$ be a nice tree decomposition of G with introduce edge nodes, as defined in Section 2. We add the vertices s and t to every bag of $\mathbb{T}'$. Let $\mathbb{T}$ be the tree decomposition after adding s and t to every bag. We remark that adding s and t to every bag increases the width of the tree decomposition $\mathbb{T}'$ by at most two, and thus, tree decomposition $\mathbb{T}$ has width $\omega(\mathbb{T}) \leq \omega(\mathbb{T}') + 2$.

For each node α in the tree decomposition $\mathbb{T}$ of G, we define V_α as the set of vertices that are introduced in the subtree rooted at node α, and E_α as the set of edges that are introduced in the subtree rooted at node α. In other words, a vertex $v \in V(G)$ is in V_α if and only if there exists at least one introduce vertex node in the subtree rooted at node α that introduced vertex v. As a special case, since the vertices s and t are contained in every bag, we consider s and t as introduced by each leaf node. An edge $e \in E(G)$ is in E_α if and only if there exists an introduce edge node in the subtree rooted at node α that introduced edge e. Recall that there is a unique introduce edge node for every edge of graph G. We define $G_\alpha := (V_\alpha, E_\alpha)$ as the graph for node α. For every leaf node α in $\mathbb{T}$, we set $V_\alpha = \{s, t\}$ and $E_\alpha = \emptyset$.

In Figure 6.1, we show for an example graph G (upper-left) with $s, t \in V(G)$ a nice tree decomposition with introduce edge nodes and vertices s and t contained in each bag. Moreover, we illustrate the graphs as defined above for some tagged nodes in the tree decomposition. In the center of the figure, the modified nice tree decomposition with introduce edge nodes is shown. The graphs around the tree decomposition are the graphs for some tagged nodes, for example, the graph G_α is the graph for node α in the tree decomposition. For following examples and illustrations, we make use of this example throughout this section, and thus, we denote by $\mathbb{T}^*$ the tree decomposition in Figure 6.1.

We define a set of p forests in G_α as a *partial solution* L_α for node α. We recall that a path is also a tree and a tree is also a forest. Therefore, instead of asking for p s-t paths that share at most k edges, we can ask for p s-t forests that share at most k edges, where an s-t forest is a forest that contains at least one tree connecting vertices s and t. Note that every forest that contains a tree containing both vertices s and t can be reduced to an s-t path. A partial solution L_α has a cost value $c(L_\alpha)$, which is equal to the number of edges in G_α that appear in at least two of the p forests in L_α.

For each node α in the tree decomposition $\mathbb{T}$ of G, we consider p-tuples of pairs $\mathcal{X}^\alpha := (\mathcal{Y}_q^\alpha, Z_q^\alpha)_{q=1,\dots,p}$, where for each $q \in [p]$, $Z_q^\alpha \subseteq B_\alpha$ together with $\mathcal{Y}_q^\alpha \subseteq 2^{B_\alpha}$ is a partition of B_α, that is,

(i) $Z_q^\alpha \cup \bigcup_{M \in \mathcal{Y}_q^\alpha} M = B_\alpha$,

(ii) for all $X, Y \in \mathcal{Y}_q^\alpha \cup \{Z_q^\alpha\}$ with $X \neq Y$ holds $X \cap Y = \emptyset$.

We say that $\mathcal{X}^\alpha$ is a *signature* for node α. For each $q \in [p]$, we call the pair $(\mathcal{Y}_q^\alpha, Z_q^\alpha)$ a *segmentation* of the vertex set B_α. We call each $M \in \mathcal{Y}_q^\alpha$ a *segment* of the segmentation q

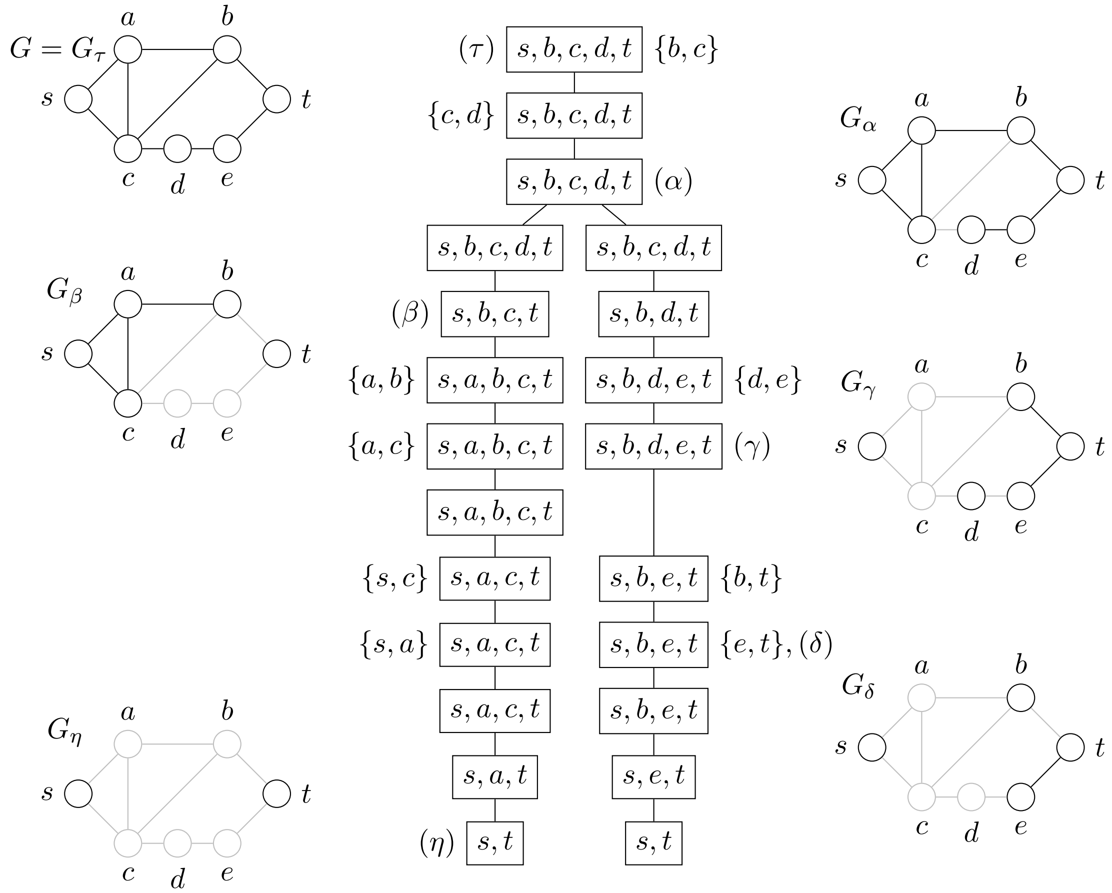

Figure 6.1: Example for a nice tree decomposition with introduce edge nodes and vertices s and t contained in every bag on an example graph G (top-left). Node τ is the root node in the tree decomposition. The graphs around the tree decomposition correspond to the graphs for the tagged nodes in the tree decomposition. For example, graph G_γ corresponds to the graph for node γ in the tree decomposition.

and we call Z_q^α the *zero*-segment of the segmentation q.

We say that the signature $\mathcal{X}^\alpha$ is a *valid* signature for node α if there is a partial solution L_α for node α such that for each $q \in [p]$, the zero-segment Z_q^α is the set of nodes in B_α that do not appear in the forest with index q and for each set $M \in \mathcal{Y}_q^\alpha$, there is a tree S in the forest with index q such that $M = B_\alpha \cap V(S)$. In other words, the sets in $\mathcal{Y}_q^\alpha$ correspond to connected components in the forest with index q of the partial solution. We say that $\mathcal{X}^\alpha$ is a signature *induced* by the partial solution L_α, if $\mathcal{X}^\alpha$ is a valid signature for node α and the partial solution L_α validates $\mathcal{X}^\alpha$. In this case, for each $q \in [p]$, the pair $(\mathcal{Y}_q^\alpha, Z_q^\alpha)$ is an *induced* segmentation. We remark that given $\mathcal{X}^\alpha$, there can be exactly one, more than one or no partial solution with signature $\mathcal{X}^\alpha$. Given a partial solution L_α for G_α, there is exactly one signature induced by L_α. Let $\mathcal{X}^\alpha$ be a signature for node α such that there is no partial solution for G_α that induces the signature $\mathcal{X}^\alpha$, then we say that $\mathcal{X}^\alpha$ is an *invalid* signature.

Given $\mathcal{X}^\alpha$ and $B \subseteq B_\alpha$, we define $\mathcal{X}^\alpha|_B$ as the signature $\mathcal{X}^\alpha$ with sets restricted to the set B, that is, $Z_q^\alpha \cap B$ and $M_q^\alpha \cap B$ for all $M_q^\alpha \in \mathcal{Y}_q^\alpha$ and for all $q \in [p]$.

Let $\mathbb{T} = (T_\mathbb{T}, (B_\alpha)_{\alpha \in V(T_\mathbb{T})})$ be a nice tree decomposition of G with introduce edge nodes and vertices s and t added to every bag. Let $\omega := \omega(\mathbb{T}) \geq \mathrm{tw}(G)$ be the width of $\mathbb{T}$, upper-bounding the treewidth of graph G. We consider the table T in the following dynamic program that we apply bottom-up on the tree decomposition $\mathbb{T}$, that is, we start to fill the entries of the table T at the leaf nodes of the tree decomposition $\mathbb{T}$ and we traverse the tree of the tree decomposition from the leaves to the root. For a node α in the tree decomposition $\mathbb{T}$ and a signature $\mathcal{X}^\alpha$ for node α, the entry $T[\alpha, \mathcal{X}^\alpha]$ is defined as

$$T[\alpha, \mathcal{X}^\alpha] := \begin{cases} \min c(L_\alpha), & \text{if } \mathcal{X}^\alpha \text{ is a valid signature,} \\ \infty, & \text{otherwise,} \end{cases}$$

where the minimum is taken over all partial solutions L_α in G_α such that L_α induces the signature $\mathcal{X}^\alpha$.

For each type of node in $\mathbb{T}$, we define a rule on how to fill each entry in T. In addition, for the types introduce vertex node, forget node, introduce edge node and join node, we prove the correctness of each rule, and we discuss the running time for applying each rule and filling all entries in T for the given type of node. We start with the leaf nodes of the tree decomposition $\mathbb{T}$.

Leaf Node. Let α be a leaf node of $\mathbb{T}$. Since s and t appear in every bag of $\mathbb{T}$, it holds that $B_\alpha = \{s, t\}$. We set

$$T[\alpha, \mathcal{X}^\alpha] := \begin{cases} 0, & \text{if } \mathcal{Y}_q^\alpha = \{\{s\}, \{t\}\} \text{ for all } q = 1, \ldots, p, \\ \infty, & \text{otherwise.} \end{cases}$$

We recall that $V_\alpha = \{s, t\}$ and $E_\alpha = \emptyset$ for every leaf node α in $\mathbb{T}$. Since there is no edge in E_α, the vertices s and t cannot appear together in one tree in a forest in any partial solution in G_α, and thus, the vertices s and t cannot appear together in one segment in any segmentation of a signature for a leaf node. Since in any solution to our problem, s and t appear in each of the p forests, we can set s and t as segments of all p segmentations.

Introduce Vertex Node. Let α be an introduce vertex node of $\mathbb{T}$ and let β be the child node of α with $B_\alpha \setminus B_\beta = \{v\}$. Two signatures $\mathcal{X}^\alpha$ and $\mathcal{X}^\beta$ are *compatible* if

$\mathcal{X}^\alpha|_{B_\beta} = \mathcal{X}^\beta$, and $v \in Z_q^\alpha$ or $\{v\} \in \mathcal{Y}_q^\alpha$ for each $q \in [p]$. We claim that

$$T[\alpha, \mathcal{X}^\alpha] = \begin{cases} \min_{\mathcal{X}^\beta \text{ compatible with } \mathcal{X}^\alpha} T[\beta, \mathcal{X}^\beta], & \text{if it exists } \mathcal{X}^\beta \text{ compatible with } \mathcal{X}^\alpha, \\ \infty, & \text{otherwise.} \end{cases}$$

Since α is an introduce vertex node for vertex $v \in V(G)$, no edge incident with v is introduced in any node in the subtree rooted at node α, and thus, vertex v is an isolated vertex in G_α. As a consequence, in every forest in all partial solutions for G_α, the introduced vertex v is either a single-vertex tree or does not appear in the forest since v cannot be connected to any vertex in G_α. A single-vertex tree is a tree that contains exactly one vertex and does not contain any edge.

Correctness. "$\geq$": Let L_α be a partial solution for G_α with signature $\mathcal{X}^\alpha$ such that $T[\alpha, \mathcal{X}^\alpha] = c(L_\alpha)$. We construct a partial solution L_β for G_β and a signature $\mathcal{X}^\beta$ such that L_β induces $\mathcal{X}^\beta$ and $\mathcal{X}^\beta$ is compatible with $\mathcal{X}^\alpha$. For each forest in the partial solution L_α, vertex v is either a single-vertex tree or does not appear in the forest, since there is no edge incident with vertex v in G_α. If vertex v appears as single-vertex tree in any forest in L_α, deleting v yields a forest in G_β. We define the partial solution L_β as L_α restricted to V_β, which are the forests without the isolated vertex v. The partial solution L_β is a partial solution for G_β with valid signature $\mathcal{X}^\beta := \mathcal{X}^\alpha|_{B_\beta}$. Signature $\mathcal{X}^\beta$ is compatible with signature $\mathcal{X}^\alpha$. It follows that

$$T[\alpha, \mathcal{X}^\alpha] = c(L_\alpha) = c(L_\beta) \geq T[\beta, \mathcal{X}^\beta] \geq \min_{\mathcal{X}'^\beta \text{ compatible with } \mathcal{X}^\alpha} T[\beta, \mathcal{X}'^\beta].$$

"$\leq$": Let L_β be a partial solution for G_β with signature $\mathcal{X}^\beta$ compatible with signature $\mathcal{X}^\alpha$ such that $T[\beta, \mathcal{X}^\beta] = c(L_\beta)$ and $T[\beta, \mathcal{X}^\beta] = \min_{\mathcal{X}'^\beta \text{ compatible with } \mathcal{X}^\alpha} T[\beta, \mathcal{X}'^\beta]$. We construct a partial solution L_α for G_α with signature $\mathcal{X}^\alpha$. For each $q \in [p]$, if $v \in Z_q^\alpha$, then we do not add v to the forest with index q in L_β. If v is a single segment in the segmentation q, i.e. $\{v\} \in \mathcal{Y}_q^\alpha$, then we add v as a single-vertex tree to the forest with index q in L_β. Since L_β is a partial solution for G_β, the constructed L_α is a partial solution for G_α with signature $\mathcal{X}^\alpha$. It follows that

$$\min_{\mathcal{X}'^\beta \text{ compatible with } \mathcal{X}^\alpha} T[\beta, \mathcal{X}'^\beta] = T[\beta, \mathcal{X}^\beta] = c(L_\beta) = c(L_\alpha) \geq T[\alpha, \mathcal{X}^\alpha].$$

Running time. For each signature $\mathcal{X}^\alpha$, we check for all $q \in [p]$ whether $v \in Z_q^\alpha$ or $\{v\} \in \mathcal{Y}_q^\alpha$ in $O(p \cdot |B_\alpha|)$ time. If for all $q \in [p]$ holds that $v \in Z_q^\alpha$ or $\{v\} \in \mathcal{Y}_q^\alpha$, then we check all signatures $\mathcal{X}^\beta$ for node β for compatibility with signature $\mathcal{X}^\alpha$, that means, we check if $\mathcal{X}^\alpha|_{B_\beta} = \mathcal{X}^\beta$. This can be done in $O(p \cdot |B_\alpha|^2)$ time. Since there are $O((|B_\beta| + 1)^{p \cdot |B_\beta|})$ signatures for node β and $|B_\beta| \leq |B_\alpha|$, the running time for this step is in $O(p \cdot (|B_\alpha| + 1)^{p \cdot |B_\alpha| + 2})$. Since there are $O((|B_\alpha| + 1)^{p \cdot |B_\alpha|})$ signatures for node α and $|B_\beta| \leq |B_\alpha| \leq \omega + 1$, the overall running time for filling the entries in T for an introduce vertex node is in $O(p \cdot (\omega + 2)^{2 \cdot p \cdot (\omega + 1) + 2})$.

