Finding Paths with Minimum Shared Edges in Graphs with Bounded Treewidth

Z.-Q. Ye¹, Y.-M. Li², H.-Q. Lu³ and X. Zhou⁴
¹Zhejiang University, Hanzhou, Zhejiang, China
²Wenzhou University, Wenzhou, Zhejiang, China
³Zhejiang University of Technology, Hangzhou, Zhejiang, China
⁴Tohoku University, Sendai, Japan

Abstract—Given a positive integer p, a graph G and a pair of two terminals s and t in G, the minimum shared-edge paths problem is to find p paths connecting s and t so as to minimize the number of edges shared among the paths. This is a generalization of the well-known edge-disjoint paths problem which asks whether there exist p pairwise edge-disjoint paths connecting the terminals. The edge-disjoint paths problem is NP-complete for given many pairs of terminals even for graphs with treewidth at most two. In this paper we show that the minimum shared-edge paths problem for a given pair of two terminals can be solved in polynomial time for graphs with bounded treewidth.

Keywords: Dynamic programming algorithm, Tree-decomposition, Treewidth

1. Introduction

Given a number p, a graph G and a pair of two terminals s and t in G, the minimum shared-edge paths problem is to find p paths connecting s and t so as to minimize the number of edges shared among the paths. This problem, introduced in [7], has an application for a security assurance demand in a geographic information system setting. Suppose that a security organization is hired to do planning for a VIP who wishes to travel safely between two locations. Given the security concerns, p paths are determined in pre-trip planning and then, just prior to actual travel, randomly one path among the p paths is chosen. The fewer edges, that are shared among the pre-trip paths, are to make the higher level of perceived security. However, if it becomes unavoidable to share edges among the paths, guards are employed on those shared edges. Since guards take some costs, we want to reduce their total number, that is, to minimize the number of shared edges.

For the special case where the number of shared edges is required to be zero, the minimum shared-edge paths problem is reduced to the “edge-disjoint paths” problem, which is to find p edge-disjoint paths connected s and t and can be solved in polynomial time using standard maximum flow algorithms. However the minimum shared-edge paths problem is NP-hard for general graphs [7].

The class of graphs with treewidth k includes trees (k = 1), series-parallel graphs (k = 2) [11], Halin graphs (k = 3), and k-terminal recursive graphs. Many problems can be solved efficiently for graphs with treewidth bounded by a constant k by a dynamic programming algorithm based on the tree-decomposition [1], [2], [5], [6], [8], [9], [10], [15], while the edge-disjoint paths problem for many pairs of terminals is NP-complete even for k = 2 [13].

In this paper we give a polynomial-time algorithm to solve the minimum shared-edge paths problem for graphs with treewidth bounded by a constant k. Our idea is to formulate the minimum shared-edge paths problem as a new type of an edge-coloring problem, and to bound the size of a dynamic programming (DP) table by \(O((p + 1)(k + 4)^{2k + 8})\), applying and extending techniques developed for the ordinary edge-coloring problem [3], [12], [14]. We use the fact that when doing dynamic programming upward in a tree-decomposition only certain informations of the partial solutions must be kept. These informations concern basically the connectivity amongst the vertices of a basis graph inside a solution. So the state space that is to be remembered is in a sense given by the partitions of the vertex set of a basis graph.

The paper is organized as follows. In Section 2 we present some preliminary definitions. In Section 3 we give a simple algorithm, that cannot always run in polynomial time, for the minimum shared-edge paths problems on graphs with treewidth bounded by a constant k. In Section 4 we modify it to a polynomial-time algorithm. In Section 5 we conclude with a generalization of our algorithm.

2. Terminology and Definitions

In this section we give some definitions. Let \(G = (V, E)\) denote a graph with vertex set \(V\) and edge set \(E\). We often denote by \(V(G)\) and \(E(G)\) the vertex set and the edge set of \(G\), respectively. We denote by \(n\) the number of vertices in \(G\). The paper deals with simple undirected graphs without multiple edges or self-loops. An edge joining vertices \(u\) and \(v\) is denoted by \((u, v)\). For \(E' \subseteq E(G)\), \(G[E']\) denotes the subgraph of \(G\) induced by the edges in \(E'\); \(G[E']\) contains every vertex of \(G\) to which at least one edge in \(E'\) is incident, and hence \(G[E']\) contains no isolated vertex.
We will use notions as: leaf, node, child and root in their usual meaning. A tree-decomposition $T = (V_T, E_T)$ of a graph $G$ is a rooted tree such that the following conditions (A1)–(A6) hold [3]:

(A1) each $X \in V_T$ is a subset of $V(G)$;
(A2) $\bigcup_{X \in V_T} X = V(G)$;
(A3) for each edge $(u, v) \in E(G)$, there is a leaf node $X \in V_T$ such that $u, v \in X$;
(A4) for any three nodes $X_1, X_2, X_3 \in V_T$, if node $X_2$ lies on the path between $X_1$ and $X_3$, then $X_1 \cap X_3 \subseteq X_2$;
(A5) $|V_T| = O(n)$; and
(A6) every internal node $X_i$ in $T$ has exactly two children $X_i$ and $X_r$ such that $X_i = X_l$ or $X_i = X_r$.

