Minimum Activation Cost Edge-Disjoint Paths in Graphs with Bounded Tree-Width

Hasna Mohsen Alqahtani(✉) and Thomas Erlebach

Department of Computer Science, University of Leicester, Leicester, UK
{hmha1,terlebach}@leicester.ac.uk

Abstract. In *activation network* problems we are given a directed or undirected graph $G = (V, E)$ with a family $\{f_{uv} : (u, v) \in E\}$ of monotone non-decreasing activation functions from D^2 to $\{0, 1\}$, where D is a constant-size subset of the non-negative real numbers, and the goal is to find activation values $x_v \in D$ for all $v \in V$ of minimum total cost $\sum_{v \in V} x_v$ such that the activated set of edges satisfies some connectivity requirements. We propose an algorithm that optimally solves the *minimum activation cost of k edge-disjoint st-paths* (*st*-MAEDP) problem in $O(|V||D|^{tw+1} tw^3 (k+1)^{(tw+3)^{2(tw+3)}} (tw+3)^{2(tw+3)+3})$ time for graphs with treewidth bounded by a constant tw.

1 Introduction

The *activation network* setting can be defined as follows. We are given a directed or undirected graph $G = (V, E)$ together with a family $\{f_{uv} : (u, v) \in E\}$ of monotone non-decreasing activation functions from D^2 to $\{0, 1\}$, where D is a constant-size subset of the non-negative real numbers, such that the activation of an edge depends on the chosen values from the domain D at its endpoints. An edge $(u, v) \in E$ is *activated* for chosen values x_u and x_v if $f_{uv}(x_u, x_v) = 1$, and the activation function f_{uv} is called *monotone non-decreasing* if $f_{uv}(x_u, x_v) = 1$ implies $f_{uv}(y_u, y_v) = 1$ for any $y_u \geq x_u$, $y_v \geq x_v$. The objective of activation network problems is to find activation values $x_v \in D$ for all $v \in V$ such that the total activation cost $\sum_{v \in V} x_v$ is minimized and the activated set of edges satisfies some connectivity requirements. Activation problems generalize several problems studied in the network literature such as power optimization, minimum broadcast tree and installation cost optimization. Several activation problems have been studied for arbitrary graphs in the recent research literature. Unfortunately, many of these problems are computationally hard. Obtaining a polynomial-time approximation algorithm is a typical approach to deal with an NP-hard problem. Another important approach is studying the problem on graph classes with nice decomposition properties such as bounded treewidth graphs to determine efficient algorithms. Panigrahi [8] shows that the fundamental problem of finding *minimum activation k edge-disjoint st-paths* (*st*-MAEDP) is NP-hard. It is an interesting area of research to investigate this problem for graphs with treewidth bounded by a constant tw. In this paper, we focus on developing a

Z. Lipták and W.F. Smyth (Eds.): IWOCA 2015, LNCS 9538, pp. 13–24, 2016.
DOI: 10.1007/978-3-319-29516-9_2

polynomial-time algorithm that optimally solves the *st*-MAEDP for graphs of bounded treewidth. (Note that k is not a constant but part of the input.)

Related Work. Activation network problems were first introduced by Panigrahi in [8] and various of these problems have been investigated in [1,2,6–8]. The *minimum activation st-path* (MAP) problem is optimally solvable in polynomial-time [8]. However, the *minimum spanning activation tree* (MSpAT) problem is NP-hard to approximate within a factor of $o(\log n)$ and there exists a $O(\log n)$-approximation algorithm for this problem [8]. Activation network problems generalize several problems studied in the network literature such as power optimization problems, which are modelled by a graph $G = (V, E)$ where each edge (u, v) in G is assigned a threshold power requirement θ_{uv}. For undirected graphs, an edge (u, v) is activated for chosen values x_u and x_v if each of these values is at least θ_{uv}, and it is activated if $x_u \geq \theta_{uv}$ for the directed case. The *minimum power st-path* (MPP) problem can be solved in polynomial time for both directed and undirected graphs [5]. For directed graphs, the *minimum power k node-disjoint st-paths* problem is also optimally solvable in polynomial time [4,11]. However, the *minimum power k edge-disjoint st-paths* problem is unlikely to admit even a polylogarithmic approximation algorithm for both the directed and undirected variants [4].

There is a $2k$-approximation algorithm for the *st*-MAEDP problem and a 2-approximation algorithm for the *minimum activation node-disjoint st-paths* (*st*-MANDP) problem [6]. In [1], we have studied the *st*-MAEDP and *st*-MANDP problems when $k = 2$, denoted by *st*-MA2EDP and *st*-MA2NDP respectively. We proved that a ρ-approximation algorithm for the *st*-MA2NDP problem implies a ρ-approximation algorithm for the *st*-MA2EDP problem. In the same paper, we obtained a 1.5-approximation algorithm for the *st*-MA2NDP problem and hence for the *st*-MA2EDP problem. The problems *st*-MAEDP and *st*-MANDP for the restricted version of activation networks with $|D| = 2$ and a single activation function for all edges have also been studied in [1]. Under this restriction, the *st*-MANDP problem is optimally solvable in polynomial time for arbitrary k (except for one case of the activation function, in which we require $k = 2$). The *st*-MAEDP problem, however, remains NP-hard [1,8]. So far these problems for an arbitrary constant-size D and fixed $k \geq 2$ are not known to be NP-hard. Recently, in [2], we considered the *st*-MANDP and the problem of finding *minimum activation cost node-disjoint paths* (MANDP) between k disjoint terminal pairs, $(s_1, t_1), \ldots, (s_k, t_k)$, for graphs of bounded treewidth. We proposed algorithms that optimally solve the *st*-MANDP problem in polynomial-time and the MANDP problem in linear-time for graphs with bounded treewidth [2]. There exists a polynomial-time algorithm that optimally solves the *st*-MA2EDP problem for graphs of bounded treewidth [1,2]. Other relevant work, applications and motivations of activation network problems have been addressed in [1,2,6–8].