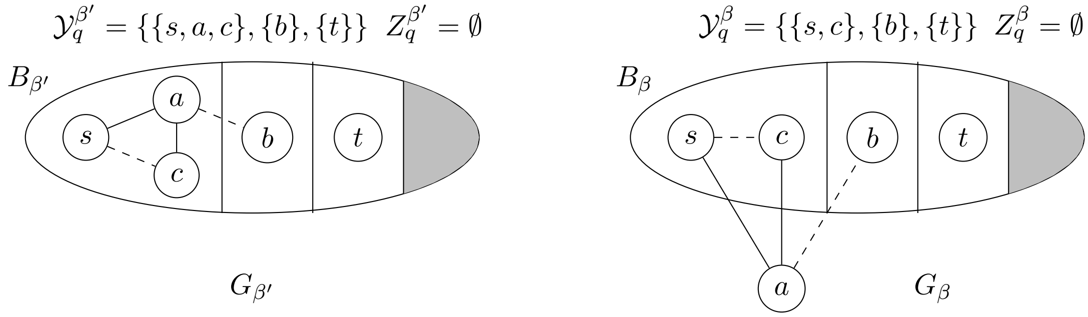

Figure 6.2: Example for a segmentation q of two compatible signatures $\mathcal{X}^\beta$ and $\mathcal{X}^{\beta'}$ for a forget node β with child node β' in $\mathbb{T}^*$.

Forget Node. Let α be a forget node of $\mathbb{T}$ and let β be the child node of α with $B_\beta \backslash B_\alpha = \{v\}$. Two signatures $\mathcal{X}^\alpha$ and $\mathcal{X}^\beta$ are *compatible* if $\mathcal{X}^\alpha = \mathcal{X}^\beta|_{B_\alpha}$. We claim that

$$T[\alpha, \mathcal{X}^\alpha] = \min_{\mathcal{X}^\beta \text{ compatible with } \mathcal{X}^\alpha} T[\beta, \mathcal{X}^\beta].$$

Since node α is a forget node for the vertex $v \in V(G)$, all edges incident with v have been introduced in the subtree rooted at α. Therefore, every possible way of v appearing in a forest has been considered. We remark that G_α and G_β are equal.

In Figure 6.2, we provide an example for a segmentation q of two compatible signatures $\mathcal{X}^\beta$ and $\mathcal{X}^{\beta'}$ for a forget node β with child node β' in $\mathbb{T}^*$ of Figure 6.1. All lines connecting two vertices correspond to the edges in the graphs $G_{\beta'}$ and G_β, where only the solid lines are the edges in the partial solutions that induce the heading segmentations. Node β forgets vertex a. The vertices $s, a, c \in V_{\beta'}$ form a segment in $\mathcal{Y}_q^{\beta'}$. As node β forgets vertex a, the vertices $s, c \in V_\beta$ form a segment in $\mathcal{Y}_q^\beta$, since they are connected via the vertex a.

Correctness. "$\geq$": Let L_α be a partial solution for G_α with signature $\mathcal{X}^\alpha$ such that $T[\alpha, \mathcal{X}^\alpha] = c(L_\alpha)$. We construct a partial solution L_β for G_β and a signature $\mathcal{X}^\beta$ such that L_β induces $\mathcal{X}^\beta$ and $\mathcal{X}^\beta$ is compatible with $\mathcal{X}^\alpha$. Since $G_\alpha = G_\beta$, the set of p forests $L_\beta := L_\alpha$ is a partial solution for G_β. We set $\mathcal{X}^\beta|_{B_\alpha} := \mathcal{X}^\alpha$. For each $q \in [p]$, if vertex v does not appear in the forest with index q, then we set $Z_q^\beta := Z_q^\alpha \cup \{v\}$ and $\mathcal{Y}_q^\beta := \mathcal{Y}_q^\alpha$. If vertex v appears in the forest with index q, then we set $Z_q^\beta := Z_q^\alpha$, and we add v to the segmentation $\mathcal{Y}_q^\beta$ as follows. If vertex v appears as a single-vertex tree in the forest with index q, then we add $\{v\}$ to $\mathcal{Y}_q^\beta$. If vertex v appears in a tree with vertices in $M \in \mathcal{Y}_q^\alpha$, then we set $\mathcal{Y}_q^\beta := (\mathcal{Y}_q^\alpha \backslash \{M\}) \cup \{M \cup \{v\}\}$. Signature $\mathcal{X}^\beta$ is compatible with signature $\mathcal{X}^\alpha$, and the partial solution L_β induces $\mathcal{X}^\beta$. It follows that

$$T[\alpha, \mathcal{X}^\alpha] = c(L_\alpha) = c(L_\beta) \geq T[\beta, \mathcal{X}^\beta] \geq \min_{\mathcal{X}'^\beta \text{ compatible with } \mathcal{X}^\alpha} T[\beta, \mathcal{X}'^\beta].$$

"$\leq$": Let L_β be a partial solution for G_β with signature $\mathcal{X}^\beta$ compatible with $\mathcal{X}^\alpha$ such that $T[\beta, \mathcal{X}^\beta] = c(L_\beta)$ and $T[\beta, \mathcal{X}^\beta] = \min_{\mathcal{X}'^\beta \text{ compatible with } \mathcal{X}^\alpha} T[\beta, \mathcal{X}'^\beta]$. We construct a partial solution for G_α that induces $\mathcal{X}^\alpha$. Since $G_\alpha = G_\beta$, we set $L_\alpha := L_\beta$ as the partial solution L_α for G_α. Since $\mathcal{X}^\alpha = \mathcal{X}^\beta|_{B_\alpha}$, the partial solution L_α induces $\mathcal{X}^\alpha$. It follows that

$$\min_{\mathcal{X}'^\beta \text{ compatible with } \mathcal{X}^\alpha} T[\beta, \mathcal{X}'^\beta] = T[\beta, \mathcal{X}^\beta] = c(L_\beta) = c(L_\alpha) \geq T[\alpha, \mathcal{X}^\alpha].$$

Running time. For a signature $\mathcal{X}^\alpha$, we check whether $\mathcal{X}^\alpha = \mathcal{X}^\beta|_{B_\alpha}$ for all signatures $\mathcal{X}^\beta$ for node β. This can be done in $O(p \cdot (|B_\beta| + 1)^{p \cdot |B_\beta| + 2})$ time. Since $|B_\alpha| \leq |B_\beta| \leq \omega + 1$, the overall running time for filling all entries in T for a forget node is in $O(p \cdot (\omega + 2)^{2 \cdot p \cdot (\omega+1)+2})$.

Introduce Edge Node. Let α be an introduce edge node of $\mathbb{T}$, let β be the child node of α, and let $e = \{v, w\}$ be the edge introduced by node α. Two signatures $\mathcal{X}^\alpha$ and $\mathcal{X}^\beta$ are *compatible* if for each $q \in [p]$, one of the following conditions holds:

(i) $\mathcal{Y}_q^\alpha = \mathcal{Y}_q^\beta$, or

(ii) $\mathcal{Y}_q^\alpha = (\mathcal{Y}_q^\beta \setminus \{M_1, M_2\}) \cup \{M_1 \cup M_2\}$ with $M_1, M_2 \in \mathcal{Y}_q^\beta$, $M_1 \neq M_2$, and $v \in M_1$ and $w \in M_2$.

If $\mathcal{X}^\alpha$ and $\mathcal{X}^\beta$ are compatible, then let $Q \subseteq [p]$ be the set of indices such that for all $q \in Q$ (ii) holds and for all $q \in [p] \setminus Q$ (i) holds. We say that $\mathcal{X}^\alpha$ and $\mathcal{X}^\beta$ are *share-compatible* if $|Q| \geq 2$. We claim that

$$T[\alpha, \mathcal{X}^\alpha] = \min_{\mathcal{X}^\beta \text{ compatible with } \mathcal{X}^\alpha} \left(T[\beta, \mathcal{X}^\beta] + \begin{cases} 1, & \text{if } \mathcal{X}^\beta \text{ and } \mathcal{X}^\alpha \text{ are share-compatible,} \\ 0, & \text{otherwise} \end{cases} \right).$$

In other words, two signatures $\mathcal{X}^\alpha$ for node α and $\mathcal{X}^\beta$ for node β are compatible, if and only if for all $q \in [p]$, either by (i) it holds that the segmentation q in $\mathcal{X}^\alpha$ is equal to the segmentation q of $\mathcal{X}^\beta$, or by (ii) it holds that the segmentation q of $\mathcal{X}^\alpha$ is the result of merging two segments in the segmentation q of $\mathcal{X}^\beta$, where none of the two segments is the zero-segment, and vertex v is in the one segment, and vertex w is in the other segment. This corresponds to connecting two trees by edge e in the forest with index q, where v is in the one tree and w in the other tree. Note that connecting two vertex-disjoint trees by exactly one edge yields a tree. The deletion of edge e in every forest of a partial solution for G_α that includes the edge e yields a partial solution for G_β. We remark that $G_\alpha = G_\beta + \{e\}$, that is, G_α differs from G_β only by the additional edge e.

In Figure 6.3, we provide an example for a segmentation q of two compatible signatures $\mathcal{X}^\delta$ and $\mathcal{X}^{\delta'}$ of an introduce edge node δ with child node δ' in $\mathbb{T}^*$. Graph $G_{\delta'}$ does

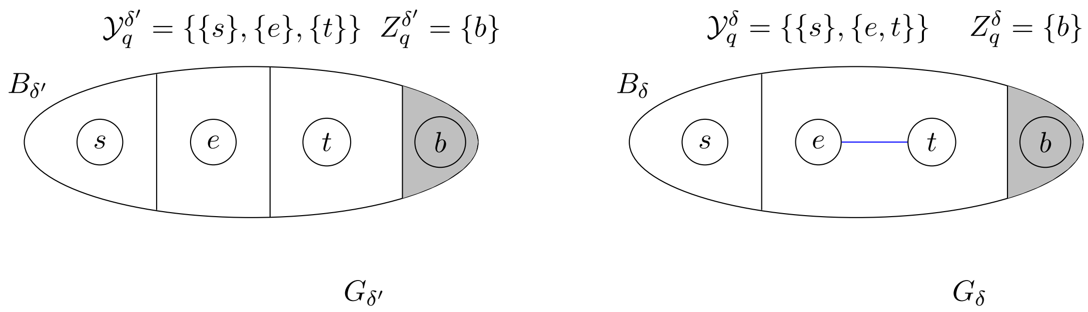

Figure 6.3: Example for a segmentation q of two compatible signatures $\mathcal{X}^\delta$ and $\mathcal{X}^{\delta'}$ of an introduce edge node δ with child node δ' in $\mathbb{T}^*$.

not contain any edge. Edge $\{e, t\}$ is introduced by node δ. This allows to connect the segments containing vertex e on the one hand, and vertex t on the other hand, using edge $\{e, t\}$.

Correctness. "$\geq$": Let L_α be a partial solution for G_α with signature $\mathcal{X}^\alpha$ such that $T[\alpha, \mathcal{X}^\alpha] = c(L_\alpha)$. We construct a partial solution L_β for G_β and a signature $\mathcal{X}^\beta$ such that L_β induces $\mathcal{X}^\beta$ and $\mathcal{X}^\beta$ is compatible with $\mathcal{X}^\alpha$. For each $q \in [p]$, if edge e is not part of the forest with index q in L_α, then the forest is a forest in G_β as well. Then, we set $Z_q^\beta := Z_q^\alpha$ and $\mathcal{Y}_q^\beta := \mathcal{Y}_q^\alpha$. If edge e is part of the forest with index q in L_α, then deleting edge e from the forest with index q disconnects a tree of the forest such that two trees result, with v in the one tree and w in the other tree. Let $M \in \mathcal{Y}_q^\alpha$ be the segment in the segmentation q with $v, w \in M$. Let M_1, M_2 be the induced sets by splitting tree T_α in the forest with index q in L_α at edge e, that is, if T_1 and T_2 are the connected subgraphs of $T_\alpha \backslash \{e\}$, then $M_1 := V(T_1) \cap B_\alpha$ and $M_2 := V(T_2) \cap B_\alpha$. We set $\mathcal{Y}_q^\beta := (\mathcal{Y}_q^\alpha \backslash \{M\}) \cup \{M_1, M_2\}$ and $Z_q^\beta := Z_q^\alpha$. Signature $\mathcal{X}^\beta$ for node β is compatible with signature $\mathcal{X}^\alpha$ for node α.