The width of $T$ is defined as $\max\{|X| - 1 : X \in V_T\}$, and the treewidth, denoted by $tw(G)$, of $G$ is the minimum $k$ such that $G$ has a tree-decomposition of width $k$. We denote by $X_{01}$ the root of a tree-decomposition. Assume that $k$ is a bounded positive integer. Since a tree-decomposition $T$ of a graph with treewidth $k$ can be found in linear time [3], [4], we may assume that its tree-decomposition $T$ are given.

We next recursively define an edge-set $E_i \subseteq E$ for each node $X_i$ of $T$ as follows. Let $rep : E \rightarrow V_T$ such that $rep(e)$ is a leaf node of $T$ and the two ends of the edge $e$ is in $rep(e)$. If $X_i$ is a leaf of $T$, then let $E_i = \{s \in E | rep(e) = X_i\}$; if $X_i$ is an internal node of $T$ having two children $X_l$ and $X_r$, then let $E_i = E_l \cup E_r$. Thus node $X_i$ of $T$ corresponds to a subgraph $G[E_i]$ of $G$ induced by the edges in $E_i$. The subgraph $G[E_i]$ is often denoted simply by $G_i$. Then $G_i$ is an edge-disjoint union of two subgraphs $G_l$ and $G_r$, which share common vertices only in $X_i$.

3. Simple Algorithm

In this section we give a straightforward dynamic programming algorithm. Although all our algorithms only compute the minimum number $\omega(G, p)$ of shared edges among all $p$ paths connecting $s$ and $t$, they can be easily modified so that they actually find such $p$ paths connecting $s$ and $t$ with minimum number of shared edges.

The main result of this section is the following theorem.

**Theorem 3.1:** Let $G = (V, E)$ be a graph with $n$ vertices given by its tree-decomposition with width $\leq k$. Let $(s, t)$ be a pair of two vertices in $G$, and let $p$ be a positive integer. Then one can compute $\omega(G, p)$ in time $O \left( n \left( p2^{2k(k+1)/2} + p(k + 4)^{2(k+4)p+3} \right) \right)$.

If $k$ is bounded, the first term in the braces above, $p2^{2k(k+1)/2}$, is bounded by a polynomial in $n$ if $p = O(\log n)$. The second term $p(k + 4)^{2(k+4)p+3}$ is also bounded by a polynomial if $p = O(\log n)$ since $p$ is in the single exponent over a constant $k + 4$. On the other hand, both of the terms are bounded by a constant if $p = O(1)$. Thus we have the following corollary.

**Corollary 3.2:** If $p = O(\log n)$, then the minimum shared-edge paths problem can be solved for graphs with bounded treewidth in polynomial time. If $p = O(1)$, then the problem can be solved for graph with bounded treewidth in linear time.

In the remainder of this section we will give a proof of Theorem 3.1. Our idea is to formulate the minimum shared-edge paths problem as a new type of an edge-coloring problem, and then to solve the coloring problem using dynamic programming with a table of size at most $O(k + 4)^{(k+4)p}$. We employ techniques developed for the ordinary edge-coloring problem and the edge-disjoint paths problem [3], [14], [16].

Let $G = (V, E)$ be a graph, and let $(s, t)$ be a pair of two vertices in $V$ called *terminals*. Let $p$ be a positive integer, and let $C = \{1, 2, \cdots, p\}$ be the set of colors. Any mapping $f : E \rightarrow 2^C$ is called a *coloring of graph* $G$. For a color $c \in C$, we denote by $G(f, c)$ the so-called "color class" for $c$, that is, the subgraph of $G$ induced by the edges which are colored by a set of colors including $c$. We call $f$ a *correct coloring* of $G$ if, for each color $c \in C$, $G(f, c)$ has a connected component containing both terminals $s$ and $t$.

Let $\omega(G, f)$, called the *cost of $f$*, be the number of edges which are in at least two graphs $G(f, c), c \in C$. Let $\omega(G, p)$ be the minimum cost among all correct colorings of $G$. The *minimum shared-edge paths problem* is to compute $\omega(G, p)$ for a given graph $G$.

Let $X_i$ be a node of a tree-decomposition $T$ of a graph $G$. We say that a coloring of graph $G_i = G[E_i]$ is *extensible* if it can be extended to a correct coloring of $G = G[E_{01}]$ without changing the coloring of any edge in $E_i$, where $X_{01}$ is the root of $T$.

When doing dynamic programming upward in a tree-decomposition, only certain informations of all extensible colorings must be kept in a DP table. The informations are called "color vectors," and the number of distinct color vectors is bounded by $(k + 4)^{(k+4)p}$, as we show below.