Our Results. We develop a polynomial-time algorithm for the *st*-MAEDP problem for graphs with treewidth bounded by tw. Our algorithm efficiently combines an edge-coloring scheme with dynamic programming over a *nice tree-decomposition* (see Sect. 2). The edge-coloring scheme was introduced in [12] to

develop a polynomial-time algorithm that solves the *minimum shared-edge kst-paths* (MSEP) problem (finding k paths between s and t with minimum number of shared edges among the paths) for graphs of bounded treewidth. The MSEP is a generalization of the problem of finding k edge disjoint st-paths.

The rest of the paper is organized as follows. In Sect. 2 we recall some definitions and results of the class of graphs with bounded treewidth. Then in Sect. 3, we present a polynomial-time algorithm that solves the st-MAEDP problem optimally for graphs with treewidth bounded by tw. We conclude the paper by stating some open problems in Sect. 4.

2 Preliminaries

In this paper we consider the class of graphs of bounded treewidth. A graph $G = (V, E)$ has treewidth tw if it has a *tree-decomposition* of width tw [9]. The *tree-decomposition* concept is defined as follows.

Definition 1. *Given a graph $G = (V, E)$, a tree $\mathcal{T} = (I, F)$ and a family $\mathcal{X} = \{X_i\}_{i \in I}$ of subsets of V (called bags). The pair $(\mathcal{X}, \mathcal{T})$ is called a tree-decomposition of G if it satisfies the following conditions:*

- $V = \bigcup_{i \in I} X_i$.
- *For every edge $(v, w) \in E$, there exists an $i \in I$ with $v \in X_i$ and $w \in X_i$.*
- *For every vertex $v \in V$, the nodes $i \in I$ with $v \in X_i$ form a subtree of $\mathcal{T}$.*

The width of $(\mathcal{X}, \mathcal{T})$ is the number $\max_{i \in I} |X_i| - 1$. The treewidth tw of the graph G is the minimum width among all possible tree-decompositions of the graph.

Theorem 1 ([3]). *For any fixed tw, there exists a linear-time algorithm that checks whether a given graph $G = (V, E)$ has treewidth at most tw, and if so, outputs a tree-decomposition $(\mathcal{X}, \mathcal{T})$ of G with width at most tw.*

Definition 2. *A tree-decomposition $(\mathcal{X}, \mathcal{T})$ is called a* nice *tree-decomposition, if $\mathcal{T}$ is a binary tree rooted at some $r \in I$ that satisfies the following:*

- *Each node is either a leaf, or has exactly one or two children.*
- *Let $i \in I$ be a leaf. Then $X_i = \{u, v\}$ for some $(u, v) \in E$.*
- *For every edge $(u, v) \in E$, there is exactly one leaf $i \in I$ such that $u, v \in X_i$. (We say that the edge (u, v) is* associated *with that leaf $i \in I$).*
- *Let $j \in I$ be the only child of $i \in I$, then either $X_i = X_j \cup \{v\}$ or $X_i = X_j \setminus \{v\}$. The node i is called an introduce node or forget node, respectively.*
- *Let $j, j' \in I$ be the two child nodes of a node $i \in I$, then $X_j = X_{j'} = X_i$. The node i is called a join node of $\mathcal{T}$.*

Scheffler presented in [10] a special tree-decomposition that follows the structure of a nice tree-decomposition as defined above but with no restriction on the size of leaf bags (i.e., it does not require leaf bags to be of size 2). We call that type of tree-decomposition a Scheffler-type nice tree-decomposition. Any given tree-decomposition for a graph $G = (V, E)$ with treewidth at most tw can be

easily converted into a Scheffler-type nice tree-decomposition of width tw and size $O(|V|)$ in linear time, if tw is a fixed constant [10]. One can also transform a Scheffler-type nice tree-decomposition to have leaves with bags of size 2 in linear time. Therefore, a nice tree-decomposition with leaf bags of size 2 and width tw can be constructed from any given tree-decomposition of the same treewidth. However, each leaf node of a Scheffler-type nice tree-decomposition may produce $O(tw^3)$ new nodes in the construction of leaves with size 2. The resulting tree-decomposition, therefore, has $O(|V|tw^3)$ nodes.

Throughout this paper, we consider simple undirected graphs $G = (V, E)$ given as an input with a nice tree-decomposition of width at most tw. We define X_i^+ to be the set of all vertices in X_j for all nodes $j \in I$ such that $j = i$ or j is a descendant of i. We denote by G_i^+ a partial graph of G. For a leaf node i, G_i^+ is the subgraph of G with vertex set X_i and the edge of G that is associated with i. For a non-leaf node i, G_i^+ is the graph that is the union of G_j^+ over all children j of i. Note that the graph G_r^+ for the root r of the tree-decomposition is equal to G.

3 Minimum Activation Cost k Edge-Disjoint st-Paths

Given are an activation network $G = (V, E)$ and a pair of source and destination vertices $s, t \in V$. In this section, we consider the st-MAEDP problem where the goal is to find activation values $\{x_v : v \in V\}$ of minimum total cost $\sum_{v \in V} x_v$ such that the activated set of edges contains k edge-disjoint st-paths $\mathcal{P}^{st} = \mathcal{P}_1, \ldots, \mathcal{P}_k$. We present a polynomial-time algorithm that solves the st-MAEDP problem optimally in the case of graphs of bounded tree-width using dynamic programming techniques. The algorithm follows a bottom-up approach to compute a number of possible sub-solutions per nice tree-decomposition node $i \in I$. It is easy to compute the sub-solutions for a leaf node $i \in I$ because the partial graph G_i^+ consists of two vertices that are connected by an edge in G associated with i. For a non-leaf node, we use the information previously computed for its children. The algorithm also constructs a table tab_i to store the computed information for each node $i \in I$.