Let L_β be the set of p forests in L_α restricted to edge set E_β. Then, L_β is a partial solution for G_β and induces signature $\mathcal{X}^\beta$. If edge e appears in more than one of the p forests in L_α, then $c(L_\beta) = c(L_\alpha) - 1$ and the signatures $\mathcal{X}^\alpha$ and $\mathcal{X}^\beta$ are share-compatible. It follows that

$$T[\alpha, \mathcal{X}^\alpha] = c(L_\alpha) = c(L_\beta) + 1 \geq T[\beta, \mathcal{X}^\beta] + 1 \geq \min_{\mathcal{X}'^\beta \text{ compatible with } \mathcal{X}^\alpha} T[\beta, \mathcal{X}'^\beta] + 1.$$

If edge e appears in at most one of the p forests in the partial solution L_α, then

$$T[\alpha, \mathcal{X}^\alpha] = c(L_\alpha) = c(L_\beta) \geq T[\beta, \mathcal{X}^\beta] \geq \min_{\mathcal{X}'^\beta \text{ compatible with } \mathcal{X}^\alpha} T[\beta, \mathcal{X}'^\beta].$$

"$\leq$": Let L_β be a partial solution for G_β with signature $\mathcal{X}^\beta$ compatible with $\mathcal{X}^\alpha$ such that $T[\beta, \mathcal{X}^\beta] = c(L_\beta)$ and $T[\beta, \mathcal{X}^\beta] = \min_{\mathcal{X}'^\beta \text{ compatible with } \mathcal{X}^\alpha} T[\beta, \mathcal{X}'^\beta]$. We construct a partial solution L_α for G_α that induces signature $\mathcal{X}^\alpha$. For each $q \in [p]$, if condition (i) holds, that is, if $\mathcal{Y}_q^\alpha = \mathcal{Y}_q^\beta$, then we set the forest with index q in L_α to the forest with index q in the partial solution L_β. In case of condition (ii), that is, if v and w belong to the same segment in the segmentation q of $\mathcal{X}^\alpha$ but are not in the same segment in the segmentation q of $\mathcal{X}^\beta$, then we add edge e to the forest with index q in the partial solution L_β and we set the resulting forest as the forest with index q in the partial solution L_α. Since the vertices v and w are in two vertex-disjoint trees in the forest with index q in L_β, adding edge e connects the two trees at the vertices v and w, which results again in a tree. The set of p forests L_α, constructed as mentioned above, is a partial solution for G_α and induces signature $\mathcal{X}^\alpha$. If the signatures $\mathcal{X}^\beta$ and $\mathcal{X}^\alpha$ are share-compatible, then the partial solution L_α is the result of adding edge e to at least two forests in L_β, and thus, the number of common edges of the forests increases by exactly one, i.e. $c(L_\beta) = c(L_\alpha) - 1$. It follows that

$$\min_{\mathcal{X}'^\beta \text{ compatible with } \mathcal{X}^\alpha} T[\beta, \mathcal{X}'^\beta] = T[\beta, \mathcal{X}^\beta] = c(L_\beta) = c(L_\alpha) - 1 \geq T[\alpha, \mathcal{X}^\alpha] - 1.$$

If the signatures $\mathcal{X}^\beta$ and $\mathcal{X}^\alpha$ are compatible but not share-compatible, then the partial solution L_α is the result of adding edge e to at most one forest in L_β. It follows that

$$\min_{\mathcal{X}'^\beta \text{ compatible with } \mathcal{X}^\alpha} T[\beta, \mathcal{X}'^\beta] = T[\beta, \mathcal{X}^\beta] = c(L_\beta) = c(L_\alpha) \geq T[\alpha, \mathcal{X}^\alpha].$$

Running time. For each signature $\mathcal{X}^\alpha$, we check all signatures $\mathcal{X}^\beta$ for node β for compatibility, that means, we need to check for each $q \in [p]$ if the segmentations are equal (i) or if the segmentation q of $\mathcal{X}^\alpha$ is derived by merging two segments in the segmentation q of $\mathcal{X}^\beta$ (ii). To check condition (i) as well as to check condition (ii) can be done in $O(p \cdot |B_\alpha|^2)$ time. Therefore, the overall running time for filling all entries in T for an introduce edge node is in $O(p \cdot (\omega + 2)^{2 \cdot p \cdot (\omega+1)+2})$.

Join Node. Let α be a join node of $\mathbb{T}$ and let β, γ be the two child nodes of α. A signature $\mathcal{X}^\alpha$ for node α and a pair of two signatures $\mathcal{X}^\beta$ for node β and $\mathcal{X}^\gamma$ for node γ are *compatible* if for all $q \in [p]$ it holds that

(i) $Z_q^\alpha = Z_q^\beta = Z_q^\gamma$,

(ii) $v, w \in M^\alpha \in \mathcal{Y}_q^\alpha$ with $v \neq w$ if and only if there exists $\ell \geq 1$ and $M_1, \dots, M_\ell \in \mathcal{Y}_q^\beta \cup \mathcal{Y}_q^\gamma$ with $|M_i \cap M_{i+1}| = 1$ for all $i = 1, \dots, \ell - 1$ and $v \in M_1$ and $w \in M_\ell$,

(iii) for all $M^\beta \in \mathcal{Y}_q^\beta$ and $M^\gamma \in \mathcal{Y}_q^\gamma$ holds $|M^\beta \cap M^\gamma| \leq 1$, and

(iv) there do not exist $\ell \geq 3$ and $M_1, \dots, M_\ell \in \mathcal{Y}_q^\beta \cup \mathcal{Y}_q^\gamma$ with $|M_i \cap M_{i+1}| = 1$ for all $i = 1, \dots, \ell - 1$ and $M_i \neq M_j$ for all $i, j \in [\ell]$, $i \neq j$, such that $v \in M_1$ and $v \in M_\ell$.

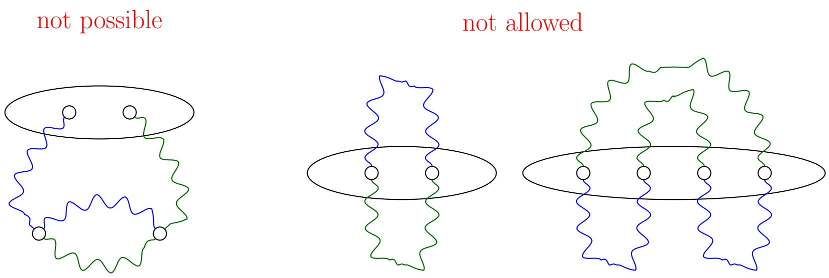

Figure 6.4: Sketch of scenarios when combining two forests with the same index in two partial solutions for the two child nodes of a join node.

We claim that

$$T[\alpha, \mathcal{X}^\alpha] = \min_{(\mathcal{X}^\beta, \mathcal{X}^\gamma) \text{ compatible with } \mathcal{X}^\alpha} (T[\beta, \mathcal{X}^\beta] + T[\gamma, \mathcal{X}^\gamma]).$$

In other words, a signature $\mathcal{X}^\alpha$ is compatible with a pair of two signatures $\mathcal{X}^\beta$ for node β and $\mathcal{X}^\gamma$ for node γ, if and only if for every $q \in [p]$ it holds that

(i) the vertices that appear in the segmentations with index q in all three signatures are the same,

(ii) every segment in the segmentation q of $\mathcal{X}^\alpha$ is a union of segments in the segmentation q in $\mathcal{X}^\beta$ and segments in the segmentation q in $\mathcal{X}^\gamma$,

(iii) every pair of segments with one segment in the segmentation q in $\mathcal{X}^\beta$ and one segment in the segmentation q in $\mathcal{X}^\gamma$ has at most one vertex in $B_\beta = B_\gamma$ in common, and

(iv) there is no chain of at least three segments in the union of the segmentations with index q in $\mathcal{X}^\beta$ and $\mathcal{X}^\gamma$ with one vertex in the first and last segment.

We say that segments $M_1, \ldots, M_\ell$, $\ell \geq 2$, form a *chain* of segments, if $|M_i \cap M_{i+1}| = 1$ for all $i = 1, \ldots, \ell - 1$ and $M_i \neq M_j$ for all $i, j \in [\ell]$, $i \neq j$.

Intuitively, (i)-(iv) define how to combine segmentations in signatures of child nodes to segmentations in a signature of a join node. The condition (ii) ensures that every forest with index q in a partial solution L_α for G_α is a union of the two forests with index q in some partial solutions L_β for G_β and L_γ for G_γ, respectively.

Figure 6.4 exemplifies three scenarios of cycle creation caused by a union of two forests, where the colors green and blue indicate each of the two forests. We used curved lines to highlight that there could be trees connecting two vertices. The scenario on the left-hand side illustrates a situation that is not possible. The scenario implies that there are vertices that have been forgotten in the subtrees rooted at each child node of the

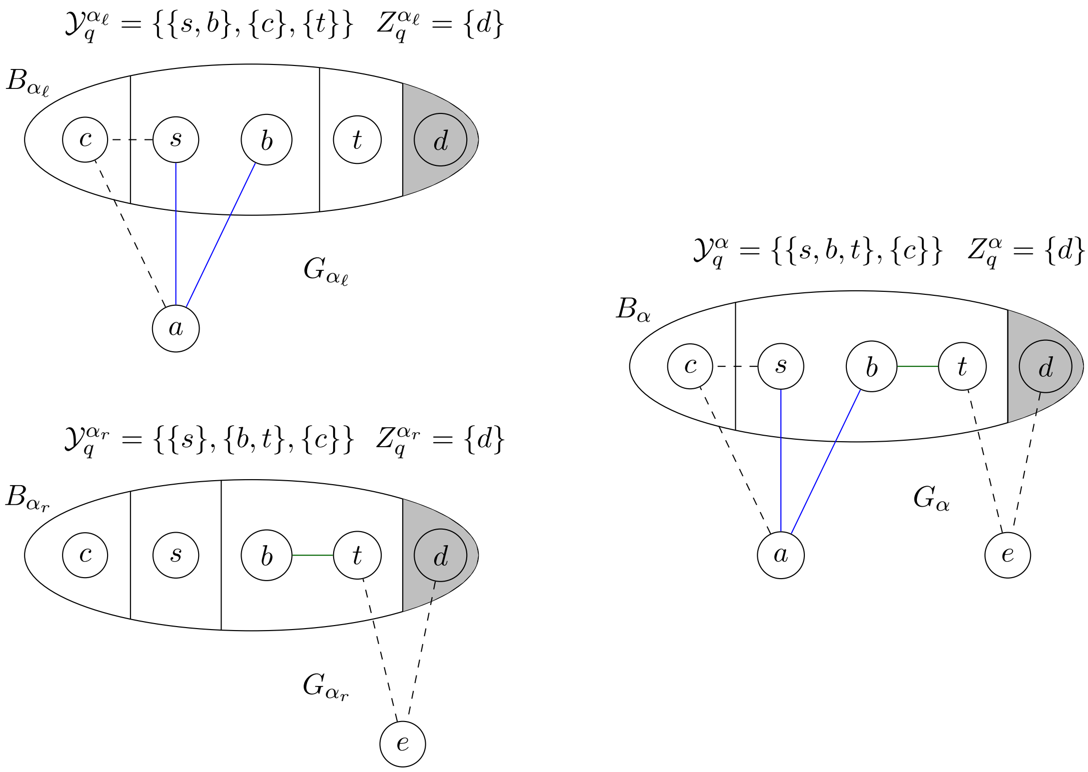

Figure 6.5: Example for a segmentation q of three compatible signatures $\mathcal{X}^\alpha$, $\mathcal{X}^{\alpha_\ell}$, and $\mathcal{X}^{\alpha_r}$ for a join node α with child nodes α_ℓ and α_r in $\mathbb{T}^*$.

tree decomposition, which is not possible by the definition of a tree decomposition (cf. Section 2). The two scenarios on the right-hand side illustrate two scenarios that are not allowed to occur by our definition of compatibility of join nodes. More precisely, conditions (iii) and (iv) ensure that none of these two scenarios occurs.

The conditions (iii) and (iv) ensure that a union of two forests in L_β and L_γ does not close a cycle. Condition (iii) prevents the following creation of cycles. If there is a tree T_β in the forest with index q in L_β and a tree T_γ in the forest with index q in L_γ that have at least two vertices in common, then the union of these two trees creates a cycle in G_α. Condition (iv) prevents the following creation of cycles. Let v be a vertex in V_α such that there exist some trees $T_1, \ldots, T_\ell$ in the forests with index q in L_β and L_γ such that $|V(T_i) \cap V(T_{i+1})| = 1$ for $i = 1, \ldots, \ell - 1$ and $v \in V(T_1)$ and $v \in V(T_\ell)$. Then the graph $T^\alpha = T_1 \cup \ldots \cup T_\ell$ as union of the trees in G_α contains a cycle and vertex v is part of a cycle in T^α.

In Figure 6.5, we provide an example for a segmentation q of three compatible signatures $\mathcal{X}^\alpha$, $\mathcal{X}^{\alpha_\ell}$, and $\mathcal{X}^{\alpha_r}$ for a join node α with child nodes α_ℓ and α_r in $\mathbb{T}^*$. The segments $\{s,b\}$ and $\{t\}$ in $\mathcal{Y}_q^{\alpha_\ell}$ together with the segments $\{s\}$ and $\{b,t\}$ in $\mathcal{Y}_q^{\alpha_r}$ form segment $\{s, b, t\}$ in $\mathcal{Y}_q^\alpha$. Note that the four conditions for compatibility hold. Condition (i) holds since $Z_q^\alpha = Z_q^{\alpha_\ell} = Z_q^{\alpha_r} = \{d\}$. Moreover, note that condition (iii) holds. According to condition (ii), note that for any pair in the segment $\{s, b, t\} \in \mathcal{Y}_q^\alpha$, the segments $\{s, b\} \in \mathcal{Y}_q^{\alpha_\ell}$ and $\{b, t\} \in \mathcal{Y}_q^{\alpha_r}$ provide a required chain of segments. Conversely, for any possible chain of segments in $\mathcal{Y}_q^{\alpha_\ell} \cup \mathcal{Y}_q^{\alpha_r}$, segment $\{s, b, t\} \in \mathcal{Y}_q^\alpha$ is the required segment in condition (ii). According to condition (iv), note that there is no chain of at least three segments in $\mathcal{Y}_q^{\alpha_\ell} \cup \mathcal{Y}_q^{\alpha_r}$ such that a vertex $v \in \{s, b, c, t\}$ appears in the first and last segment of the chain.

Correctness. "$\geq$": Let L_α be a partial solution for G_α with signature $\mathcal{X}^\alpha$ such that $T[\alpha, \mathcal{X}^\alpha] = c(L_\alpha)$. We construct a partial solution L_β for G_β, a partial solution L_γ for G_γ and two signatures $\mathcal{X}^\beta$ and $\mathcal{X}^\gamma$, such that the pair $(\mathcal{X}^\beta, \mathcal{X}^\gamma)$ is compatible with $\mathcal{X}^\alpha$, the partial solution L_β induces signature $\mathcal{X}^\beta$ and the partial solution L_γ induces signature $\mathcal{X}^\gamma$. If we restrict each forest in L_α to the edge sets E_β and E_γ, then each forest restricted to E_β is a forest in G_β and each forest restricted to E_γ is a forest in G_γ. Therefore, restricting each forest in L_α to E_β yields a partial solution L_β for G_β, and restricting each forest in L_α to E_γ yields a partial solution L_γ for G_γ. We set $\mathcal{X}^\beta$ and $\mathcal{X}^\gamma$ as the signatures induced by the partial solutions L_β and L_γ respectively.