For a set $X$ we denote by $F(X)$ the set of all families of pairwise disjoint subsets of $X$. If $x = |X| \geq 1$, then

$$|F(X)| \leq (x + 1)^{x+1}. \tag{1}$$

For a node $X_i$ of $T$ we call a $p$-tuple $C(X_i) = (Y_1, Y_2, \cdots, Y_p)$ a *color vector* on $X_i$, where $Y_c, 1 \leq c \leq p$, is a family in $F(X_i \cup \{s, t\})$. Simpily we define $F_{st}(X_i) = F(X_i \cup \{s, t\})$. We say that a color vector $C(X_i) = (Y_1, Y_2, \cdots, Y_p)$ on $X_i$ is *active* if $G_i = G[E_i]$ has a coloring $f$ such that $Y_c = Y(X_i; f, c)$ for each color $c \in C$, where

$$Y(X_i; f, c) = \{V(D) \cap (X_i \cup \{s, t\}) | D \text{ is a connected component of } G_i(f, c) \}.$$

Such a vector $C(X_i)$ is called the *color vector of the coloring* $f$. (Thus a color vector indicates which vertices
in $X_i$ are connected to each other or are reachable from terminals in a color class. Observe that there is a special case where one of the sets in $\mathcal{Y}_c$ is just $\{s, t\}$. This encodes the fact that the color $c$ already connects terminals $s$ and $t$ entirely in $G_i$ without using any vertex in $X_i$ except $s$ and $t$.

We now have the following lemma.

Lemma 3.3: Let $X_i$ be any node of a tree-decomposition $T$ of a graph $G$. Let two colorings $f$ and $g$ of $G_i = G[E_i]$ have the same color vector. Then $f$ is extensible if and only if $g$ is extensible.

Proof: It suffices to prove that if $f$ is extensible then $g$ is also extensible. Suppose that $f$ is extensible. Then $f$ can be extended to a correct coloring $f^*$ of $G = (V, E)$, where $f^*(e) = f(e)$ for $e \in E_i$. Let $g^*$ be a coloring of $G$ extended from $g$ as follows: $g^*(e) = g(e)$ for $e \in E_i$, and $g^*(e) = f^*(e)$ for $e \in E - E_i$. Since the subgraph $G_i$ of $G$ is connected to other parts of $G$ only through vertices in $X_i$ and $f$ and $g$ have the same color vector and $f^*$ is a correct coloring, $g^*$ is a correct coloring.

Thus a color vector on $X_i$ characterizes an equivalence class of extensible colorings of $G_i$. Since $|X_i| \leq k + 1$, $|X_i \cup \{s, t\}| \leq k + 3$. Therefore by Eq. (1) we have $|F(X_i \cup \{s, t\})| \leq (k + 4)^k$. Hence there are at most $O((k + 4)^{(k+4)p})$ color vectors $C(X_i) = \{\mathcal{Y}_1, \mathcal{Y}_2, \cdots, \mathcal{Y}_p\}$ on $X_i$. Let

$$\omega(X_i, C) = \min \{\omega(G_i, f) | f \text{ is a coloring of } G_i \text{ with the color vector } C\},$$

and let $\omega(X_i, C) = \infty$ if no such coloring $f$. Then clearly we have the following lemma.

Lemma 3.4: Let $C$ be any color vector on a node $X_i$ of $T$. Then $C$ is active if and only if $\omega(X_i, C) \neq \infty$.

The main step of our algorithm is to compute a table of all $\omega(X_i, C)$ for all active color vectors $C$ on each node of $T$ from leaves to the root $X_0$ of $T$ by means of dynamic programming. From the table on $X_0$ one can easily compute $\omega(G, p)$, as follows.

Lemma 3.5: Let $G$ be a graph with a tree-decomposition $T$ rooted at $X_0$. Then

$$\omega(G, p) = \min_C \omega(X_0, C),$$

where the minimum is taken over all active color vectors $C = \{\mathcal{Y}_1, \mathcal{Y}_2, \cdots, \mathcal{Y}_p\}$ on $X_0$ such that, for each color $c \in C$, there is a set $\mathcal{Y}_c$ containing both $s$ and $t$. Furthermore Eq. (3) can be computed in time $O((k + 4)^{(k+4)p})$.

We first compute the table of $\omega(X_i, C)$ for all active color vectors $C$ on each leaf $X_i$ of $T$ as follows:

1. enumerate all colorings $f : E_i \to 2^C$ of $G_i$; and
2. compute all active color vectors $C(X_i) = \{\mathcal{Y}_1, \mathcal{Y}_2, \cdots, \mathcal{Y}_p\}$ on $X_i$ from the colorings $f$ of $G_i$.

Since $|C| = p$ and $|E_i| \leq k(k+1)/2$ for leaf $X_i$, the number of distinct colorings $f : E_i \to 2^C$ is at most $2^{p(k+1)/2}$. For each coloring $f$ of $G_i$, one can compute the color vector of $f$ in time $O(p)$. Note that $k = O(1)$. Therefore, steps (1) and (2) above and hence all $\omega(G_i, C)$ can be computed for a leaf in time $O(2^p 2^{k(k+1)/2})$. Since $T$ has $O(n)$ leaves, the table on all leaves can be computed in time $O(n 2^{p(2k(k+1)/2)})$, which corresponds to the first term in the braces of the complexity mentioned in Theorem 3.1.