We use an edge-coloring scheme to compute the sub-solutions per tree node. Let $C = \{0, 1, \ldots, k\}$ be a set of colors. We consider a coloring $f_i : E(G_i^+) \to C$ for the graph G_i^+. For each color $c \in C$, we define $G_i^+(f_i, c)$ to be the subgraph of G_i^+ induced by the edges with color c. Each color $c \in C \setminus \{0\}$ represents the edges used by $\mathcal{P}_c$. Denote by $P(X)$ the set of all possible partitions of the set X. We define $C(X_i) = (\mathcal{Y}_1, \ldots, \mathcal{Y}_k)$ to be a *color vector* on X_i such that $\mathcal{Y}_c \in P(X_i \cup \{s, t\})$ for all $c \in C \setminus \{0\}$. $P_{st}(X_i)$ denotes the set $P(X_i \cup \{s, t\})$. A color vector $C(X_i) = (\mathcal{Y}_1, \ldots, \mathcal{Y}_k)$ on X_i is called *active* if G_i^+ has a coloring f_i such that every element of the partition $\mathcal{Y}_c$, for each $c \in C \setminus \{0\}$, is a set resulting from the intersection between $X_i \cup \{s, t\}$ and the vertex set of a connected component of $G_i^+(f_i, c)$ (see Fig. 1 for an example of an active color vector). $\mathcal{Y}(X_i, f_i, c)$ denotes the partition $\mathcal{Y}_c$ of $X_i \cup \{s, t\}$. This color vector concept was introduced in [12] for developing a polynomial-time algorithm that optimally

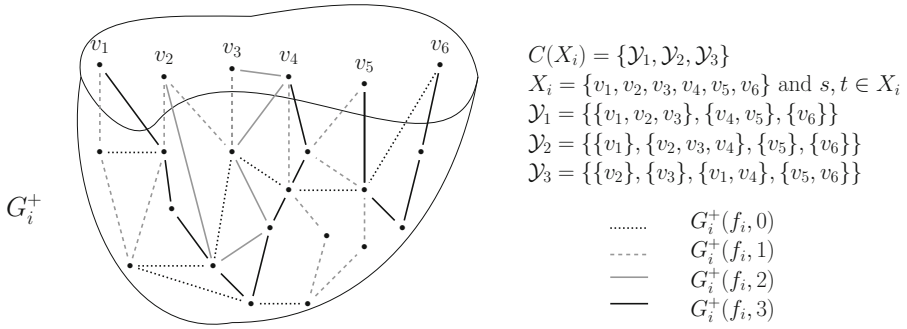

Fig. 1. An example of an active color vector

solves the MSEP problem for graphs with bounded treewidth. One simple way to compute the sub-solutions is by storing a color vector and an activation-value function per each row of table tab_i. However, the number of color vectors per node i can only be bounded by $(tw+3)^{k(tw+3)}$ because $|X_i| \leq tw+1$, $|X_i \cup \{s,t\}| \leq tw+3$ and $|P_{st}(X_i)| \leq (tw+3)^{tw+3}$. Therefore, the number of color vectors is not always polynomially bounded. Hence the size of the table tab_i is also not always polynomially bounded. Therefore we define a mapping $\gamma_i : P_{st}(X_i) \to \{0, 1, \ldots, k\}$ to be a *count* on X_i to obtain a polynomial-time algorithm to optimally solve the *st*-MAEDP problem. We say that the count γ_i on X_i is an *active count* if G_i^+ has a coloring f_i with a color vector $C(X_i) = (\mathcal{Y}_1, \ldots, \mathcal{Y}_k)$ such that $\gamma_i(A) = |\{c \in C \setminus \{0\} : A = \mathcal{Y}_c\}|$ for each $A \in P_{st}(X_i)$. For any active count γ_i, it is clear that $\sum_{A \in P_{st}(X_i)} \gamma_i(A) = k$. In Fig. 1, the counts would be 1 for all partitions $A \in \{\mathcal{Y}_1, \mathcal{Y}_2, \mathcal{Y}_3\} \subseteq P_{st}(X_i)$ and 0 otherwise. Consider a solution $\mathcal{P} = \mathcal{P}_1, \ldots, \mathcal{P}_k$ for the *st*-MAEDP problem. Let $\mathcal{P}^i = \mathcal{P}[G_i^+]$ be the induced solution in a partial graph G_i^+ (i.e., the set of vertices and edges that are both in $\mathcal{P}$ and in G_i^+). Since the interaction between $\mathcal{P}^i$ in G_i^+ and the rest of the graph happens only in vertices of X_i, we can consider an activation-value function $\Lambda_i : X_i \to D$ and a count $\gamma_i : P_{st}(X_i) \to \{0, 1, \ldots, k\}$ to represent $\mathcal{P}^i$ in G_i^+. The idea of using counts instead of color vectors is based on [12].

3.1 Processing the Tree Decomposition

For a tree node $i \in I$, the table tab_i has multiple rows and each row represents a solution of a unique combination of a count γ_i and a function of activation values Λ_i. Let $val(\gamma_i, \Lambda_i)$ denote the minimum cost value of an assignment of activation values for G_i^+ which satisfies the restrictions Λ_i and activates an edge-colored subgraph of G_i^+ that satisfies the count γ_i. The value $val(\gamma_i, \Lambda_i)$ is also stored in tab_i. We compute the sub-solution tables starting at the leaves towards the root.

Leaf. Let $i \in I$ be a leaf, $X_i = \{u, v\}$. Let (γ_i, Λ_i) be any row of tab_i. We distinguish the following cases and define the value $val(\gamma_i, \Lambda_i)$ for each case.

Each case corresponds to a possible sub-solution in G_i^+. Recall that G_i^+ is a single edge. The sub-solution's cost $val(\gamma_i, \Lambda_i)$ is set to $\sum_{v \in X_i} \Lambda_i(v)$ if one of the following cases applies:

- $\gamma_i(A') = k-1$ for $A' = \{\{u\},\{v\},\{s\},\{t\}\}$ and $\gamma_i(A'') = 1$ for $A'' = \{\{u,v\},\{s\},\{t\}\}$ and $\gamma_i(A) = 0$ for all $A \in P_{st}(X_i) \setminus \{A', A''\}$, where $f_{uv}(\Lambda_i(u), \Lambda_i(v)) = 1$ and $s,t \notin X_i$. Intuitively, this means that the sub-solution is a path with one edge not containing s or t.
- $\gamma_i(A') = k-1$ for $A' = \{\{u\},\{v\}\}$ and $\gamma_i(A'') = 1$ for $A'' = \{\{u,v\}\}$ and $\gamma_i(A) = 0$ for all $A \in P_{st}(X_i) \setminus \{A', A''\}$, where $f_{uv}(\Lambda_i(u), \Lambda_i(v)) = 1$ and $s,t \in X_i$. Intuitively, this means that the sub-solution is a path with one edge containing s and t.
- $\gamma_i(A') = k-1$ for $A' = \{\{u\},\{v\},\{s\}\}$ and $\gamma_i(A'') = 1$ for $A'' = \{\{u,v\},\{s\}\}$ and $\gamma_i(A) = 0$ for all $A \in P_{st}(X_i) \setminus \{A', A''\}$, where $f_{uv}(\Lambda_i(u), \Lambda_i(v)) = 1$ and $s \notin X_i$ and $t \in X_i$. Intuitively, the sub-solution is a path with one edge and one endpoint equal to t. (The roles of s and t can be exchanged.)
- $\gamma_i(A') = k$ for $A' = \{\{u\},\{v\},\{s\},\{t\}\}$ and $\gamma_i(A) = 0$ for all $A \in P_{st}(X_i) \setminus \{A'\}$, where $s,t \notin X_i$. Intuitively, this means that the sub-solution has no edges.
- $\gamma_i(A') = k$ for $A' = \{\{u\},\{v\}\}$ and $\gamma_i(A) = 0$ for all $A \in P_{st}(X_i) \setminus \{A'\}$, where $s,t \in X_i$. Intuitively, this means that the sub-solution has no edges.
- $\gamma_i(A') = k$ for $A' = \{\{u\},\{v\},\{s\}\}$ and $\gamma_i(A) = 0$ for all $A \in P_{st}(X_i) \setminus \{A'\}$, where $s \notin X_i$ and $t \in X_i$. Intuitively, this means that the sub-solution has no edges. (The roles of s and t can be exchanged.)

In these cases we construct an edge-colored subgraph of G_i^+ plus activation-values that may be part of a global solution. If none of the above cases applies, $val(\gamma_i, \Lambda_i) = +\infty$.

Introduce. Let $i \in I$ be an introduce node, and $j \in I$ its only child. We have $X_j \subset X_i$, $|X_i| = |X_j| + 1$ and let v be the additional vertex in X_i. The vertex v is isolated since i does not introduce edges in G_i^+. For every row (γ_j, Λ_j) in tab_j, there are $|D|$ rows in tab_i such that for all $A \in P_{st}(X_j)$ and all $u \in X_i \setminus \{v\}$, $\gamma_i(A \cup \{\{v\}\}) = \gamma_j(A)$ if $v \notin \{s,t\}$ and $\gamma_i(A) = \gamma_j(A)$ if $v \in \{s,t\}$ and $\Lambda_i(u) = \Lambda_j(u)$. The sub-solution's cost $val(\gamma_i, \Lambda_i)$ for these rows is set to $val(\gamma_j, \Lambda_j) + \Lambda_i(v)$.

Forget. Let $i \in I$ be a forget node, and $j \in I$ its only child. We have $X_i \subset X_j$, $|X_j| = |X_i| + 1$ and let v be the discarded vertex. Let A^{-v} be the partition $A \in P_{st}(X_j)$ after removing the vertex v if $v \notin \{s,t\}$ and A^{-v} equals the partition A if $v \in \{s,t\}$. Note that $A^{-v} \in P_{st}(X_i)$ for all $A \in P_{st}(X_j)$. For $A \in P_{st}(X_i)$, we say that $W(A)$ is the set of all partitions $B \in P_{st}(X_j)$ such that $B^{-v} = A$ (i.e., $W(A) = \{B \in P_{st}(X_j) : B^{-v} = A\}$). For each row (γ_i, Λ_i), we consider all (γ_j, Λ_j) such that for all $u \in X_i$ and $A \in P_{st}(X_i)$, $\Lambda_i(u) = \Lambda_j(u)$ and $\gamma_i(A) = \sum_{B \in W(A)} \gamma_j(B)$. The sub-solution's cost $val(\gamma_i, \Lambda_i)$ is the minimum of $val(\gamma_j, \Lambda_j)$ over all these (γ_j, Λ_j).

Join. Let $i \in I$ be a join node, and $j, j' \in I$ its two children. We have $X_i = X_j = X_{j'}$. We call a mapping $\beta_i : P_{st}(X_i) \times P_{st}(X_i) \to \{0, 1, \ldots, k\}$ a pair-count on X_i. We define β_i to be an active pair-count if G_i^+ has a coloring f_i such that $\beta_i(A_j, A_{j'}) = |\{c \in C : A_j = \mathcal{Y}(X_j, f_j, c) \text{ and } A_{j'} = \mathcal{Y}(X_{j'}, f_{j'}, c)\}|$ for each $A_j, A_{j'} \in P_{st}(X_i)$ where f_j and $f_{j'}$ are the restriction of f_i to $E(G_j^+)$ and $E(G_{j'}^+)$, respectively. Let $val(\beta_i, \Lambda_i)$ denote the minimum cost value of an assignment of activation values for G_i^+ which satisfies the restrictions Λ_i and activates an edge-colored subgraph of G_i^+ that satisfies the pair-count β_i. The algorithm computes all $val(\beta_i, \Lambda_i)$ of all β_i on X_i from all pairs of sub-solutions (γ_j, Λ_j) and $(\gamma_{j'}, \Lambda_{j'})$ such that both have the same activation-value function ($\Lambda_j(u) = \Lambda_{j'}(u)$ for all $u \in X_i$) and the pair of active counts γ_j and $\gamma_{j'}$ satisfy the following conditions:

C_1. $\gamma_j(A_j) = \sum_{A \in P_{st}(X_i)} \beta_i(A_j, A)$ for all $A_j \in P_{st}(X_i)$.
C_2. $\gamma_{j'}(A_{j'}) = \sum_{A \in P_{st}(X_i)} \beta_i(A, A_{j'})$ for all $A_{j'} \in P_{st}(X_i)$.