We show that the pair of signatures $\mathcal{X}^\beta$ and $\mathcal{X}^\gamma$ is compatible with $\mathcal{X}^\alpha$. Condition (i) holds for every $q \in [p]$ since every vertex that does not appear in the forest with index q in L_α neither appears in the forests with index q nor in L_β nor in L_γ. Since the segmentations with index q are induced by L_β and L_γ, it follows that $Z_q^\alpha = Z_q^\beta = Z_q^\gamma$ for all $q \in [p]$.

Suppose that there exists a $q \in [p]$ such that condition (iii) does not hold for $q \in [p]$. This means that there exist $M^\beta \in \mathcal{Y}_q^\beta$ and $M^\gamma \in \mathcal{Y}_q^\gamma$ with $|M^\beta \cap M^\gamma| \geq 2$. Let $v, w \in M^\beta \cap M^\gamma$. Let T^β be the tree in the forest with index q in L_β corresponding to M^β and let T^γ be the tree in the forest with index q in L_γ corresponding to M^γ. Note that $v, w \in V(T^\beta) \cap V(T^\gamma)$. By our construction of L_β and L_γ, there is a tree T^α in the forest with index q in L_α, such that T^β is a subtree of T^α restricted to E_β, and T^γ is a subtree of T^α restricted to E_γ. Since $v, w \in V(T^\alpha)$, there is a v-w path in T^α using only edges in E_β and a v-w path in T^α using only edges in E_γ. Since $E_\beta \cap E_\gamma = \emptyset$, the two paths form a cycle in T^α. This is a contradiction to the fact that T^α is a tree.

For condition (ii), direction "$\Rightarrow$", we consider $q \in [p]$, $M^\alpha \in \mathcal{Y}_q^\alpha$ and $v, w \in M^\alpha$, $v \neq w$, if such a $M^\alpha \in \mathcal{Y}_q^\alpha$ exists. Segment M^α corresponds to a tree T^α in the forest with index q in L_α. Since $v, w \in M^\alpha$, the vertices v and w appear in tree T^α. Since $E_\alpha = E_\beta \cup E_\gamma$ and $E_\beta \cap E_\gamma = \emptyset$, the restriction of T^α to E_β and E_γ splits the tree in maximal subtrees $T_1, \ldots, T_\ell$ alternating by G_β and G_γ. Note that $|V(T_i) \cap V(T_j)| \leq 1$ for all $i, j \in [\ell]$, $i \neq j$, and $T^\alpha = T_1 \cup \ldots \cup T_\ell$. Let $M_1, \ldots, M_\ell \in \mathcal{Y}_q^\beta \cup \mathcal{Y}_q^\gamma$ be segments such that segment M_i corresponds to subtree T_i for all $i \in [\ell]$. We claim that if $|V(T_i) \cap V(T_j)| = 1$ for some $i \neq j$ and $u \in V(T_i) \cap V(T_j)$, then $u \in B_\alpha$.

Suppose that $u \notin B_\alpha = B_\beta = B_\gamma$. Since the trees $T_1, \ldots, T_\ell$ are maximal subtrees of tree T^α restricted to E_β and E_γ, one of the trees T_i or T_j is a tree in G_β, and the other is a tree in G_γ. Therefore, vertex u is incident with an edge in E_β and an edge in E_γ. Thus, vertex u appears in the subtree rooted at node β and in the subtree rooted at node γ. This is a contradiction to the fact that $\mathbb{T}$ is a tree decomposition, and hence, $u \in B_\alpha = B_\beta = B_\gamma$.

Moreover, if $|V(T_i) \cap V(T_j)| = 1$ for some $i \neq j$ and $u \in V(T_i) \cap V(T_j)$, then $u \in M_i$ and $u \in M_j$. If there is a $j \in [\ell]$ such that $v, w \in M_j$, then we are done. Thus, let $v \in M_{j_1}$ and $w \in M_{j_2}$ with $j_1, j_2 \in [\ell]$, $j_1 \neq j_2$. Then there exists a subset $S_1, \ldots, S_{\ell'}$ of the trees $T_1, \ldots, T_\ell$ with $\ell' \leq \ell$, $S_1 = T_{j_1}$, $S_{\ell'} = T_{j_2}$ and $|V(S_i) \cap V(S_{i+1})| = 1$ for all $i = 1, \ldots, \ell' - 1$. Let $M_{S_1}, \ldots, M_{S_{\ell'}}$ be the corresponding segments to $S_1, \ldots, S_{\ell'}$. Then, $|M_{S_i} \cap M_{S_{i+1}}| = 1$ for all $i = 1, \ldots, \ell' - 1$, $v \in M_{S_1}$ and $w \in M_{S_{\ell'}}$, and hence, direction "$\Rightarrow$" of condition (ii) is proven.

For condition (ii), direction "$\Leftarrow$", we consider $q \in [p]$, $\ell \geq 1$ and $M_1, \ldots, M_\ell \in \mathcal{Y}_q^\beta \cup \mathcal{Y}_q^\gamma$ with $|M_i \cap M_{i+1}| = 1$ for all $i = 1, \ldots, \ell - 1$, $v \in M_1$ and $w \in M_\ell$. We show that there exists a segment $M^\alpha \in \mathcal{Y}_q^\alpha$ with $v, w \in M^\alpha$. Let $T_1, \ldots, T_\ell$ be trees in the forests with index q in L_β and L_γ such that tree T_i corresponds to segment M_i for all $i \in [\ell]$. Since $|M_i \cap M_{i+1}| = 1$ for all $i = 1, \ldots, \ell - 1$, it follows that $|V(T_i) \cap V(T_{i+1})| = 1$ for all $i = 1, \ldots, \ell - 1$. Therefore, $T_1, \ldots, T_\ell$ are subtrees of a tree T^α in the forest with index q in L_α with $v, w \in V(T^\alpha)$. Let M^α be the segment corresponding to T^α. Then, segment M^α contains the vertices v and w, i.e. $v, w \in M^\alpha$, and hence, direction "$\Leftarrow$" of condition (ii) is proven.

Suppose that there exists a $q \in [p]$ such that condition (iv) does not hold for $q \in [p]$. Then there exist a vertex $v \in B_\alpha$, an integer $\ell \geq 3$ and segments $M_1, \ldots, M_\ell \in \mathcal{Y}_q^\beta \cup \mathcal{Y}_q^\gamma$ with $|M_i \cap M_{i+1}| = 1$ for all $i = 1, \ldots, \ell - 1$ and $M_i \neq M_j$ for all $i \neq j$, such that $v \in M_1$ and $v \in M_\ell$. Let $T_1, \ldots, T_\ell$ be the trees in the forests with index q in L_β and L_γ such that tree T_i corresponds to segment M_i for all $i \in [\ell]$. Note that $|V(T_i) \cap V(T_{i+1})| = 1$ for all $i = 1, \ldots, \ell - 1$, and vertex v appears in the trees T_1 and T_ℓ. For all $i = 1, \ldots, \ell - 1$, let w_i be the vertex in the intersection $V(T_i) \cap V(T_{i+1})$ of the vertex sets of the trees T_i and T_{i+1}. By construction, the union of the trees $T' := T_1 \cup \ldots \cup T_\ell$ is a subtree of a tree T^α in the forest with index q in L_α. Thus, the tuple $(v, w_1, w_2, \ldots, w_{\ell-1}, v)$ represents a cycle in T', and thus, in T^α. This is a contradiction to the fact that L_α is a partial solution for G_α, and hence, condition (iv) holds.

We conclude that the pair of signatures $\mathcal{X}^\beta$ and $\mathcal{X}^\gamma$ is compatible with $\mathcal{X}^\alpha$. Since $E_\beta \cap E_\gamma = \emptyset$, the number of edges that appear in at least two forests in L_α is the sum of the number of edges that appear in at least two forests in L_β and the number of edges that appear in at least two forests in L_γ. It follows that

$$T[\alpha, \mathcal{X}^\alpha] = c(L_\alpha) = c(L_\beta) + c(L_\gamma) \geq T[\beta, \mathcal{X}^\beta] + T[\gamma, \mathcal{X}^\gamma] \\ \geq \min_{(\mathcal{X}'^\beta, \mathcal{X}'^\gamma) \text{ compatible with } \mathcal{X}^\alpha} \left(T[\beta, \mathcal{X}'^\beta] + T[\gamma, \mathcal{X}'^\gamma] \right).$$

"$\leq$": Let L_β and L_γ be partial solutions for G_β and G_γ with signatures $\mathcal{X}^\beta$ and $\mathcal{X}^\gamma$, as pair compatible with signature $\mathcal{X}^\alpha$ for node α, such that $T[\beta, \mathcal{X}^\beta] = c(L_\beta)$, $T[\gamma, \mathcal{X}^\gamma] = c(L_\gamma)$ and $T[\beta, \mathcal{X}^\beta] + T[\gamma, \mathcal{X}^\gamma] = \min_{(\mathcal{X}'^\beta, \mathcal{X}'^\gamma) \text{ compatible with } \mathcal{X}^\alpha} (T[\beta, \mathcal{X}'^\beta] + T[\gamma, \mathcal{X}'^\gamma])$. We construct a partial solution L_α for G_α with signature $\mathcal{X}^\alpha$. We claim that for each $q \in [p]$, the union of the forests with index q in L_β and L_γ yields a forest in G_α, that induces the segmentation $(\mathcal{Y}_q^\alpha, Z_q^\alpha)$ in signature $\mathcal{X}^\alpha$.

Let $B := B_\alpha$. We remark that $B_\alpha = B_\beta = B_\gamma$ since α is a join node in $\mathbb{T}$. We claim that the intersection of the vertex sets of G_β and G_γ are only the vertices in B, that is $V_\beta \cap V_\gamma = B$. Suppose that there is a vertex $v \in (V_\beta \cap V_\gamma) \backslash B$. Then the graph induced by the node set $\{\rho \in V(T_\mathbb{T}) \mid v \in B_\rho\}$ is not connected. This contradicts the fact that $\mathbb{T}$ is a tree decomposition, and thus, $V_\beta \cap V_\gamma = B$.

Recall that for each $q \in [p]$, the zero-segments are equal in all three segmentations, that is, $Z_q^\alpha = Z_q^\beta = Z_q^\gamma$. Hence, the vertex sets in both forests with index q in L_β and L_γ are the same. In addition, we know that $E_\beta \cap E_\gamma = \emptyset$ and therefore, the two forests with index q in L_β and L_γ do not have any edge in common. We need to show that for all $q \in [p]$ the union of the forests with index q in L_β and L_γ does not contain a cycle in G_α. Suppose there is a $q \in [p]$ such that the union of the forests with index q in L_β and L_γ contains a cycle in G_α.

Case 1: There is a tree T_1 in the forest with index q in L_β and a tree T_2 in the forest with index q in L_γ, such that the union $T_0 := T_1 \cup T_2$ contains a cycle. Let $M_1 \in \mathcal{Y}_q^\beta$ and $M_2 \in \mathcal{Y}_q^\gamma$, such that segment M_1 corresponds to tree T_1 and segment M_2 corresponds to tree T_2. Since graph T_0 contains a cycle in G_α, the trees T_1 and T_2 have at least two vertices in common. Because of $V(T_1) \subseteq V_\beta$, $V(T_2) \subseteq V_\gamma$ and $V_\beta \cap V_\gamma = B$, the common vertices are in the vertex set B. This means that there are two vertices $v, w \in B$ such that $v, w \in M_1$ and $v, w \in M_2$. This contradicts condition (iii), and hence, there are no two trees in the forests with index q in L_β and L_γ such that their union contains a cycle in G_α.

Case 2: There are trees $T_1, \ldots, T_\ell$, $\ell \geq 3$, in the forests with index q in L_α and L_β, such that their union $T_0 := T_1 \cup \ldots \cup T_\ell$ contains a cycle in G_α and $T_0 \backslash T_i$ does not contain a cycle in G_α for all $i \in [\ell]$. It follows that $|V(T_i) \cap V(T_j)| \leq 1$ for all $i, j \in [\ell]$ with $i \neq j$. Let $M_1, \ldots, M_\ell \in \mathcal{Y}_q^\beta \cup \mathcal{Y}_q^\gamma$, such that segment M_i corresponds to tree T_i for all $i \in [\ell]$. Since T_0 contains a cycle in G_α, there exists an ordering π on the set $[\ell]$, such that $|V(T_{\pi(i)}) \cap V(T_{\pi(i+1)})| = 1$ for all $i = 1, \ldots, \ell' - 1$ and $|V(T_{\pi(\ell)}) \cap V(T_{\pi(1)})| = 1$. Since $V(T_i) \cap V(T_j) \subseteq B$ for all $i, j \in [\ell]$ with $i \neq j$, it follows that

$|M_{\pi(i)} \cap M_{\pi(i+1)}| = 1$ for all $i = 1, \ldots, \ell-1$. Let v be the vertex such that $\{v\} = V(T_{\pi(1)}) \cap V(T_{\pi(\ell)})$. Since $V(T_{\pi(1)}) \cap V(T_{\pi(\ell)}) \subseteq B$, the segments $M_{\pi(1)}$ and $M_{\pi(\ell)}$ contain vertex v. Altogether, this contradicts condition (iv), and hence, there are no trees $T_1, \ldots, T_\ell$, $\ell \geq 3$, in the forests with index q in G_α and G_β such that their union $T_0 = T_1 \cup \ldots \cup T_\ell$ contains a cycle in G_α.

We conclude that there are no two forests with index q in L_β and L_γ, such that their union contains a cycle, and thus, L_α is a partial solution for G_α. Moreover, by condition (ii), L_α induces signature $\mathcal{X}^\alpha$. It follows that

$$\begin{aligned} \min_{(\mathcal{X}'^\beta, \mathcal{X}'^\gamma) \text{ compatible with } \mathcal{X}^\alpha} (T[\beta, \mathcal{X}'^\beta] + T[\gamma, \mathcal{X}'^\gamma]) &= T[\beta, \mathcal{X}^\beta] + T[\gamma, \mathcal{X}^\gamma] = c(L_\beta) + c(L_\gamma) \\ &= c(L_\alpha) \geq T[\alpha, \mathcal{X}^\alpha]. \end{aligned}$$

Running time. For each signature $\mathcal{X}^\alpha$, we check all pairs of signatures $\mathcal{X}^\beta$, $\mathcal{X}^\gamma$ for node β and γ for compatibility, that means we check conditions (i)-(iv) for $O((|B_\beta|+1)^{p \cdot |B_\beta|} \cdot (|B_\gamma|+1)^{p \cdot |B_\gamma|})$ pairs of signatures with respect to the signature $\mathcal{X}^\alpha$. Let $B := B_\alpha$. Recall that $B_\alpha = B_\beta = B_\gamma$.