Fig. 1: Venn diagrams of $\mathcal{Y}_{cl}$ and $\mathcal{Y}_{cr}$ with two clusters $Y_{c1}$ and $Y_{c2}$.

We next compute $\omega(X_i, C)$ for all active color vectors $C$ on each internal node $X_i$ of $T$ from leaves to the root. Lemma 3.6 below shows how to compute them on $X_i$ from all active color vectors on the left and right children $X_l$ and $X_r$ of $X_i$. We now introduce a notion of a family $U(\mathcal{Y}_{lc}, \mathcal{Y}_{rc}; W)$ for families $\mathcal{Y}_{lc}, \mathcal{Y}_{rc}$ and a set $W \subseteq V$. Let $C(X_l) = (\mathcal{Y}_{l1}, \mathcal{Y}_{l2}, \cdots, \mathcal{Y}_{lp})$ be an active color vector on $X_l$, and let $C(X_r) = (\mathcal{Y}_{r1}, \mathcal{Y}_{r2}, \cdots, \mathcal{Y}_{rp})$ be an active color vector on $X_r$. Figure 1 illustrates Venn diagrams of $\mathcal{Y}_{lc}$ and $\mathcal{Y}_{rc}$, $c \in C$, where the sets in $\mathcal{Y}_{lc}$ are indicated by circles of solid lines and the sets in $\mathcal{Y}_{rc}$ by circles of dotted lines. All vertices shared by graphs $G_l$ and $G_r$ are contained in $X_i$, and $X_i \subseteq X_l \cup X_r$. Therefore each family $\mathcal{Y}_c$ in a color vector $C(X_i)$ on $X_i$ corresponds to a “cluster” in Fig. 1, formally defined as follows. For each color $c \in C$, let $G_{bc} = (\mathcal{Y}_{lc} \cup \mathcal{Y}_{rc}, E_{bc})$ be a bipartite graph with partite sets $\mathcal{Y}_{lc}$ and $\mathcal{Y}_{rc}$, where a vertex $Y_{lc} \in \mathcal{Y}_{lc}$ and a vertex $Y_{rc} \in \mathcal{Y}_{rc}$ are joined by an edge in $E_{bc}$ if $Y_{lc} \cap Y_{rc} \neq \emptyset$. Let $D_{c1}, D_{c2}, \cdots, D_{cb}$ be the connected components of $G_{bc}$, and for each $j$, $1 \leq j \leq b$, let $Y_{cj} = \bigcup Y_{c \in V(D_{cj})}$. Then $Y_{cj}$ is the “cluster” mentioned above, and corresponds to the vertex set of a connected component of $G_i(f, c)$ for the
coloring $f$ of $G_i$, extended from the colorings of $G_i$ and $G_r$ having color vectors $C(X_i)$ and $C(X_r)$, respectively. For a set $W \subseteq V$ we define a family $U(\mathcal{Y}_{lc}, \mathcal{Y}_{rc}; W)$ of vertex sets, as follows:

$$U(\mathcal{Y}_{lc}, \mathcal{Y}_{rc}; W) = \{Y_{cj} \cap W \mid 1 \leq j \leq b\}.$$

We have the following lemma.

**Lemma 3.6:** Let an internal node $X_i$ of $T$ have two children $X_l$ and $X_r$. Then, for any color vector $C_i = (Y_1, Y_2, \ldots, Y_p)$ on $X_i$,

$$\omega(X_i, C_i) = \min_{C_i', C_i''} \omega(X_i, C_i') + \omega(X_r, C_i''),$$

where the minimum is taken over all color vectors $C_i' = (Y_1, Y_2, \ldots, Y_p)$ on $X_l$ and $C_i'' = (Y_1, Y_2, \ldots, Y_p)$ on $X_r$ satisfying

$$\mathcal{Y}_c = U(\mathcal{Y}_{lc}, \mathcal{Y}_{rc}; X_i \cup \{s, t\})$$

for each color $c \in C$.\[\tag{5}\]

**Proof:** We first prove that

$$\omega(X_i, C_i) \geq \min_{C_i', C_i''} \omega(X_i, C_i') + \omega(X_r, C_i'').$$ \[\tag{6}\]

If $C_i$ is not active on $X_i$, then by Lemma 3.4 $\omega(X_i, C_i) = \infty$ and hence Eq. (6) holds. Therefore one may assume that $C_i$ is active and hence $G_i$ has a coloring $f$ with the active color vector $C_i = (Y_1, Y_2, \ldots, Y_p)$ such that

$$\omega(G_i, f) = \omega(X_i, C_i).$$

Let $f_l$ and $f_r$ be restrictions of $f$ to $E_l$ and $E_r$, respectively. Let $C_l = (Y_1, Y_2, \ldots, Y_p)$ be the active color vector of the coloring $f_l$, and let $C_r = (Y_1, Y_2, \ldots, Y_p)$ be the active color vector of the coloring $f_r$. Clearly $E_l = E_l \cup E_r$ and $E_l \cap E_r = \emptyset$. Furthermore all vertices shared by graphs $G_i$ and $G_r$ are contained in set $X_i \cap X_r \subseteq X_i$. Moreover $X_i \subseteq X_l \cup X_r$ since either $X_l = X_i$ or $X_r = X_i$. Therefore one can easily observe that