The value $val(\beta_i, \Lambda_i)$ is set to be the summation value of the pair of sub-solutions (γ_j, Λ_j) and $(\gamma_{j'}, \Lambda_{j'})$ that satisfy the above conditions minus the activation cost of X_i. To determine the pair (γ_i, Λ_i) that corresponds to the pair (β_i, Λ_i) of pair-count and an activation-value function, we construct a bipartite graph for each pair of partitions with a pair-count greater than 0 as follows. For each pair $A_j, A_{j'} \in P_{st}(X_i)$ where $\beta_i(A_j, A_{j'}) \geq 1$, we construct a bipartite graph $H_{\beta_i}^{(A_j, A_{j'})} = (A_j \cup A_{j'}, E_{\beta_i}^{(A_j, A_{j'})})$ with partite sets A_j and $A_{j'}$, where the vertices $a_j \in A_j$ and $a_{j'} \in A_{j'}$ are joined by an edge in $E_{\beta_i}^{(A_j, A_{j'})}$ iff $a_j \cap a_{j'} \neq \emptyset$. Assume that $D_1, D_2, \ldots, D_b$ are the connected components of the graph $H_{\beta_i}^{(A_j, A_{j'})}$. We define $\mathcal{U}(A_j, A_{j'})$ to be the family of vertex sets $\{\bigcup_{v \in D_l} v : 1 \leq l \leq b\}$. We set the value $val(\gamma_i, \Lambda_i)$ to be the minimum $val(\beta_i, \Lambda_i)$ over all (β_i, Λ_i) such that for each $A_i \in P_{st}(X_i)$:

$$\gamma_i(A_i) = \sum \beta_i(A_j, A_{j'})$$

where the summation above is taken over all pairs $A_j, A_{j'} \in P_{st}(X_i)$ such that $A_i = \mathcal{U}(A_j, A_{j'})$.

Extracting the Solution at the Root. The algorithm checks all the pairs (γ_r, Λ_r) of the root bag X_r such that for all $A \in P_{st}(X_r)$ where $\gamma_r(A) \geq 1$, there is a set in A containing both s and t. In this case (γ_r, Λ_r) corresponds to a feasible solution. The output of the algorithm is the minimum cost value among all the feasible solutions obtained at the root. For each row (γ_i, Λ_i) of bag X_i, we store the rows of X_i's children that were used in the calculation of $val(\gamma_i, \Lambda_i)$. Computing the optimum solution is possible by traversing top-down in the decomposition tree to the leaves (traceback) to get the activation values, and then running a maximum flow algorithm on the unit-capacity graph of the activated edges to get the k edge-disjoint st-paths.

3.2 Analysis

The following lemmas analyse the running time of the algorithm and show that we can efficiently compute an optimal solution for the *st*-MAEDP problem for graphs with bounded treewidth. Let an instance of the problem be given by an activation network $G = (V, E)$ with treewidth bounded by tw and terminals $s, t \in V$. Let $\mathcal{P}^{OPT}$ represent an optimal solution for this instance. We use $\mathcal{C}(\mathcal{P}^i)$ to denote the activation cost of a sub-solution $\mathcal{P}^i$ in a partial graph G_i^+.

Lemma 1. *The st-MAEDP algorithm requires* $O(|V||D|^{tw+1}(k+1)^{(tw+3)^{2(tw+3)}}(tw+3)^{2(tw+3)+3}tw^3)$ *time.*

Proof. The running time of the algorithm depends on the size of the tables and the combination of tables during the bottom-up traversal. For each set of vertices X, there are at most $|X|^{|X|}$ possible partitions. Therefore, for each node i, $|P_{st}(X_i)| \leq (|X_i|+2)^{(|X_i|+2)} \leq (tw+3)^{(tw+3)}$. That means there are at most $(k+1)^{(tw+3)^{(tw+3)}}$ possible active counts γ on X_i. The table tab_i in a processed bag X_i contains no more than $(k+1)^{(tw+3)^{(tw+3)}}|D|^{tw+1}$ rows corresponding to the possible active counts γ and the $|D|$ possible activation values for each vertex of X_i. Consider all possible row combinations with equal activation functions for two tables for a join node. Since there are at most $(tw+3)^{2(tw+3)}$ possible pairs of partitions, there are at most $(k+1)^{(tw+3)^{2(tw+3)}}$ possible active pair-counts on X_i. For each pair-count β_i, there is a pair of active counts γ_j and $\gamma_{j'}$ such that the conditions C_1 and C_2 are satisfied. Computing γ_j and $\gamma_{j'}$ from β_i takes $O((tw+3)^{2(tw+3)})$ time. Therefore, the computation of the value $val(\beta, \Lambda)$ for each combination of pair-count β and activation function Λ needs $O((tw+3)^{2(tw+3)})$ time. Thus we see that all pairs (β, Λ) and $val(\beta, \Lambda)$ can be computed in $O((k+1)^{(tw+3)^{2(tw+3)}}(tw+3)^{2(tw+3)}|D|^{tw+1})$ time. Since $|A| \leq tw+3$ for all $A \in P_{st}(X_i)$, the bipartite graph $H_{\beta_i}^{(A_j, A_{j'})}$ defined in the join node contains no more than $(tw+3)^2$ edges and can be constructed in $O((tw+3)^3)$ time. Therefore, for each β_i, one can compute the active count γ_i that satisfies $\gamma_i(A_i) = \sum \beta_i(A_j, A_{j'})$ where the summation is taken over all $A_j, A_{j'} \in P_{st}(X_i)$ such that $A_i = \mathcal{U}(A_j, A_{j'})$ and update the value $val(\gamma_i, \Lambda_i)$ to be equal to $val(\beta_i, \Lambda_i)$ if $val(\beta_i, \Lambda_i) \leq val(\gamma_i, \Lambda_i)$ in $O((tw+3)^{2(tw+3)}(tw+3)^3)$ time. Thus, we can compute all $val(\gamma, \Lambda)$ of all combinations of active count γ and activation-value function Λ on X_i in time $O((k+1)^{(tw+3)^{2(tw+3)}}(tw+3)^{2(tw+3)+3}|D|^{tw+1})$. Since the tree-decomposition T has $O(|V|tw^3)$ nodes, one can compute all the tables for all nodes in $O(|V||D|^{tw+1}tw^3(k+1)^{(tw+3)^{2(tw+3)}}(tw+3)^{2(tw+3)+3})$ time. □