For each pair, we can check condition (i) in $O(p \cdot |B|^3)$ time. We can check conditions (ii)-(iv) in $O(p \cdot |B|^3)$ time as follows.

For each $q \in [p]$, we construct a graph $\hat{G}_q$ in the following way. We set $V(\hat{G}_q) := \{v_i \mid M_i \in \mathcal{Y}_q^\beta \cup \mathcal{Y}_q^\gamma\}$ and $E(\hat{G}_q) := \{\{v_i, v_j\} \in V(\hat{G}_q)^2 \mid |M_i \cap M_j| = 1,\, M_i, M_j \in \mathcal{Y}_q^\beta \cup \mathcal{Y}_q^\gamma\}$. We can construct the graph $\hat{G}_q$ in $O(|B|^3)$ time. We can check condition (iii) while constructing graph $\hat{G}_q$. If condition (iv) does not hold, then there exists a cycle in $\hat{G}_q$. We can detect a cycle in $\hat{G}_q$ in $O(|B|^2)$ time, for example by applying a depth-first search on $\hat{G}_q$, and thus, we can check condition (iv) in $O(|B|^2)$ time.

For condition (ii), we compare the corresponding segments of the vertex sets of the connected components in $\hat{G}_q$ with the segments in $\mathcal{Y}_q^\alpha$. Finding the connected components in $\hat{G}_q$ can be done in $O(|B|^2)$ time, for example by applying a depth-first search in $\hat{G}_q$. The comparison of the segments can be done in $O(|B|^2)$ time. Thus, condition (ii) can be verified in $O(|B|^2)$ time. We conclude that for each $q \in [p]$, we can check conditions (ii)-(iv) in $O(|B|^3)$ time.

We can check conditions (i)-(iv) for each pair of signatures for node β and node γ in $O(p \cdot |B|^3)$ time. Therefore, the overall running time for filling all entries in T for a join node is in $O(p \cdot (\omega+2)^{3 \cdot p \cdot (\omega+1)+3})$.

We described how to fill the entries in the table T of the dynamic program according to each type of nodes in the tree decomposition $\mathbb{T}$. We proved the correctness of each rule of filling an entry and discussed the running time for the filling of an entry for each of the types of nodes. We use the dynamic program to prove Theorem 6.1.

Proof of Theorem 6.1. Let G be graph with $s, t \in V(G)$ given together with a tree decomposition $\mathbb{T}' = (T', (B'_\alpha)_{\alpha \in V(T')})$ of width $\omega' := \omega(\mathbb{T}')$ of G. We modify the tree decomposition $\mathbb{T}'$ in polynomial time to a nice tree-decomposition with introduce edge nodes of equal width, and add the vertices s and t to every bag. Let $\mathbb{T}$ be the nice tree decomposition with introduce edge nodes and vertices s and t contained in every bag obtained from $\mathbb{T}'$. Note that $\omega := \omega(\mathbb{T}) \leq \omega' + 2$. We apply the dynamic program described above bottom-up on the tree decomposition $\mathbb{T}$. The dynamic program runs in $O(p \cdot (\omega + 2)^{3 \cdot p \cdot (\omega+1)+4} \cdot |V(G)|)$ time. Since $\omega \leq \omega' + 2$, it follows that the dynamic program runs in $O(p \cdot (\omega' + 4)^{3 \cdot p \cdot (\omega'+3)+4} \cdot |V(G)|)$ time. Finally, we read out the minimum number of shared edges for p s-t paths in the entries of the root node in $\mathbb{T}$ as follows.

Let τ be the root node of $\mathbb{T}$. Note that $\{s, t\} \subseteq B_\tau$. Let $\mathcal{F}$ be the set of all signatures for node τ such that for all signatures $\mathcal{X}^\tau = (\mathcal{Y}_q^\tau, Z_q^\tau)_{q=1,\dots,p}$ in $\mathcal{F}$ it holds that for all $q \in [p]$ there exists a segment $M \in \mathcal{Y}_q^\tau$ with $\{s, t\} \subseteq M$. Due to our construction, a segment of a segmentation corresponds to a tree in a partial solution for the given graph. Hence, a set of p segmentations, where for each of the p segmentations there exists a segment that contains the vertices s and t, corresponds to a solution for Minimum Shared Edges with p paths. Thus, the minimum number of shared edges for p s-t paths equals $\min_{\mathcal{X}^\tau \in \mathcal{F}} T[\tau, \mathcal{X}^\tau]$. $\square$

We remark that we can modify the dynamic program in such a way that we can solve the weighted variant of Minimum Shared Edges, that is, with weights $w : E(G) \to \mathbb{N}$ on the edge set of the input graph. The cost of the partial solutions is the sum of the weights of shared edges, and thus the entry in the table of the dynamic program. For an introduce edge node, in the case of share-compability, we increase the value of the entry by the weight of the introduced edge. More precisely, for a introduce edge node α that introduces edge e and a signature $\mathcal{X}^\alpha$ for node α, the filling rule is adjusted by

$$T[\alpha, \mathcal{X}^\alpha] = \min \left(T[\beta, \mathcal{X}^\beta] + \begin{cases} w(e), & \text{if } \mathcal{X}^\beta \text{ and } \mathcal{X}^\alpha \text{ are share-compatible,} \\ 0, & \text{otherwise} \end{cases} \right),$$

where the minimum is taken over all signatures $\mathcal{X}^\beta$ for node β compatible with $\mathcal{X}^\alpha$.

In the remainder of this section, we study an implication of our dynamic program for Short Minimum Shared Edges (SMSE).

Theorem 6.2. *Let G be a graph and $s, t \in V(G)$ given together with a tree decomposition of G of width ω, and let $p, \lambda \in \mathbb{N}$ and $k \in \mathbb{N}_0$. Then, instance (G, s, t, p, k, λ) of* Short Minimum Shared Edges *can be solved in FPT-time with respect to the number p of paths, the upper bound λ on the length of the paths, and the width ω of the given tree decomposition.*

We show that we can modify the dynamic program given in this section according to SMSE. Let G be a graph and $s,t \in V(G)$ given together with a tree decomposition of G of width ω. We modify the tree decomposition to a nice tree decomposition with introduce edge nodes and vertices s and t contained in every bag as described above. If α is a node in the tree decomposition, then we consider the signature $\mathcal{X}^\alpha := (\mathcal{Y}_q^\alpha, Z_q^\alpha, \ell_q^\alpha)_{q=1,\ldots,p}$, where ℓ_q^α is an integer in $[\lambda] \cup \{0\}$. Intuitively, for each $q \in [p]$, the integer ℓ_q indicates the number of edges in the forest with index q. We add the following additional requirements to each type of node.

- If α is a leaf node, then set $\ell_q^\alpha = 0$ for all $q \in [p]$.
- If α is an introduce vertex node or a forget node with child node β, then add as additional requirement for compatibility that $\ell_q^\alpha = \ell_q^\beta$ for all $q \in [p]$.
- If α is an introduce edge node of edge e with child node β, then add as additional requirement for compatibility that $\ell_q^\alpha = \ell_q^\beta + 1$, if the segmentation q for node α is the result of merging two segments in the segmentation q for node β by edge e, that is, if $\mathcal{X}^\alpha$ and $\mathcal{X}^\beta$ are share-compatible, and that $\ell_q^\alpha = \ell_q^\beta$ otherwise.
- If α is a join node with two child nodes β and γ, then add as additional requirement for compatibility that $\ell_q^\alpha = \ell_q^\beta + \ell_q^\gamma$ for all $q \in [p]$. The number of edges in a forest that is the result of the union of two edge-disjoint forests is exactly the sum of the number of edges of the two forests in the union.

If for any signature of a node there is no compatible signature in the child node, or do not exist two compatible signatures in the two child nodes, then the corresponding entry in the table of the dynamic program is set to infinity. Note that the number of signatures for a node α of a tree decomposition of width ω is in $O((\lambda+1)^p \cdot (\omega+2)^{p\cdot(\omega+1)})$. Thus, the adapted dynamic program solves SMSE(p,λ,ω) in $O((\lambda+1)^{3p} \cdot p \cdot (\omega+4)^{3\cdot p\cdot(\omega+3)+4} \cdot |V(G)|)$ time.

Proof of Theorem 6.2. Let G be graph with $s,t \in V(G)$ given together with a tree decomposition $\mathbb{T}' = (T', (B'_\alpha)_{\alpha \in V(T')})$ of width $\omega' := \omega(\mathbb{T}')$ of G. We modify the tree decomposition $\mathbb{T}'$ as described in the proof of Theorem 6.1 to a nice tree decomposition $\mathbb{T}$ with introduce edge nodes and vertices s and t contained in every bag.

We apply the adapted dynamic program described above bottom-up on the tree decomposition $\mathbb{T}$. The adapted dynamic program runs in $O((\lambda+1)^{3p} \cdot p \cdot (\omega'+4)^{3\cdot p\cdot(\omega'+3)+4} \cdot |V(G)|)$ time. Finally, note that the entry for a signature in the table for the root node is different from infinity if and only if each of the forests corresponding to the segmentations contains at most λ edges. Thus, we read out the minimum number of shared edges for p s-t paths in the entries of the root node in $\mathbb{T}$ as described in the proof of Theorem 6.1. □

Finally, we remark that SMSE(p, λ) is fixed-parameter tractable on planar graphs.

Theorem 6.3. Short Minimum Shared Edges *is fixed-parameter tractable on planar graphs with respect to the number p of paths and the upper bound λ on the length of the paths.*

Proof. To prove Theorem 6.3, we show that we can reduce each instance (G, s, t, p, k, λ) of Short Minimum Shared Edges to an equivalent instance $(G', s, t, p, k, \lambda)$ of Short Minimum Shared Edges in such a way that $\operatorname{diam}(G') \leq 2 \cdot \lambda$, that is, that the diameter of G' is upper-bounded by $2 \cdot \lambda$. First, we apply for the vertices s and t a breadth-first search. If for a vertex $v \in V(G)$ it holds that $\operatorname{dist}_G(s, v) + \operatorname{dist}_G(v, t) > \lambda$ in G, then we can delete v in G, since no s-t path in G of length at most λ contains vertex v. Let G' be the graph obtained from G by deleting all vertices $v \in V(G)$ with $\operatorname{dist}_G(s, v) + \operatorname{dist}_G(v, t) > \lambda$ in G. It follows that (G, s, t, p, k, λ) is a yes-instance of SMSE if and only if $(G', s, t, p, k, \lambda)$ is a yes-instance of SMSE. Moreover, since for every vertex $v \in V(G')$ holds that $\operatorname{dist}_{G'}(s, v) \leq \lambda$, for every two vertices $w, u \in V(G')$ it holds that $\operatorname{dist}_{G'}(v, w) \leq 2 \cdot \lambda$. Hence, the diameter $\operatorname{diam}(G)$ of G is upper-bounded by $2 \cdot \lambda$.

Eppstein [7] showed that any planar graph that allows a rooted spanning tree of depth at most ℓ has a tree decomposition of width at most $3 \cdot \ell$ that can be found in $O(\ell \cdot n)$ time, where n denotes the number of vertices in the graph. By our construction, graph G' allows a rooted spanning tree of depth at most $2 \cdot \lambda$ since $\operatorname{diam}(G') \leq 2 \cdot \lambda$. Thus, we can compute a tree decomposition $\mathbb{T}$ of G' of width $\omega \leq 6 \cdot \lambda$ in $O(\lambda \cdot |V(G')|)$ time. Given G' and $\mathbb{T}$, by Theorem 6.2, we can solve instance $(G', s, t, p, k, \lambda)$ in FPT-time. Since instance $(G', s, t, p, k, \lambda)$ is equivalent to instance (G, s, t, p, k, λ), SMSE(p, λ) can be solved in FPT-time on planar graphs. $\square$

7 Fixed-Parameter Tractability with Respect to the Number of Paths

In the previous section, we presented an algorithm that solves an instance of MSE in $O(p \cdot (w+4)^{3\cdot p\cdot(\omega+3)+4} \cdot |V(G)|)$ time, where G is the graph, p is the number of paths and ω is an upper bound on the treewidth of graph G. As a consequence, MSE(p, ω) is fixed-parameter tractable, as we already know due to Ye et al. [19]. Moreover, due to Ye et al. [19], we know that MSE(ω) is in XP. By Theorem 3.10, MSE(p, k) is fixed-parameter tractable and MSE(k) is in XP. By Theorem 5.1, MSE(k) is $W[2]$-hard. This section completes the picture by studying the tractability of MSE(p). We show that Minimum Shared Edges is fixed-parameter tractable with respect to the number p of paths, which is the main result of this section:

Theorem 7.1. Minimum Shared Edges *is fixed-parameter tractable with respect to the number p of paths.*

More precisely, we show that MSE(p) can be solved in $O(p^2 \cdot (h(p)+4)^{3\cdot p\cdot(h(p)+3)+4} \cdot |G|)$ time, where G is the input graph, and h is a function only depending on p.

In the following, we prepare the proof of Theorem 7.1. Let $G = (V, E)$ be a graph with two vertices $s, t \in V$. Let $p \in \mathbb{N}$ and $k \in \mathbb{N}_0$ be two integers. We consider instance (G, s, t, p, k) of Minimum Shared Edges. We apply some modifications on graph G to finally obtain a graph G^*. We show that, on the one hand, the treewidth of graph G^* is upper-bounded by a function only depending on p, since one of our modifications is the treewidth reduction technique due to Marx et al. [14]. The treewidth of the graph obtained from the treewidth reduction technique is upper-bounded by a function only depending on p. Given the upper bound on the treewidth of graph G^*, we can compute in linear-time a tree decomposition of graph G^* [4]. By Theorem 6.1, we can solve instance (G^*, s, t, p, k) in FPT-time with respect to the number p of paths and the width of the given tree decomposition, where here the width of the tree decomposition is a function only depending on p. On the other hand, we show that, by our modifications, there is a one-to-one correspondence between all minimal s-t cuts in graph G^* of size at most $p-1$ and all minimal s-t cuts in G of size at most $p-1$. We show that if an instance is a yes-instance of MSE, then we can find a solution such that each shared edge participates in a minimal s-t cut of size at most $p-1$. Using this fact, we show that the instances (G, s, t, p, k) and (G^*, s, t, p, k) are equivalent. Altogether, we show that we can solve instance (G, s, t, p, k) by solving instance (G^*, s, t, p, k) in FPT-time with respect to the number p of paths.