$$\mathcal{Y}_c = U(\mathcal{Y}_{lc}, \mathcal{Y}_{rc}; X_i \cup \{s, t\})$$

for each color $c \in C$, and hence $C_l$ and $C_r$ satisfy Eq. (5). We thus have

$$\omega(X_i, C_i) = \omega(G_i, f) = \omega(G_i, f_l) + \omega(G_r, f_r) \geq \omega(G_i, C_l) + \omega(G_r, C_r) \geq \min_{C_i', C_i''} \omega(X_i, C_i') + \omega(X_r, C_i''),$$ \[\tag{7}\]

completing to prove Eq. (6).

We then prove that

$$\omega(X_i, C_i) \leq \min_{C_i', C_i''} \omega(X_i, C_i') + \omega(X_r, C_i'').$$ \[\tag{8}\]

Since $E_i = E_l \cup E_r$ and $E_l \cap E_r = \emptyset$, the following extension $g$ of $g_l$ and $g_r$

$$g(e) = \begin{cases} g_l(e) & \text{if } e \in E_l, \\ g_r(e) & \text{if } e \in E_r \end{cases}$$

is a coloring of $G_i$. Since by Eq. (5) $\mathcal{Y}_c = U(\mathcal{Y}_{lc}, \mathcal{Y}_{rc}; X_i \cup \{s, t\})$ for each color $c \in C$, one can observe that $C_i = (Y_1, Y_2, \ldots, Y_p)$ is the color vector of $g$. Hence $C_i$ is an active color vector on $X_i$. Furthermore we have

$$\omega(X_i, C_i) \leq \omega(G_i, g) = \omega(G_i, g_l) + \omega(G_r, g_r) = \omega(X_i, C_l) + \omega(X_r, C_r) = \min_{C_i', C_i''} \omega(X_i, C_i') + \omega(X_r, C_i''),$$

completing to prove Eq. (8).

By Eqs. (6) and (8) we have verified Eq. (4). \[\blacksquare\]

Since $|\mathcal{Y}_{lc}|, |\mathcal{Y}_{rc}| \leq k + 4$, the bipartite graph $G_{Bc} = (\mathcal{Y}_{lc} \cup \mathcal{Y}_{rc}, E_{Bc})$ has at most $(k+4)^2$ edges, that is, $|E_{Bc}| \leq (k+4)^2$. Clearly one can check in time $O(k+4)$ whether $Y_{lc} \cap Y_{rc} \neq \emptyset$. Therefore each bipartite graph $G_{Bc}$ can be constructed in time $O((k+4)^3)$, and hence all $p$ bipartite graphs can be constructed in time $O(p(k+4)^3)$. Thus one can compute $\omega(X_i, C)$ of each active color vector on $X_i$ and $X_r$ in time $O(p(k+4)^3)$. By Eq. (2) there are at most $(k+4)^{2(k+4)p}$ active color vectors on $X_i$ and at most $(k+4)^{2(k+4)p}$ active color vectors on $X_r$. Therefore there are at most $(k+4)^{2(k+4)p}$ pairs of active color vectors on $X_i$ and $X_r$. Thus one can compute all $\omega(X_i, C)$ of active color vectors on $X_i$ in time $O(p(k+4)^{2(k+4)p}+3)$. Since $T$ has $O(n)$ internal nodes, one can compute the tables for all internal nodes in time $O(np(k+4)^{2(k+4)p}+3)$, which corresponds to the second term of the complexity in Theorem 3.1.

This completes a proof of Theorem 3.1.

**4. Polynomial-Time Algorithm**

The main result of this section is the following theorem.

**Theorem 4.1:** Let $G = (V, E)$ be a graph of $n$ vertices given by its tree-decomposition with width $\leq k$. Let $(s, t)$ be a pair of two vertices in $G$, and let $p$ be the positive integer. Then one can compute $\omega(G, p)$ in time

$$O \left( n(p+1)^{(k+1)/2} + n(p+1)^{(k+4)^2+k+8} \right).$$

If $k$ is a bounded constant, then we have the following corollary.

**Corollary 4.2:** The minimum shared-edge paths problem can be solved in polynomial time for graph with bounded treewidth.

In the remainder of this section we will give a proof of Theorem 4.1. Our idea is to reduce the size of a DP table to
$O((p+1)^{(k+4)^{k+4}})$ by considering “correct colorings within a permutation,” “counts” and “pair-counts” defined below.

Clearly the following lemma holds.

**Lemma 4.3:** Let $f$ be a coloring of $G_i = G[E_i]$ for a node $X_i$, and let $\varphi : C \rightarrow C$ be any permutation (bijection) of $C$. Then the composite $\varphi \circ f : E_i \rightarrow 2^C$ of $f$ and $\varphi$ is extensible if and only if $f$ is extensible, where $\varphi \circ f(e) = \{\varphi(c) \mid c \in f(e)\}$.