Lemma 2. *For any processed bag* X_i*, let* $\mathcal{P}_i^{OPT}$ *be the induced solution of* $\mathcal{P}^{OPT}$ *in* G_i^+ *and* $(\gamma_i^{OPT}, \Lambda_i^{OPT})$ *be the corresponding count and activation values, then*

$$val(\gamma_i^{OPT}, \Lambda_i^{OPT}) \leq \mathcal{C}(\mathcal{P}_i^{OPT}) \tag{1}$$

Proof. We use induction over the tree decomposition to prove that the value of $(\gamma_i^{OPT}, \Lambda_i^{OPT})$ is at most the activation cost of $\mathcal{P}_i^{OPT}$. The base case are the

leaf nodes where the hypothesis clearly holds. Let us assume that the induction hypothesis holds for all descendants of bag X_i. We want to prove that the induction hypothesis holds also for X_i and that can be proven by showing that the hypothesis holds for all different types of the bag X_i.

Introduce. Let us assume that i is an introduce node and j its only child and let v be the additional vertex in X_i. The statement (1) holds for X_j because it is a child of the bag X_i. Let $(\gamma_j^{OPT}, \Lambda_j^{OPT})$ be the corresponding active count and activation values of $\mathcal{P}_j^{OPT}$. The following statement holds:

$$val(\gamma_j^{OPT}, \Lambda_j^{OPT}) \leq \mathcal{C}(\mathcal{P}_j^{OPT})$$

$(\gamma_i^{OPT}, \Lambda_i^{OPT})$ and $(\gamma_j^{OPT}, \Lambda_j^{OPT})$ both agree on the activation values for all vertices in $X_i \setminus \{v\}$. The additional vertex v is an isolated vertex in the induced graph G_i^+ and the one vertex set $\{v\}$ is in A for all $A \in P_{st}(X_i)$ such that $\gamma_i^{OPT}(A) \geq 1$. Since $\Lambda_i^{OPT}(v) \in D$ is the activation value of the vertex v, then:

$$\begin{aligned} val(\gamma_i^{OPT}, \Lambda_i^{OPT}) &= val(\gamma_j^{OPT}, \Lambda_j^{OPT}) + \Lambda_i^{OPT}(v) \\ &\leq \mathcal{C}(\mathcal{P}_j^{OPT}) + \Lambda_i^{OPT}(v) \\ &= \mathcal{C}(\mathcal{P}_i^{OPT}) \end{aligned}$$

The statement (1) holds for an introduce node.

Forget. Assume that i is a forget node and j its only child and let v be the discarded vertex in X_i. The statement (1) holds for X_j because it is a child of the bag X_i. Let $(\gamma_j^{OPT}, \Lambda_j^{OPT})$ be the corresponding active count and activation values of $\mathcal{P}_j^{OPT}$. The following statement holds:

$$val(\gamma_j^{OPT}, \Lambda_j^{OPT}) \leq \mathcal{C}(\mathcal{P}_j^{OPT})$$

$(\gamma_i^{OPT}, \Lambda_i^{OPT})$ and $(\gamma_j^{OPT}, \Lambda_j^{OPT})$ both agree on activation values for all vertices in $X_j \setminus \{v\}$. The discarded vertex v is either an isolated vertex or part of a connected component in the induced solution $\mathcal{P}_j^{OPT}$ and that means the value $val(\gamma_j^{OPT}, \Lambda_j^{OPT})$ is one of the values that has been considered when calculating $val(\gamma_i^{OPT}, \Lambda_i^{OPT})$. Then:

$$val(\gamma_i^{OPT}, \Lambda_i^{OPT}) \leq val(\gamma_j^{OPT}, \Lambda_j^{OPT}) \leq \mathcal{C}(\mathcal{P}_j^{OPT}) = \mathcal{C}(\mathcal{P}_i^{OPT})$$

Then the statement (1) holds for a forget node.

Join. Assume that i is a join node and j and j' its children. Let $(\gamma_i^{OPT}, \Lambda_i^{OPT})$ be the corresponding active count and activation values of the induced solution $\mathcal{P}_i^{OPT}$ in G_i^+. $\mathcal{P}_i^{OPT}$ is the union of its children sub-solutions $\mathcal{P}_j^{OPT}$ and $\mathcal{P}_{j'}^{OPT}$. Let $(\gamma_j^{OPT}, \Lambda_j^{OPT})$ and $(\gamma_{j'}^{OPT}, \Lambda_{j'}^{OPT})$ be the corresponding active count and activation values of $\mathcal{P}_j^{OPT}$ and $\mathcal{P}_{j'}^{OPT}$, respectively. The statement (1) holds for X_j and $X_{j'}$ because they are children of the bag X_i. For all $l \in \{j, j'\}$ the following statement holds:

$$val(\gamma_l^{OPT}, \Lambda_l^{OPT}) \leq \mathcal{C}(\mathcal{P}_l^{OPT})$$

Let $(\beta_i^{OPT}, \Lambda_i^{OPT})$ be the corresponding active pair-count and activation values of the induced solution $\mathcal{P}_i^{OPT}$ in G_i^+ which satisfies conditions C_1 and C_2 for $(\gamma_j^{OPT}, \Lambda_j^{OPT})$ and $(\gamma_{j'}^{OPT}, \Lambda_{j'}^{OPT})$. We know that $\mathcal{P}_i^{OPT}$ is a sub-solution in G_i^+, therefore, $val(\beta_i^{OPT}, \Lambda_i^{OPT})$ is one of the values that has been considered when computing $val(\gamma_i^{OPT}, \Lambda_i^{OPT})$. Then:

$$\begin{aligned} val(\gamma_i^{OPT}, \Lambda_i^{OPT}) &\leq val(\beta_i^{OPT}, \Lambda_i^{OPT}) \\ &= val(\gamma_j^{OPT}, \Lambda_j^{OPT}) + val(\gamma_j^{OPT}, \Lambda_j^{OPT}) - \sum_{v \in X_i} \Lambda_i^{OPT}(v) \\ &\leq \mathcal{C}(\mathcal{P}_j^{OPT}) + \mathcal{C}(\mathcal{P}_{j'}^{OPT}) - \sum_{v \in X_i} \Lambda_i^{OPT}(v) \\ &= \mathcal{C}(\mathcal{P}_i^{OPT}) \end{aligned}$$