Figure 7.1 serves as an overview of the following modifications and the graphs obtained by the sequence of modifications. Let (G, s, t, p, k) be an instance of MSE, where G is the input graph with $s, t \in V(G)$. First, we obtain a graph H by subdividing each edge in G. We denote by V_E the set of vertices obtained from the subdivi-

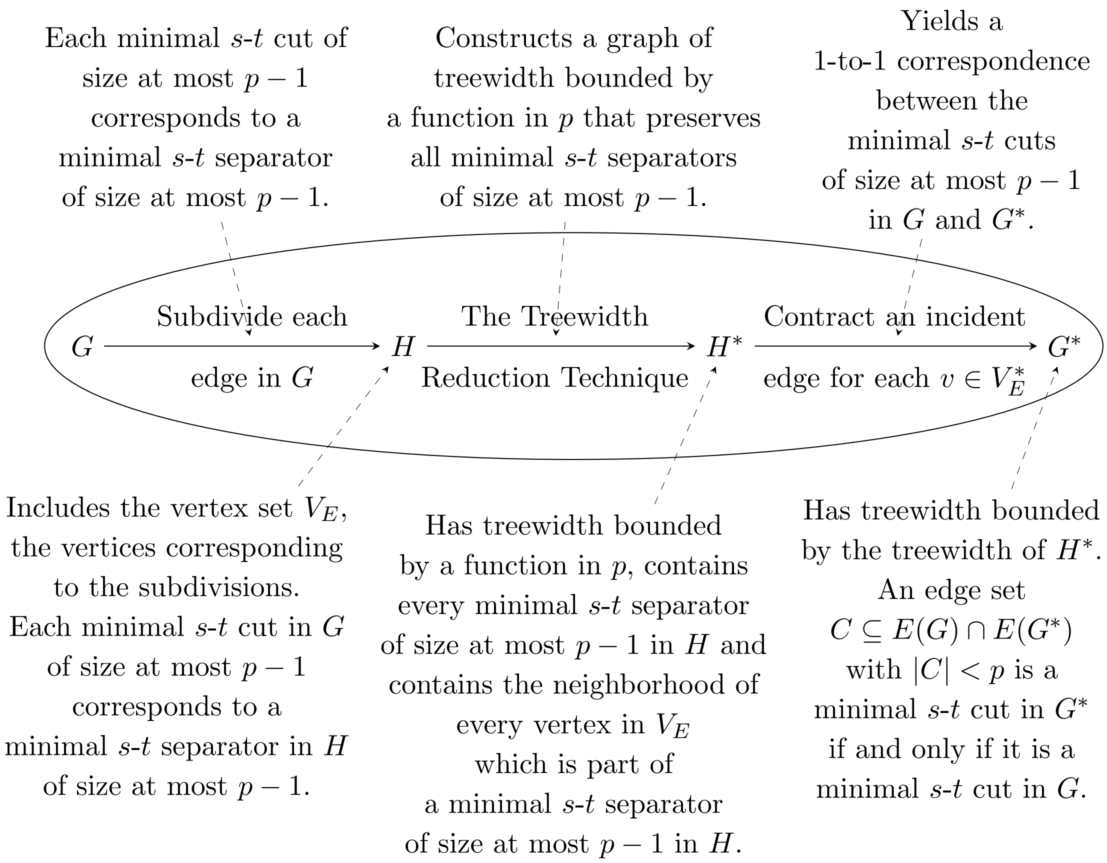

Figure 7.1: Overview of the strategy behind the proof of Theorem 7.1.

sions. As a consequence, every minimal s-t cut in G of size at most $p-1$ corresponds to a minimal s-t separator in H of size at most $p-1$. Next, we apply the treewidth reduction technique on H to obtain the graph H^*. By the treewidth reduction technique, graph H^* contains all minimal s-t separators in H of size at most $p-1$ and the treewidth of graph H^* is upper-bounded by a function only depending on p. We denote by $V_E^* := \{v \in V(H^*) \mid v \in V_E\}$ the set of vertices in V_E which are preserved by the treewidth reduction technique in H^*. Finally, we contract an incident edge for each vertex in $V_E^* \subseteq V(H^*)$ to obtain the graph G^*. We already discussed the properties of G^* above. We provide some short comments for each modification and for each graph in the sequence of modifications in Figure 7.1.

In the following, we modify step by step graph G to graph G^*. We discuss each step and we prove the properties of the obtained graphs described above. Finally, we give a proof of Theorem 7.1.

We start with the following lemma which states that if our instance is a yes-instance, then we can find a solution where each of the shared edges is part of a minimal s-t cut of size smaller than the number p of paths.

Lemma 7.2. *If (G, s, t, p, k) is a yes-instance of* MSE *and G has a minimal s-t cut of size smaller than p, then there exists a solution $F \subseteq E$ such that each $e \in F$ is in a minimal s-t cut of size smaller than p in G.*

Recall that if G does not have a minimal s-t cut of size smaller than p, then we can find p s-t paths without sharing an edge.

Proof. We make use of the contraction equivalent of MSE introduced in Section 3. We show that for every minimal solution for MSE it holds that each edge of the solution is part of a minimal s-t cut of size smaller than p, where a solution is minimal if it is not a superset of another solution.

Let $G = (V, E)$ be the graph. Let (G, s, t, p, k) be a yes-instance of MSE. Then there exists a solution $L \subseteq E$, $|L| \leq k$, such that graph $G_L := G/L$ with unit edge capacities allows a maximum s-t flow of value at least p. We call a solution L minimal if there is no edge $e \in L$ such that graph $G/(L\backslash\{e\})$ with unit edge capacities allows a maximum s-t flow of value at least p.

Let L be a minimal solution and let $e \in L$. Suppose that e is not part of a minimal s-t cut of size smaller than p in G. Let $L' := L\backslash\{e\}$ and $G_{L'} := G/L'$. We consider the following two cases.

Case 1: The maximum s-t flow of $G_{L'}$ has value smaller than p. Then, using the max-flow min-cut theorem, $G_{L'}$ has an s-t cut C of size smaller than p. Since $e \notin C$, contracting edge e in $G_{L'}$ does not affect cut C. Therefore, C is also an s-t cut of size smaller than p in G_L and, again by the max-flow min-cut theorem, this implies a maximum flow of value smaller than p in G_L. This is a contradiction to the fact that L is a solution.

Case 2: The maximum s-t flow of $G_{L'}$ has value at least p. Then L' is a solution, which contradicts the minimality of L.

Since $|L| \leq k$ and each edge in L is in a minimal s-t cut of size smaller than p in G, this completes the proof. $\square$

As mentioned before, as part of our approach we want to use the treewidth reduction technique due to Marx et al. [14]. Given a graph $G = (V, E)$ with $T = \{s, t\} \subseteq V(G)$ and an integer $\ell \in \mathbb{N}$, first the treewidth reduction technique computes the set C of vertices containing all vertices in G which are part of a minimal s-t separator of size at most ℓ in G. Then, it constructs the so-called *torso* of graph G given C and T, that is the induced subgraph $G[C \cup T]$ with additional edges between $v, w \in C \cup T$ with $\{v, w\} \notin E(G)$ if there is a v-w path in G whose internal vertices are not contained in $C \cup T$. Finally, each of these additional edges is subdivided and ℓ additional

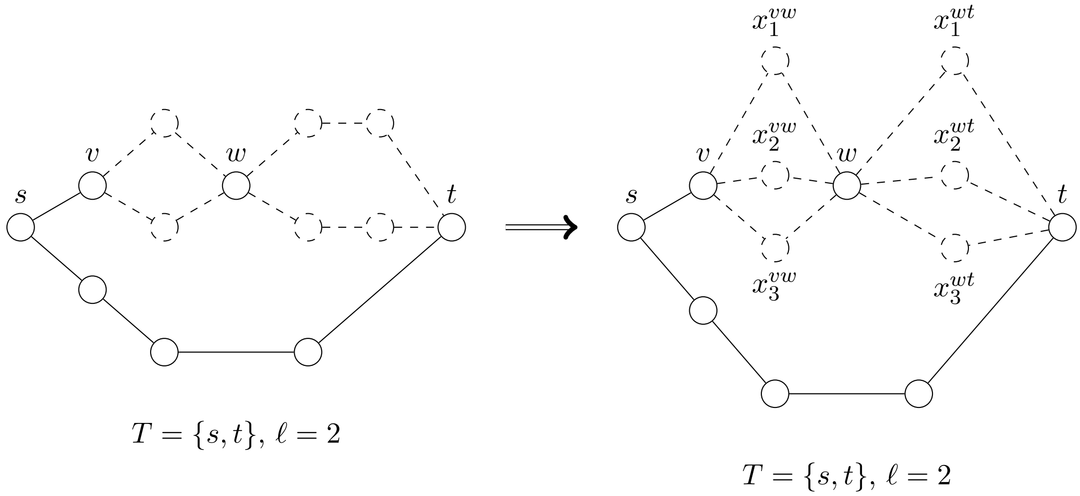

Figure 7.2: Example for the treewidth reduction technique.

copies of each of that subdivisions are introduced, that is, if $\{v, w\}$ is one of these additional edges, then the vertices $x_1^{vw}, \ldots, x_{\ell+1}^{vw}$ are added and edge $\{v, w\}$ is replaced by the edges $\{\{v, x_1^{vw}\}, \ldots, \{v, x_{\ell+1}^{vw}\}, \{x_1^{vw}, w\}, \ldots, \{x_{\ell+1}^{vw}, w\}\}$. In the following, we denote these paths by *copy paths*. The resulting graph contains all minimal s-t separators of size at most ℓ in G and has treewidth upper-bounded by $h(\ell)$ for some function h only depending on ℓ.

Theorem 7.3 (Marx et al. [14, Theorem 2.15])**.** *Let G be a graph, $T \subseteq V(G)$, and let ℓ be an integer. Let C be the set of all vertices of G participating in a minimal s-t separator of size at most ℓ for some $s, t \in T$. For every fixed ℓ and $|T|$, there is a linear-time algorithm that computes a graph G^* having the following properties:*

(1) $C \cup T \subseteq V(G^*)$

(2) For every $s, t \in T$, a set $L \subseteq V(G^*)$ with $|L| \leq \ell$ is a minimal s-t separator of G^* if and only if $L \subseteq C \cup T$ and L is a minimal s-t separator of G.

(3) The treewidth of G^* is at most $h(\ell, |T|)$ for some function h.

(4) $G^*[C \cup T]$ is isomorphic to $G[C \cup T]$.

Figure 7.2 shows an example for the application of the treewidth reduction technique. We use dashed edges and vertices to highlight the changes when applying the treewidth reduction technique with $T = \{s, t\}$ and parameter $\ell = 2$. On the left-hand side, the original graph is shown. On the right-hand side, the resulting graph after applying the treewidth reduction technique with $T = \{s, t\}$ and $\ell = 2$ on the left-hand side graph is shown.

Considering Lemma 7.2, we are interested in minimal s-t cuts of size smaller than p in G. The treewidth reduction technique guarantees to preserve minimal s-t separators of a specific size, but does not guarantee to preserve minimal s-t cuts of a specific size. Thus, we need to modify our graph G in such a way that each minimal s-t cut in G corresponds to a minimal s-t separator in the modified graph. We modify graph G in the following way.

Step 1. We subdivide each edge in $E(G)$, that means for each edge $e = \{v, w\}$ in $E(G)$ we add a vertex x_e and replace edge e by edge $\{v, x_e\}$ and edge $\{x_e, w\}$. We say that vertex x_e as well as edge $\{v, x_e\}$ and edge $\{x_e, w\}$ *correspond* to edge e, and vice versa. We denote by $V_E := \{x_e \mid e \in E\}$ and by E' the edge set replacing the edges in E. Let $H := (V \cup V_E, E')$ be the resulting graph.

In the following, we denote by H the graph obtained from G by applying Step 1. Note that each edge in H is incident with exactly one vertex in V_E and one vertex in V. Thus, no two vertices in V_E and no two vertices in V are neighbors. Moreover, note that each vertex in V_E has degree exactly two. It holds that $|V \cup V_E| = |V| + |E|$ and $|E'| = 2 \cdot |E|$, since each edge in E corresponds to exactly one vertex and two edges.

Lemma 7.4. *(G, s, t, p, k) is a yes-instance of* MSE *if and only if $(H, s, t, p, 2k)$ is a yes-instance of* MSE.

Proof. Intuitively, every edge in G corresponds to two edges in H and every two edges in H both incident with an vertex in V_E correspond to an edge in G.

"$\Rightarrow$": Consider a solution for the yes-instance (G, s, t, p, k) of MSE. For each edge $e = \{v, w\} \in E(G)$ that is shared in the solution, consider the corresponding two edges $\{v, x_e\}$ and $\{x_e, w\}$ in graph H. Sharing these at most $2k$ edges yields a solution for instance $(H, s, t, p, 2k)$ of MSE.

"$\Leftarrow$": Consider a minimal solution for the yes-instance $(H, s, t, p, 2k)$. Observe that in such a solution, a vertex in V_E is incident with either no or two shared edges. Each vertex in V_E that appears in at least two s-t paths is incident with two shared edges. Each vertex in V_E corresponds to one edge in G. Let $F \subseteq E(G)$ be the set of edges such that $e = \{v, w\} \in F$ if the edges $\{v, x_e\}$ and $\{x_e, w\}$ in $E(H)$ are shared in the solution for $(H, s, t, p, 2k)$. Note that $|F| \leq k$ since there are at most $2k$ shared edges. Thus, F is a solution for instance (G, s, t, p, k) of MSE. □

Recall that we are interested in s-t cuts in G. By our modification from Step 1 of G to H, for each edge in G there is a corresponding vertex in V_E in H. The following lemma gives a one-to-one correspondence between s-t cuts in G and those s-t separators in H that contain only vertices in V_E.

Lemma 7.5. *If C is an s-t cut in G, then $V_C := \{w \in V_E \mid w$ corresponds to $e \in C\}$ is an s-t separator in H. If $W \subseteq V_E$ is an s-t separator in H, then $C_W := \{e \in E \mid e$ corresponds to $w \in W\}$ is an s-t cut in G.*

Proof. Let C be an s-t cut in G. Suppose that the set $V_C := \{w \in V_E \mid w$ corresponds to $e \in C\}$ is not an s-t separator in H. Then there exists a path P' avoiding V_C in H connecting s and t. Since no two vertices in V_E are neighbors and no two vertices in V are neighbors, the vertices in path P' alternate in V and V_E. Since we know that the vertices in V_E correspond to edges in G, $P := P' \cap V$ describes a path in G connecting s and t avoiding all edges in C. This is a contradiction to the fact that C is an s-t cut in G, and hence set V_C is an s-t separator in H.