As known from Eq. (2), the number of distinct color vectors on $X_i$ is not polynomially bounded unless $p = O(\log n)$. However, the number of distinct “counts” classifying all colorings of $G_i$ is polynomially bounded even if $p = O(n)$, as follows.

We call a mapping $\gamma : F_{st}(X_i) \rightarrow \{0, 1, 2, \cdots , p\}$ a count on a node $X_i$. A count $\gamma$ on $X_i$ is defined to be active if $G_i$ has a coloring $f$ with a color vector $C(X_i) = (\gamma_1, \gamma_2, \cdots , \gamma_p)$ such that $\gamma$ satisfies

$$\gamma(A) = |\{c \in C \mid A = \gamma_c\}|$$

for each $A \in F_{st}(X_i)$. Such a count $\gamma$ is called the count of the coloring $f$. Clearly, for any active count $\gamma$,

$$\sum_{A \in F_{st}(X_i)} \gamma(A) = |C| = p.$$

We now have the following lemma.

**Lemma 4.4:** Let two colorings $f$ and $g$ of $G_i = G[E_i]$ for a node $X_i$ have the same count on $X_i$. Then $f$ is extensible if and only if $g$ is extensible.

**Proof:** It suffices to prove that if $f$ is extensible then $g$ is also extensible. Assume that $f$ is extensible. Then $f$ can be extended to a correct coloring $f^{*}$ of $G$. Since $f$ and $g$ have the same count, the following equation holds for any families $A \in F_{st}(X_i)$:

$$|\{c \in C \mid A = \gamma(X_i; f, c)\}| = |\{c \in C \mid A = \gamma(X_i; g, c)\}|.$$

Therefore there exists a permutation $\varphi : C \rightarrow C$ such that

$$\gamma(X_i; \varphi \circ f, c) = \gamma(X_i; g, c)$$

for each color $c \in C$. Let $g^{*}$ be a coloring of $G$ extended from $g$ as follows:

$$g^{*}(e) = \begin{cases} g(e) & \text{if } e \in E_i, \\ \varphi(f^{*}(e)) & \text{otherwise}. \end{cases}$$

We then claim that $g^*$ is a correct coloring of $G$ and hence $g$ is extensible. It suffices to prove that for each color $c \in C$ graph $G(g^{*}, c)$ contains the terminals $s$ and $t$, both in the same connected component of $G(g^{*}, c)$. Let $c$ be any color in $C$. Since $f^{*}$ is a correct coloring of $G$, graph $G(f^{*}, c)$ contains the terminals $s$ and $t$, both in the same connected component of $G(f^{*}, c)$. Therefore graph $G(\varphi \circ f^{*}, \varphi(c))$ contains $s$ and $t$, both in the same connected component of $G(\varphi \circ f^{*}, \varphi(c))$. The coloring $\varphi \circ f^{*}$ is the same as the coloring $g^{*}$ for the edges in $E - E_i$. Furthermore $\gamma(X_i; \varphi \circ f, c) = \gamma(X_i; g, c)$. Therefore graph $G(g^{*}, \varphi(c))$ contains $s$ and $t$, both in the same connected component of $G(g^{*}, \varphi(c))$. Thus we have proved that $g^*$ is a correct coloring of $G$.

By Lemma 4.4 an active count $\gamma$ characterizes an equivalence class of extensible colorings of $G_i$. Since $|X_i| \leq k + 1$, by Eq. (1) $|F_{st}(X_i)| \leq (k + 4)^{k+4}$. Therefore there are at most $(k + 4)^{k+4}$ distinct $A \in F_{st}(X_i)$. Thus the number $n_\gamma$ of distinct active counts $\gamma : F_{st}(X_i) \rightarrow \{0, 1, \cdots , p\}$ is at most

$$n_\gamma \leq (p + 1)^{(k+4)^{k+4}}. \tag{9}$$

The number $n_\gamma$ is bounded by a polynomial in $p$. For a count $\gamma$ on $X_i$, let

$$\omega(X_i, \gamma) = \min \{\omega(G_i, f) \mid f \text{ is a coloring of } G_i \text{ with the count } \gamma\},$$

and let $\omega(X_i, \gamma) = \infty$ if no such a coloring exists.

From the table on the root $X_0$, containing all $\omega(X_0, \gamma)$ of all active counts $\gamma$, one can easily compute $\omega(G, p)$, as follows:

$$\omega(G, p) = \min_\gamma \{\omega(X_0, \gamma)\} \tag{10}$$

where the minimum is taken over all counts $\gamma$ on the root $X_0$. By Eqs. (9) and (10), $\omega(G, p)$ can be computed in time $O((p+1)^{(k+4)^{k+4}})$. We thus need to compute a table of all $\omega(X_i, \gamma)$ on each $X_i$ by means of dynamic programming, described below.