Then the induction hypothesis holds for a join node. □

Lemma 3. *For any processed bag X_i, any pair (γ_i, Λ_i) where $val(\gamma_i, \Lambda_i) = c_i < \infty$ corresponds to an edge-coloring plus activation-values $\mathcal{P}^i = (f_i, \Lambda_i^+)$ where $f_i : E(G_i^+) \to \{0, \ldots, k\}$ and $\Lambda_i^+ : X_i^+ \to D$ with the following properties:*

- *The active count of $\mathcal{P}^i$ in X_i is γ_i.*
- *The activation values of $\mathcal{P}^i$ in X_i are Λ_i.*
- *The total activation cost in X_i^+ is c_i.*

Proof. We prove by induction that for any bag X_i there exists an edge-coloring and activation-values $\mathcal{P}^i$ with the above properties. The base case are the leaf nodes where the hypothesis clearly holds. Let us assume that the induction hypothesis holds for all the descendants of bag X_i. We want to prove that the induction hypothesis holds also for X_i and that can be proved by showing that the hypothesis holds for all different types of the bag X_i.

Introduce. Assume that i is an introduce node and j its only child and v the additional vertex in X_i. Let (γ_i, Λ_i) be some entry with $val(\gamma_i, \Lambda_i) = c_i < \infty$. The induction hypothesis holds for X_j. Let (γ_j, Λ_j) be the corresponding active count and activation function that have been used for the calculation of $val(\gamma_i, \Lambda_i)$. Then (γ_j, Λ_j) corresponds to an edge-coloring and activation-values $\mathcal{P}^j$ that satisfies the above properties. From the algorithm, (γ_i, Λ_i) and (γ_j, Λ_j) both agree on the activation values for all vertices in $X_i \setminus \{v\}$. Moreover, the additional vertex v is an isolated vertex in the induced graph G_i^+ and the one vertex set $\{v\}$ is in A for all $A \in P_{st}(X_i)$ such that $\gamma_i(A) \geq 1$. The value of (γ_i, Λ_i) is equal to the value of (γ_j, Λ_j) added to the activation value of the vertex v. The union of the isolated vertex v and $\mathcal{P}^j$ is an edge-coloring with activation-values that satisfies all properties of the induction hypothesis. Thus, the induction hypothesis holds for an introduce node.

Forget. Assume that i is a forget node and j its only child and v the discarded vertex. Let (γ_i, Λ_i) be some entry with $val(\gamma_i, \Lambda_i) = c_i < \infty$. The induction hypothesis holds for X_j. Let (γ_j, Λ_j) be the corresponding active count and

activation function that have been used for the calculation of $val(\gamma_i, \Lambda_i)$. Then (γ_j, Λ_j) corresponds to an edge-coloring and activation-values $\mathcal{P}^j$ that satisfies the above properties. From the algorithm, (γ_j, Λ_j) has the minimum value cost among all possible rows of X_j that can produce (γ_i, Λ_i) and both agree on activation values for all vertices in $X_j \setminus \{v\}$. The discarded vertex v is part of the edge-coloring $\mathcal{P}^j$. Therefore (γ_i, Λ_i) corresponds to the edge-coloring and activation-values $\mathcal{P}^j$ and we can set $\mathcal{P}^i = \mathcal{P}^j$. The induction hypothesis holds for a forget node.

Join. Assume that i is a join node and j and j' its children. Let (γ_i, Λ_i) be some entry with $val(\gamma_i, \Lambda_i) = c_i < \infty$. Since $val(\gamma_i, \Lambda_i) = c_i < \infty$, then there is a pair of active pair-count and activation function (β_i, Λ_i) such that $val(\gamma_i, \Lambda_i) = val(\beta_i, \Lambda_i)$ and for each $A_i \in P_{st}(X_i)$, $\gamma_i(A_i) = \sum \beta_i(A_j, A_{j'})$ where the summation is taken over all pairs $A_j, A_{j'} \in P_{st}(X_i)$ satisfying $A_i = \mathcal{U}(A_j, A_{j'})$. Let (γ_j, Λ_j) and $(\gamma_{j'}, \Lambda_{j'})$ be the pairs that have been used for the calculation of $val(\beta_i, \Lambda_i)$ which satisfy the conditions C_1 and C_2. We know that $val(\gamma_j, \Lambda_j) < \infty$ and $val(\gamma_{j'}, \Lambda_{j'}) < \infty$ and the induction hypothesis holds for X_j and $X_{j'}$. Therefore, (γ_j, Λ_j) and $(\gamma_{j'}, \Lambda_{j'})$ correspond to edge-colorings with activation-values $\mathcal{P}^j = (f_j, \Lambda_j^+)$ and $\mathcal{P}^{j'} = (f_{j'}, \Lambda_{j'}^+)$, respectively. For each pair of partitions A_j and $A_{j'}$ where $\gamma_j(A_j) = r_j$, $\gamma_{j'}(A_{j'}) = r_{j'}$ and $\beta_i(A_j, A_{j'}) = r > 0$, there are r_l colors for partition A_l in $\mathcal{P}^l = (f_l, \Lambda_l)$ for $l \in \{j, j'\}$. Therefore, we choose r unused colors $z_1^j, z_2^j, \ldots, z_r^j$ for A_j in $\mathcal{P}^j$ and r unused colors $z_1^{j'}, z_2^{j'}, \ldots, z_r^{j'}$ for $A_{j'}$ in $\mathcal{P}^{j'}$ and then recolor both z_w^j in $G_j^+(f_j, z_w^j)$ and $z_w^{j'}$ in $G_{j'}^+(f_{j'}, z_w^{j'})$ with a new color $z_w^{jj'}$ for all $w \in \{1, \ldots, r\}$. The colors $z_1^j, z_2^j, \ldots, z_r^j$ in $\mathcal{P}^j$ and colors $z_1^{j'}, z_2^{j'}, \ldots, z_r^{j'}$ in $\mathcal{P}^{j'}$ are now marked as used. Note that there are always enough unused colors because the conditions C_1 and C_2 are satisfied. After recoloring $\mathcal{P}^j$ and $\mathcal{P}^{j'}$ and since for each $A_i \in P_{st}(X_i)$, $\gamma_i(A_i) = \sum \beta_i(A_j, A_{j'})$ where the summation is taken over all pairs $A_j, A_{j'} \in P_{st}(X_i)$ satisfying $A_i = \mathcal{U}(A_j, A_{j'})$, it follows that the active count of the union $\mathcal{P}^j \cup \mathcal{P}^{j'}$ in X_i is γ_i. Moreover, (γ_i, Λ_i), (γ_j, Λ_j) and $(\gamma_{j'}, \Lambda_{j'})$ all have the same activation-value function $(\Lambda_i(u) = \Lambda_j(u) = \Lambda_{j'}(u)$ for all $u \in X_i)$. That means the activation values of the union $\mathcal{P}^j \cup \mathcal{P}^{j'}$ in X_i are Λ_i. The algorithm also computes the cost value of (γ_i, Λ_i) as follows:

$$val(\gamma_i, \Lambda_i) = val(\beta_i, \Lambda_i) = val(\gamma_j, \Lambda_j) + val(\gamma_{j'}, \Lambda_{j'}) - \sum\nolimits_{v \in X_i} \Lambda_i(v)$$

Since $val(\gamma_j, \Lambda_j)$ and $val(\gamma_{j'}, \Lambda_{j'})$ are the total activation costs in X_j^+ and $X_{j'}^+$, respectively, then the summation of these activation costs minus the activation values for all $u \in X_i$ is the total activation cost in X_i^+. We can set $\mathcal{P}^i$ to be the edge-coloring and activation values $\mathcal{P}^j \cup \mathcal{P}^{j'}$ that satisfies the properties of the lemma. The induction hypothesis holds for a join node. □

We obtain the following theorem by combining the above lemmas.

Theorem 2. *The st-MAEDP problem for graphs with treewidth bounded by tw can be solved optimally in* $O(|V||D|^{tw+1} tw^3 (k+1)^{(tw+3)^{2(tw+3)}} (tw+3)^{2(tw+3)+3})$ *time.*

Corollary 1. *For any fixed k, the st-MAEDP problem for graphs of bounded treewidth can be solved optimally in linear-time FPT parameterized by the treewidth tw.*

4 Conclusion

To the best of our knowledge, the *st*-MAEDP problem for graphs with treewidth bounded by tw considered here has not been addressed before. We established an algorithm that solves the problem optimally in $O(|V||D|^{tw+1}tw^3(k+1)^{(tw+3)^{2(tw+3)}}(tw+3)^{2(tw+3)+3})$ time. Our algorithm also solves the *st*-MAEDP problem when $k = 2$ in linear-time and this is an improvement over the cubic algorithm obtained in [2]. It would be interesting if one can obtain a faster or even linear-time algorithm for the *st*-MAEDP problem in graphs with treewidth bounded by a constant.

References

1. Alqahtani, H.M., Erlebach, T.: Approximation algorithms for disjoint *st*-paths with minimum activation cost. In: Spirakis, P.G., Serna, M. (eds.) CIAC 2013. LNCS, vol. 7878, pp. 1–12. Springer, Heidelberg (2013)
2. Alqahtani, H.M., Erlebach, T.: Minimum activation cost node-disjoint paths in graphs with bounded treewidth. In: Geffert, V., Preneel, B., Rovan, B., Štuller, J., Tjoa, A.M. (eds.) SOFSEM 2014. LNCS, vol. 8327, pp. 65–76. Springer, Heidelberg (2014)
3. Bodlaender, H.L.: A linear time algorithm for finding tree-decompositions of small treewidth. In: STOC 1993. ACM, pp. 226–234 (1993)
4. Hajiaghayi, M.T., Kortsarz, G., Mirrokni, V.S., Nutov, Z.: Power optimization for connectivity problems. In: Jünger, M., Kaibel, V. (eds.) IPCO 2005. LNCS, vol. 3509, pp. 349–361. Springer, Heidelberg (2005)
5. Lando, Y., Nutov, Z.: On minimum power connectivity problems. In: Arge, L., Hoffmann, M., Welzl, E. (eds.) ESA 2007. LNCS, vol. 4698, pp. 87–98. Springer, Heidelberg (2007)
6. Nutov, Z.: Survivable network activation problems. In: Fernández-Baca, D. (ed.) LATIN 2012. LNCS, vol. 7256, pp. 594–605. Springer, Heidelberg (2012)
7. Nutov, Z.: Approximating Steiner networks with node-weights. SIAM J. Comput. **39**(7), 3001–3022 (2010)
8. Panigrahi, D.: Survivable network design problems in wireless networks. In: 22nd Annual SIAM Symposium on Discrete Algorithms, pp. 1014–1027. ACM (2011)
9. Robertson, N., Seymour, P.D.: Graph minors. XIII. The disjoint paths problem. J. Comb. Theory, Ser. B **63**, 65–110 (1995)
10. Scheffler, P.: A practical linear time algorithm for disjoint paths in graphs with bound tree-width. Technical report 396, Dept. Mathematics, Technische Universität Berlin (1994)
11. Srinivas, A., Modiano, E.: Finding minimum energy disjoint paths in wireless ad-hoc networks. Wireless Netw. **11**(4), 401–417 (2005)
12. Ye, Z.Q., Li, Y.M., Lu, H.Q., Zhou, X.: Finding paths with minimum shared edges in graphs with bounded treewidth. In: The 2013 World Congress in Computer Science, Computer Engineering, and Applied Computing, FCS 2013, pp. 40–47 (2013). ISBN: 1601322429

http://www.springer.com/978-3-319-29515-2

Combinatorial Algorithms
26th International Workshop, IWOCA 2015, Verona, Italy, October 5-7, 2015, Revised Selected Papers
Lipták, Z.; Smyth, W.F. (Eds.)
2016, XIX, 360 p. 62 illus. in color., Softcover
ISBN: 978-3-319-29515-2