Let $W \subseteq V_E$ be an s-t separator in H. Suppose that the set $C_W := \{e \in E \mid e$ corresponds to $w \in W\}$ is not an s-t cut in G. Then there exists a path P avoiding C_W in G connecting s and t. Let $V_P \subseteq V(H)$ be the set of vertices in H such that each vertex in V_P either corresponds to an edge in P or is an endpoint of an edge in P. We remark that $W \cap V_P = \emptyset$. Moreover, set V_P is the set of vertices of an s-t path in H. This is a contradiction to the fact that W is an s-t separator in H, and hence set C_W is an s-t cut in G. □

In the following lemma, we show that Lemma 7.5 holds also for minimal s-t cuts and minimal s-t separators. This is important, since we will use a combination of the treewidth reduction technique and Lemma 7.2 later on.

Lemma 7.6. *Every minimal s-t cut in G corresponds to a minimal s-t separator in H.*

Proof. Let C be a minimal s-t cut in G. By Lemma 7.5, we know that $V_C := \{w \in V_E \mid w$ corresponds to $e \in C\}$ is an s-t separator in H. If V_C is a minimal s-t separator in H, then we are done. Thus, suppose that V_C is an s-t separator in H, but V_C is not a minimal s-t separator in H. Then there exists a vertex $w \in V_C$ such that $V_C \backslash \{w\}$ is an s-t separator in H. Let $e \in C$ be the edge in G corresponding to vertex w. Since $V_C \backslash \{w\} \subseteq V_E$, again by Lemma 7.5 we know that $C \backslash \{e\}$ is an s-t cut in G. This is a contradiction to the fact that C is a minimal s-t cut in G, and hence, V_C is a minimal s-t separator in H. □

We know that each minimal s-t cut in G corresponds to a minimal s-t separator in H. Next, we show that every vertex in the neighborhood of each minimal s-t separator containing only vertices in V_E belongs to a minimal s-t separator. Recall that for $W \subseteq V$ we denote by $N_G(W)$ the open neighborhood of the vertex set W in G and by $N_G[W] := W \cup N_G(W)$ the closed neighborhood of the vertex set W in G.

Lemma 7.7. *Let $W \subseteq V_E \subseteq V(H)$ be the set of vertices corresponding to a minimal s-t cut of size at most $\ell \in \mathbb{N}$ in G. Then, each vertex in $N_H[W]$ is part of a minimal s-t separator of size at most ℓ in H.*

Proof. Let $W \subseteq V_E \subseteq V(H)$ be given such that W corresponds to a minimal s-t cut in G of size at most ℓ. Note that by Lemma 7.6, W is a minimal s-t separator in H. Let x be an arbitrary vertex in $N_H(W)$. First, we show that $W' := (W \backslash N_H(x)) \cup \{x\}$ is an s-t separator in H.

Suppose that W' is not an s-t separator in H. Then there exists an s-t path P in $H - W'$. Note that each vertex in $W \cap N_H(x)$ is incident with vertex x and exactly one other vertex in $V(H)$. Thus, no vertex in $W \cap N_H(x)$ appears in path P. Hence, P is an s-t path in $H - W$. This is a contradiction to the fact that W is an s-t separator in H, and hence, W' is an s-t separator in H.

Next, we show that if W' is not a minimal s-t separator in H, then there exists a set $U \subseteq W' \backslash \{x\}$ such that $W' \backslash U$ is a minimal s-t separator in H. Let W' be an s-t separator in H, but not a minimal s-t separator in H. Suppose that for all $U \subseteq W' \backslash \{x\}$ it holds that $W' \backslash U$ is not a minimal s-t separator. Then there exists a set $X \subseteq W'$ with $x \in X$ such that $W' \backslash X$ is a minimal s-t separator in H. Since $W' \backslash X = W \backslash (N_H(x) \cap W) \backslash X \subseteq W$, this contradicts the fact that W is a minimal s-t separator in H. Hence, there exists a set $U \subseteq W' \backslash \{x\}$ such that $W' \backslash U$ is a minimal s-t separator in H.

Let $U \subseteq W' \backslash \{x\}$ be a set such that $W'' := W' \backslash U$ is a minimal s-t separator. Since $x \in W''$ and $|W''| \leq |W'| \leq |W|$, vertex x appears in a minimal s-t separator in H of size at most ℓ. Since vertex x was chosen arbitrarily in $N_H(W)$, each vertex in $N_H[W]$ is part of a minimal s-t separator of size at most ℓ in H. □

We obtained graph H from graph G by applying Step 1. By Lemma 7.6, we know that each minimal s-t cut in G corresponds to a minimal s-t separator in H. Moreover, by Lemma 7.7, if we consider a minimal s-t cut of size smaller than p in G, then, for each neighbor of the vertex set in H corresponding to the minimal s-t cut in G, there exists a minimal s-t separator of size smaller than p in H that contains that neighbor. As the next step (cf. Figure 7.1) we apply the treewidth reduction technique due to Marx et al. [14] on graph H.

Step 2. We apply the treewidth reduction technique [14] on graph H with $T = \{s, t\}$ and $p - 1$ as upper bound for the size of the minimal s-t separators, which results in graph H^*.

In the following, we denote by H^* the graph obtained from G by applying Steps 1 and 2. Let $V_E^* := \{v \in V(H^*) \mid v \in V_E\}$. Graph H^* contains all minimal s-t separators of size at most $p - 1$ in H. By Lemma 7.6, every minimal s-t cut of size at most $p - 1$ in G corresponds to a minimal s-t separator of size at most $p - 1$ in H and thus, by Step 2, to a minimal s-t separator of size at most $p - 1$ in H^*. By Lemma 7.7, the neighborhood of each vertex in H corresponding to a vertex in V_E^* is contained in the vertex set $V(H^*)$. As a consequence, we can reconstruct each edge in graph G that

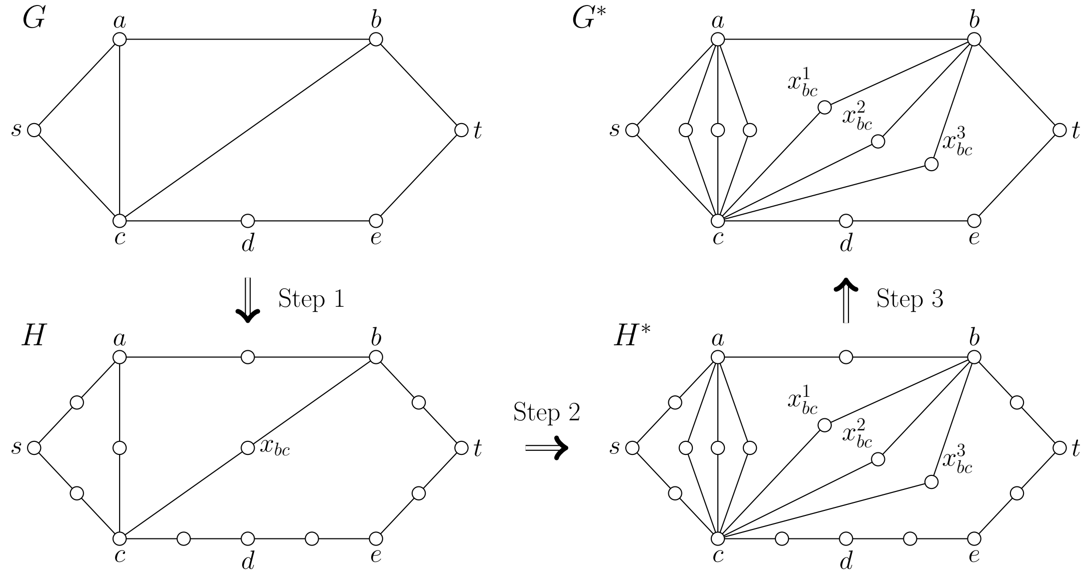

Figure 7.3: Example of Steps 1 to 3 on the example graph G (top-left) with $T = \{s, t\}$ and $p = 3$.

appears in a minimal s-t cut of size at most $p - 1$ in G as an edge in the graph H^*. As our next step (cf. Figure 7.1), we contract for each vertex in V_E^* an incident edge in graph H^*. We remark that if x^{vw} is a vertex in V_E^*, then the only edges incident with vertex x^{vw} are $\{v, x^{vw}\}$ and $\{x^{vw}, w\}$. In addition, the vertices v and w are the only neighbors of x^{vw} in graph H and in graph H^*.

Step 3. We contract for each vertex in V_E^* exactly one incident edge in H^* to obtain the graph G^*. In other words, we undo the subdivision we applied on G to obtain H.

In the following, we denote by G^* the graph obtained from G by applying Steps 1 to 3. We remark that $\text{tw}(G^*) \leq \text{tw}(H^*)$, since edge contraction does not increase the treewidth of a graph [17].

In Figure 7.3, we illustrate Steps 1 to 3 on an example graph G with $T = \{s, t\}$ and $p = 3$. The top-left graph is the original graph G. The bottom-left graph is graph H, obtained from G by applying Step 1. The bottom-right graph is graph H^*, obtained from H by applying Step 2. The top-right graph is the final graph G^*, obtained from H^* by applying Step 3.

Let $e = \{v, w\} \in E(G)$ be an edge in G and $x_e \in V_E \subseteq V(H)$ the corresponding vertex in H. Then $\{v, x_e\}$ and $\{x_e, w\}$ are the incident edges of x_e in H. If $x_e \in V(H^*)$, then one of the incident edges $\{v, x_e\}$ and $\{x_e, w\}$ with vertex x_e is contracted and yields edge $\{v, w\} \in E(G^*)$. We say that the edges $\{v, w\} \in E(G)$ and $\{v, w\} \in E(G^*)$ correspond one-to-one, and, for example, we write $\{v, w\} \in E(G) \cap E(G^*)$.

Considering the graphs G and G^*, we show that, given an s-t path in the one graph,

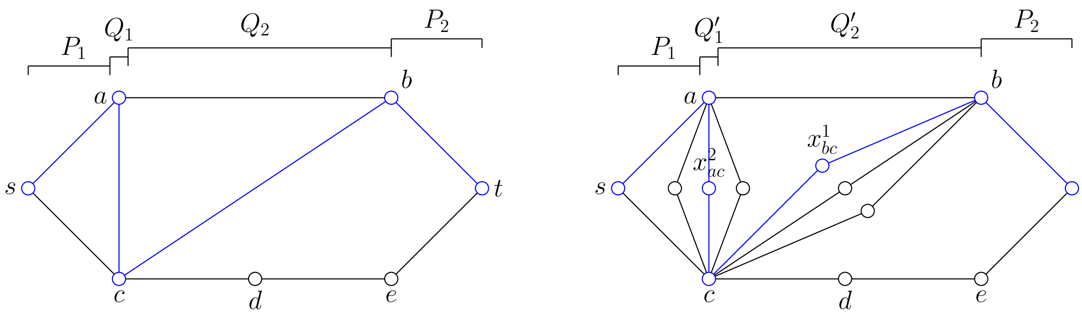

Figure 7.4: The graphs G (left-hand side) and G^* (right-hand side) from Figure 7.3. The blue colored edges and vertices belong to an s-t path in G and G^* respectively. The upper braces show the range of the consecutive subpaths P_1, Q_1,Q_2, P_2 for G and P_1, Q_1', Q_2', P_2 for G^*.

we can construct an s-t path in the other graph using a common set of edges in $E(G) \cap E(G^*)$.

Lemma 7.8. *(i) If P is an s-t path in G, then there exists an s-t path P^* in G^* that contains all edges in $E(P) \cap E(G^*)$.*

(ii) If P^* is an s-t path in G^*, then there exists an s-t path P in G that contains all edges in $E(P^*) \cap E(G)$.

Proof. (i): Let P be an s-t path in G. If P just contains edges in $E(G) \cap E(G^*)$, then we set $P^* = P$. If P contains edges in $E(G) \backslash E(G^*)$, then P has a representation of consecutive subpaths P_i, $1 \le i \le j$, and Q_i, $1 \le i \le \ell$, where $\{P_i\}_{1 \le i \le j}$ is the set of subpaths of P that just contain edges in $E(G) \cap E(G^*)$ and $\{Q_i\}_{1 \le i \le \ell}$ is the set of subpaths of P with endpoints in $V(G) \cap V(G^*)$, inner vertices in $V(G) \backslash V(G^*)$ and edges in $E(G) \backslash E(G^*)$. Since for each $1 \le i \le \ell$, path Q_i is connecting two vertices $v, w \in V(G) \cap V(G^*)$ in G, there are p edge-disjoint paths of length 2 in G^* connecting v and w using the edges in $E(G^*) \backslash E(G)$, that are the copy paths. For each $i \in [\ell]$, let Q_i' be one of the copy paths connecting the endpoints of Q_i. Figure 7.4 illustrates this correspondence on an example graph. Replacing each Q_i by such a path Q_i' in G^* yields a path P' with consecutive subpaths P_i, $1 \le i \le j$, and Q_i', $1 \le i \le \ell$, in G^* connecting s and t that contains all edges in $E(P) \cap E(G^*)$.

(ii): Let P^* be an s-t path in G^*. If P^* just contains edges in $E(G) \cap E(G^*)$, then we set $P = P^*$. If P^* contains edges in $E(G^*) \backslash E(G)$, then P^* has a representation of consecutive subpaths P_i', $1 \le i \le j$, and Q_i', $1 \le i \le \ell$, where $\{P_i'\}_{1 \le i \le j}$ is the set of subpaths of P^* that just contain edges in $E(G^*) \cap E(G)$ and $\{Q_i'\}_{1 \le i \le \ell}$ is the set of subpaths of P^* with endpoints in $V(G^*) \cap V(G)$, inner vertices in $V(G^*) \backslash V(G)$, and edges in $E(G^*) \backslash E(G)$. We remark that each Q_i' is one of the copy paths in G^*. By construction of G^*, each Q_i' connects two vertices in $V(G^*) \cap V(G)$ that are connected by a path in G with no inner vertices in $V(G^*) \cap V(G)$. Therefore, for each $i \in [\ell]$, we can replace path Q'_i by such a path Q_i in G. This yields an s-t path P in G with consecutive subpaths P'_i, $1 \leq i \leq j$, and Q_i, $1 \leq i \leq \ell$, that contains all edges in $E(P^*) \cap E(G)$. $\square$

We modified graph G to graph G^* by applying Steps 1 to 3. By Lemma 7.8, we can construct s-t paths in G and G^* that use edges in the common set of edges $E(G) \cap E(G^*)$. The next lemma states that each minimal s-t cut of size smaller than p in one of the graphs G and G^* is also a minimal s-t cut of size smaller than p in the other graph.

Lemma 7.9. *Let $C \subseteq E(G) \cap E(G^*)$. Edge set C is a minimal s-t cut in G of size smaller than p if and only if C is a minimal s-t cut in G^* of size smaller than p.*

Proof. We make use of Lemma 7.8 in the following proof. We remark that no edge in $E(G^*) \backslash E(G)$ is in any minimal s-t cut of size smaller than p in G^* since, by the treewidth reduction technique, for each of these edges there are $p-1$ copies in G^*.