We first compute the table of $\omega(X_i, \gamma)$ for all active counts $\gamma$ on each leaf $X_i$ of $T$. Since the number of all colorings $f : E_i \rightarrow 2^C$ of $G_i$ is $2^{p|E_i|}$, it is not polynomial in $p$. We do not need to enumerate all colorings of $G_i$ as the following lemma.

**Lemma 4.5:** Let $X_i$ be a leaf of $T$. Let $\gamma$ be a count on $X_i$. Then

$$\omega(X_i, \gamma) = \min_\xi \left\{\sum_{S \subseteq E_i} \xi(S) \geq 2\right\}, \tag{11}$$

where the minimum is taken over all mappings $\xi : 2^{E_i} \rightarrow \{0, 1, \cdots , p\}$ such that for each $A \in F_{st}(X_i)$

$$\gamma(A) = \sum_S \xi(S), \tag{12}$$

where the summation above is taken over all $S \subseteq E_i$ such that

$$A = \{V(D) \cap F_{st}(X_i) \mid D \text{ is a connected component of } G[S]\}. \tag{13}$$

**Proof:** We first prove

$$\omega(X_i, \gamma) = \min_\xi \left\{\sum_{S \subseteq E_i} \xi(S) \geq 2\right\}. \tag{14}$$
If \( \omega(X_i, \gamma) = \infty \), then Eq. (14) holds true. Therefore we may assume \( \omega(X_i, \gamma) \neq \infty \) and hence \( \gamma \) is active on \( X_i \). Then \( G_i \) has a coloring \( f \) with the count \( \gamma \). For each \( A \in \mathcal{F}_{st}(X_i) \), let
\[
C_f(A) = \{ c \in C \mid A = \mathcal{Y}(X_i; f, c) \},
\]
then
\[
\gamma(A) = |C_f(A)|.
\]
Let
\[
E_i(f, c) = \{ e \in E_i \mid f(e) \ni c \},
\]
for each color \( c \in C \) and we define
\[
C_g(S) = \{ c \in C \mid S = E_i(f, c) \}
\]
and
\[
\xi(S) = |C_g(S)|
\]
for each \( S \subseteq E_i \). Then \( \xi \) is a mapping \( 2^{E_i} \to \{0, 1, \ldots, p\} \) such that \( \xi(S) = |\{c \in C \mid S = E_i(f, c)\}| \) for each \( S \in E_i \) characterizes an equivalence class of colorings \( f \) of \( G_i \). Since \( |C| = p \) and \( |E_i| \leq k(k + 1)/2 \) for leaf \( X_i \), the number of distinct such mappings \( \xi : 2^{E_i} \to \{0, 1, \ldots, p\} \) is at most \((p + 1)^{2(k+1)/2}\) which is polynomial in \( p \). Note that \( k = O(1) \). Therefore, all \( \omega(X_i, \gamma) \) can be computed for a leaf in time \( O((p + 1)^{2(k+1)/2}) \). Since \( T \) has \( O(n) \) leaves, the tables on all leaves can be computed in time \( O(n(p + 1)^{2(k+1)/2}) \), which corresponds to the first term in the braces of the complexity mentioned in Theorem 4.1.

We now compute all \( \omega(X_i, \gamma) \) of all active counts \( \gamma \) on an internal node \( X_i \) from all active counts of its children \( X_l \) and \( X_r \). Note that \( E_l = E_i \setminus E_r \) and \( E_l \cap E_r = \emptyset \). We call a mapping \( \rho : \mathcal{F}_{st}(X_l) \times \mathcal{F}_{st}(X_r) \to \{0, 1, 2, \ldots, p\} \) a pair-count on \( X_i \). We define a pair-count \( \rho \) to be active if \( G_i \) has a coloring \( f \) such that, for each pair of \( A_l \in \mathcal{F}_{st}(X_l) \) and \( A_r \in \mathcal{F}_{st}(X_r) \)
\[
\rho(A_l, A_r) = \{ |c \in C \mid A_l = \mathcal{Y}(X_l; f_l, c), A_r = \mathcal{Y}(X_r; f_r, c) \},
\]
where \( f_l = f|G_l \) is the restriction of \( f \) to \( E_l \) and \( f_r = f|G_r \) is the restriction of \( f \) to \( E_r \). Such a pair-count \( \rho \) is called the pair-count of the coloring \( f \) of \( G_i \). Let
\[
\omega_{pair}(X_i, \rho) = \inf \{ \omega(G_i, f) \mid f \text{ is a coloring of } G_i \text{ with the pair-count } \rho \}
\]
and let \( \omega_{pair}(X_i, \rho) = \infty \) if no such a coloring exists. Then we have the following lemma.