"$\Rightarrow$": Let C be a minimal s-t cut of size smaller than p in G. By Lemma 7.6, C has a corresponding minimal s-t separator S_C of size smaller than p in H. By the treewidth reduction technique, S_C is a minimal s-t separator in H^*. By Lemma 7.7, every neighbor of S_C is contained in H^*. By our contraction of edges of H^* to G^*, for each vertex of S_C an incident edge is contracted and yields the edge set C again. Since S_C is a minimal s-t separator in H^* of size smaller than p and each vertex in S_C has degree exactly two, set C is a minimal s-t cut in G^* of size smaller than p.

"$\Leftarrow$": Let C be a minimal s-t cut in G^* of size smaller than p. Suppose C is not a minimal s-t cut in G of size smaller than p. We distinguish two cases.

Case 1: C is not an s-t cut in G. Then there exists a path P in G connecting s and t avoiding the edges in C. By Lemma 7.8, there exists an s-t path P^* in G^* that contains all edges in $E(P) \cap E(G^*)$. Since no edge in $E(G^*) \backslash E(G)$ is in any minimal s-t cut of size at most $p-1$ of G^*, P^* avoids the edges in C. This is a contradiction to the fact that C is a minimal s-t cut in G^*.

Case 2: C is an s-t cut in G, but C is not a minimal s-t cut in G. Then there exists $e \in C$ such that $C' := C \backslash \{e\}$ is an s-t cut in G. Since C is a minimal s-t cut in G^*, the set C' is not an s-t cut in G^*. Thus, there exists an s-t path P^* in G^* that avoids the edges in C'. By Lemma 7.8, there exists an s-t path P in G that contains all the edges in $E(P^*) \cap E(G)$. Since no edge in $E(G) \backslash E(G^*)$ is in any minimal s-t cut of size at most $p-1$ in G, path P avoids the edges in C'. Therefore, set C' is not an s-t cut in G, and thus, C is a minimal s-t cut in G. $\square$

Recalling Lemma 7.2, we know that if an instance of MSE is a yes-instance, then we can find k edges such that the k edges form a solution for the instance and each of the k edges is part of a minimal s-t cut of size smaller than p in G. By Lemma 7.9, the graphs G and G^* have the same set of minimal s-t cuts of size smaller than p in common. Combining Lemma 7.2 and Lemma 7.9 leads to the following lemma.

Lemma 7.10. *(G^*, s, t, p, k) is a yes-instance of* MSE *if and only if (G, s, t, p, k) is a yes-instance of* MSE.

Proof. We make use of the contraction equivalent of MSE.

"$\Rightarrow$": Let (G^*, s, t, p, k) be a yes-instance of MSE. By Lemma 7.2, we find a solution $F \subseteq E(G^*)$ such that each edge in F is part of a minimal s-t cut in G^* of size smaller than p. It follows that $F \subseteq E(G) \cap E(G^*)$, since by our construction no edge in $(E(G^*)\backslash E(G))$ is part of a minimal s-t cut of size smaller than p in G^*. Let $G_F := G/F$ be the graph G with all edges in F contracted. Suppose that G_F with unit edge capacities allows a maximum s-t flow of value smaller than p. Then there exists a minimal s-t cut C of size smaller than p in G_F. By Lemma 7.9, C is also a minimal s-t cut of size smaller than p in $G^*_F := G^*/F$. This is a contradiction to the fact that the value of any maximum s-t flow in G^*_F with unit edge capacities is at least p, and hence, set F is a solution for instance (G, s, t, p, k).

"$\Leftarrow$": Let (G, s, t, p, k) be a yes-instance of MSE. By Lemma 7.2, we find a solution $F \subseteq E(G)$ such that each edge in F is part of a minimal s-t cut in G of size smaller than p. It follows that $F \subseteq E(G) \cap E(G^*)$. Suppose that $G^*_F := G^*/F$ with unit edge capacities allows a maximum s-t flow of value smaller than p. Then there exists a minimal s-t cut C of size smaller than p in G^*_F. By Lemma 7.9, C is a minimal s-t cut of size smaller than p in $G_F := G/F$. This is a contradiction to the fact that the value of any maximum s-t flow in G_F with unit edge capacities is at least p, and hence, set F is a solution for instance (G^*, s, t, p, k). ☐

By Lemma 7.10, we know that the instances (G^*, s, t, p, k) and (G, s, t, p, k) are equivalent for MSE. By our construction, we know that the treewidth of G^* is upper-bounded by a function only depending on the number p of paths. In addition, we know that Minimum Shared Edges is fixed-parameter tractable with respect to the number p of paths and an upper bound on the treewidth of the input graph. Thus, we are ready to prove our main result.

Proof of Theorem 7.1. First we modify our graph $G = (V, E)$ by applying Steps 1 to 3. Let H, H^*, and G^* be the according graphs. By Theorem 7.3, the treewidth of H^* is upper-bounded by $h(p)$ for some function h. Since edge contractions do not increase the treewidth of a graph [17], it follows that $\mathrm{tw}(G^*) \leq \mathrm{tw}(H^*)$. By Lemma 7.10, the instances (G^*, s, t, p, k) and (G, s, t, p, k) are equivalent for MSE.

We know that MSE(p, ω) is fixed-parameter tractable when parameterized by the number p of paths and by an upper bound ω on the treewidth of the input graph. Since function h only depends on p and $h(p)$ is upper-bounding the treewidth of graph G^*, we can decide instance (G^*, s, t, p, k) in $f(p) \cdot O(|V(G^*)|)$ time, where f is a computable function only depending on parameter p. Since $|V(G^*)| \leq |V(G)| + p \cdot |E(G)| \leq p \cdot |G|$, and the instances (G^*, s, t, p, k) and (G, s, t, p, k) are equivalent for MSE, we can decide instance (G, s, t, p, k) in $f(p) \cdot p \cdot O(|G|)$ time, that is, in FPT-time. $\square$

Finally, note that by Theorem 6.1, any instance (G, s, t, p, k) of MSE(p) can be solved in $O(p^2 \cdot (h(p) + 4)^{3 \cdot p \cdot (h(p)+3)+4} \cdot |G|)$ time with function h as described above.

8 Conclusion

We studied the computational complexity of MINIMUM SHARED EDGES. We showed that MINIMUM SHARED EDGES is NP-complete, even on graphs with maximum degree at least five. Moreover, we showed that MSE(k) is $W[2]$-hard. We used the treewidth reduction technique due to Marx et al. [14] to show that MSE(p) is fixed-parameter tractable, demonstrating the utility of the technique.

Discussion. We showed that MINIMUM SHARED EDGES can be solved in constant time on the unbounded, undirected $\mathbb{Z} \times \mathbb{Z}$-grid graph. According to our introductory example in Berlin, this result is of potential interest for applications. Street networks like in Manhattan have a high similarity to grid graphs.

We presented an algorithm that solves MSE(p, k) in $(p-1)^k \cdot O(|G|^2)$ time. This algorithm performs well for small values of parameter k. Therefore, we think that this algorithm could be also of practical interest, since in many applications the goal is to keep the value of k as small as possible.

In our approach, solving MSE(p) in FPT-time depends on applying a dynamic program on a tree decomposition. The running times of the dynamic programs are so far of theoretical interest only. The question is whether better running times of the dynamic programs are possible. More generally speaking, since we showed that MSE(p) is fixed-parameter tractable, we see this result as the potential starting point for the race in finding the smallest function only depending on the parameter p [13].

Our dynamic program provided in Section 6 could maybe improved as follows. If we define partial solutions as a set of simple paths, then the segments in a segmentation correspond to endpoints of paths, inner vertices of the paths and vertices not appearing in any path in the partial solution. This could decrease the total number of signatures, and implicitly the running time of the dynamic program.

We introduced SHORT MINIMUM SHARED EDGES in this work. We showed that SMSE(k, λ) is $W[2]$-hard. Further, we showed that SMSE(p, λ, ω) is fixed-parameter tractable, where ω is an upper bound on the treewidth of the input graph. We showed that SMSE(p, λ) is fixed-parameter tractable on planar graphs, which is of potentially practical interest, according to a practical application as described in our introductory example at the beginning of this work and the fact that street networks can often be (approximately) represented as planar graphs [3].

Challenges for future research. We showed that MINIMUM SHARED EDGES is NP-hard on graphs with maximum degree at least five. It remains an open question whether MINIMUM SHARED EDGES remains NP-hard on graphs with maximum degree three and four. For the latter case, our reduction from VERTEX COVER to MINIMUM SHARED EDGES we gave in Section 5 can may be adapted. The critical spot in our reduction is the degree of the vertices in the constructed graph that correspond to the vertices in the graph in the Vertex Cover instance. By our reduction, the degree of these vertices is upper-bounded by the maximum degree of the graph in the Vertex Cover instance plus two. Perhaps this can be improved to plus one, which would imply that Minimum Shared Edges remains NP-hard on graphs with maximum degree four.

It remains an open question whether Minimum Shared Edges on planar graphs is NP-hard or can be solved in polynomial time. In our opinion, this research question is of special interest since in applications as described in this work, planar graphs are likely to be considered, like street networks as mentioned above.

As one of our main results, we showed on the one hand that MSE(p) is fixed-parameter tractable. As a consequence, MSE(p) admits a problem kernel and accordingly, it remains an open question whether it admits a polynomial problem kernel. On the other hand, we showed that MSE(k) is $W[2]$-hard, and additionally, we showed that MSE(k) is in XP. It remains open whether MSE(k) is $W[2]$-complete. Due to Ye et al. [19], we know that MSE(tw) is in XP, that is, when parameterized by the treewidth of the input graph. It remains open whether MSE(tw) can be proven to be $W[i]$-hard for an $i \geq 1$. Finally according to Minimum Shared Edges, it remains an open question whether our FPT algorithm with respect to the number p of paths and the number k of shared edges can be further improved, or if a running time proportional to $p^k \cdot n^{O(1)}$ is best possible, where n denotes the number of vertices in the input graph.

According to Short Minimum Shared Edges, it remains open whether our result that SMSE(p, λ) is fixed-parameter tractable on planar graphs can be transferred to general graphs. The tractability with respect to k only and λ only remains open. The tractability of SMSE(p) remains open. According to SMSE(p) and SMSE(p, λ), we believe that there could be a way to adapt our approach using the treewidth reduction technique. The critical spots in that approach facing SMSE are the subgraphs induced by the copy paths, since they may allow shorter paths that are not feasible in the original graph.

As a final remark, we would like to briefly introduce the following variant of Minimum Shared Edges. Let G be a simple, undirected graph and $s, t \in V(G)$ be two vertices in G. For every s-t path P in G, let the edges in P be labeled with the distance to vertex s. The question is whether there are p s-t paths in G that *time-share* at most k edges, where an edge is called time-shared if the edge appears with the same label in at least two paths. We call this problem Time Minimum Shared Edges since the labeling of the edges as their distance to vertex s in the paths can be interpreted as time values. We consider this problem as interesting since in the context of our introductory example, the time aspect is not considered. If two or more convoys pass the same street but at two different points in time, then any possible attacker would not have any advantage of the fact that the street is shared. Challenges for future research could address the computational complexity of Time Minimum Shared Edges when parameterized by the number p of paths or when parameterized by the number k of time-shared edges. In addition, a challenge could be to provide an FPT algorithm with respect to the number p of paths and the number k of time-shared edges. Perhaps some of our approaches presented in this work can be adapted to some extend according to Time Minimum Shared Edges.

Bibliography

[1] Yusuke Aoki, Bjarni V. Halldórsson, Magnús M. Halldórsson, Takehiro Ito, Christian Konrad, and Xiao Zhou. The minimum vulnerability problem on graphs. In Zhao Zhang, Lidong Wu, Wen Xu, and Ding-Zhu Du, editors, *Combinatorial Optimization and Applications - 8th International Conference, COCOA 2014, Wailea, Maui, HI, USA, December 19-21, 2014, Proceedings*, volume 8881 of *Lecture Notes in Computer Science*, pages 299–313. Springer, 2014. 8, 10, 40

[2] Sepehr Assadi, Ehsan Emamjomeh-Zadeh, Ashkan Norouzi-Fard, Sadra Yazdanbod, and Hamid Zarrabi-Zadeh. The minimum vulnerability problem. *Algorithmica*, 70 (4):718–731, 2014. 8

[3] Marc Barthelemy. Spatial networks. In *Encyclopedia of Social Network Analysis and Mining*, pages 1967–1976. 2014. 70

[4] Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996. 14, 58

[5] Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 150–159. IEEE Computer Society, 2011. 14, 40

[6] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity.* Texts in Computer Science. Springer, 2013. 15, 30

[7] David Eppstein. Subgraph isomorphism in planar graphs and related problems. *Journal of Graph Algorithms and Applications*, 3(3), 1999. 57

[8] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory.* Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag Berlin Heidelberg, 2006. 15

[9] M. R. Garey, David S. Johnson, and Larry J. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976. 33

[10] Andrew V. Goldberg and Satish Rao. Flows in undirected unit capacity networks. *SIAM Journal on Discrete Mathematics*, 12(1):1–5, 1999. 13

[11] Jon M. Kleinberg and Éva Tardos. *Algorithm Design.* Addison-Wesley, 2006. 7, 13, 23

[12] Ton Kloks. *Treewidth, Computations and Approximations*, volume 842 of *Lecture Notes in Computer Science*. Springer, 1994. 14

[13] Christian Komusiewicz and Rolf Niedermeier. New races in parameterized algorithmics. In Branislav Rovan, Vladimiro Sassone, and Peter Widmayer, editors, *Mathematical Foundations of Computer Science 2012 - 37th International Symposium, MFCS 2012, Bratislava, Slovakia, August 27-31, 2012. Proceedings*, volume 7464 of *Lecture Notes in Computer Science*, pages 19–30. Springer, 2012. 70

[14] Dániel Marx, Barry O'Sullivan, and Igor Razgon. Finding small separators in linear time via treewidth reduction. *ACM Transactions on Algorithms*, 9(4):30, 2013. 10, 58, 60, 61, 64, 70

[15] Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford University Press, 2006. 15

[16] Masoud T. Omran, Jörg-Rüdiger Sack, and Hamid Zarrabi-Zadeh. Finding paths with minimum shared edges. *Journal of Combinatorial Optimization*, 26(4):709–722, 2013. 7, 8, 29, 30

[17] Neil Robertson and Paul D. Seymour. Graph minors. II. algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986. 65, 68

[18] Thomas Wolle and Hans L. Bodlaender. A note on edge contraction. Technical Report UU-CS-2004, Institute of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands, 2004. 12

[19] Z.Q. Ye, Y.M. Li, H.Q. Lu, and X. Zhou. Finding paths with minimum shared edges in graphs with bounded treewidths. In *Proc. Frontiers of Computer Science (FCS) 2013*, pages 40–46, 2013. 8, 9, 10, 40, 58, 71