**Lemma 4.6:** Let an internal node \( X_i \) of \( T \) have two children \( X_l \) and \( X_r \), and let \( \rho \) be any pair-count on \( X_i \). Then
\[
\omega_{pair}(X_i, \rho) = \min_{\gamma_l, \gamma_r} \omega(X_i, \gamma_l) + \omega(X_r, \gamma_r),
\]
where the minimum is taken over all pairs of active counts \( \gamma_l \) on \( X_i \) and \( \gamma_r \) on \( X_r \) satisfying
\[
\begin{align*}
\gamma_l(A_l) &= \sum_{A \in \mathcal{F}_{st}(X_i)} \rho(A_l, A) \quad \text{for each } A_l \in \mathcal{F}_{st}(X_i); \text{ and} \\
\gamma_r(A_r) &= \sum_{A \in \mathcal{F}_{st}(X_i)} \rho(A, A_r) \quad \text{for each } A_r \in \mathcal{F}_{st}(X_r).
\end{align*}
\]
Using Lemma 4.6, we compute all \( \omega_{pair}(X_i, \rho) \) of all active pair-counts \( \rho \) on \( X_i \) from all pairs of active counts \( \gamma_l \) on \( X_i \) and \( \gamma_r \) on \( X_r \). Since there are at most \((k + 4)^{2k+8}\) such pairs, the complexity of computing all \( \omega_{pair}(X_i, \rho) \) is
\[
O((k + 4)^{2k+8}).
\]
pairs \((A_t, A_r)\) for which \(\rho(A_t, A_r) \geq 1\), there are at most \(p+1\)(k+4)\(2k+8\) possible distinct active counts \(\rho\). For each \(\rho\) of them, we check in time \(O((k+4)^{k+4}) = O(1)\) whether \(\rho\) satisfies Conditions (B1) and (B2) in Lemma 4.6. Checking Conditions (B1) and (B2) for all possible \(\rho\)'s can be done in time \(O((p+1)^{2(k+4)^{k+4}})\). Thus we have shown that all active pair-counts \(\rho\) and \(\omega_{\text{pair}}(X_i, \rho)\) on \(X_i\) can be computed in time

\[
O((p+1)(k+4)^{k+4}) = O(1)
\]

We now show how to compute all \(\omega(X_i, \gamma)\) of all active counts \(\gamma\) on an internal node \(X_i\) from all active pair-counts on \(X_i\) as in the following lemma.

**Lemma 4.7:** Let an internal node \(X_i\) of \(T\) have two children \(X_l\) and \(X_r\), and let \(\gamma\) be any count on \(X_i\). Then

\[
\omega(X_i, \gamma) = \min_{\rho'}\{\omega_{\text{pair}}(X_i, \rho')\},
\]

where the minimum is taken over all active pair-counts \(\rho'\) on \(X_i\) such that for each pair \(A \in F_{st}(X_i)\)

\[
\gamma(A) = \sum_{\rho' \in U} \rho(A_l, A_r)
\]

where the summation above is taken over all pairs of \(A_t \in F_{st}(X_l)\) and \(A_r \in F_{st}(X_r)\) satisfying

\[
A = U(A_l, A_r, X_i),
\]

where \(U\) has been defined in the previous section.

Using Lemma 4.7, we compute all \(\omega(X_i, \gamma)\) of all active counts \(\gamma\) on \(X_i\) from all active pair-counts \(\rho\) on \(X_i\). There are at most \(p+1\)(k+4)\(2k+8\) distinct active pair-counts \(\rho\). From each \(\rho\) of them, we compute \(\gamma\) satisfying Eq. (24) in time \(O(1)\). Since \(|A_t| \leq k + 4\), the bipartite graph \(G_{BC} = (A_t \cup A_r, E_c)\) defined in the previous section contains at most \((k+4)^2 = O(1)\) edges. Therefore one can check in time \(O(1)\) for \(A_t\) and \(A_r\), whether \(A = U(A_t, A_r, X_i)\), and hence one can check Eq. (25) in time \(O(1)\). Thus one can compute all \(\omega(X_i, \gamma)\) of all active counts \(\gamma\) on \(X_i\) in time

\[
O((p+1)(k+4)^{2k+8})
\]

Since \(T\) has \(O(n)\) internal nodes, one can compute the tables for all internal nodes in time \(O(n(p+1)(p+1)^{2k+8})\), which corresponds to the second term in the braces of the complexity mentioned in Theorem 4.1.

This completes a proof of Theorem 4.1.

5. Conclusion

In this paper we gave a polynomial-time algorithm for the minimum shared-edge paths problem on graphs with bounded treewidth. Our algorithms can be extended to more than one terminal pair as follows. Let \((s_1, t_1), (s_2, t_2), \ldots, (s_\alpha, t_\alpha)\) be \(\alpha\) pairs of two terminals in \(G\), and for each \(i, 1 \leq i \leq \alpha\), let \(p_i\) be a positive integer. Then the problem is to find \(\sum_{1 \leq i \leq \alpha} p_i\) paths such that there are \(p_i\) paths connecting \(s_i\) and \(t_i\) for each \(i, 1 \leq i \leq \alpha\), so as to minimize the number of edges shared among the paths. If \(\alpha\) is bounded, our algorithm can be extended to solve the problem in polynomial time for graphs with bounded treewidth.

Acknowledgments

This work is partially supported by the Japan Society for the Promotion of Science (JSPS), Grant-in-Aid for Scientific Research, Grant Numbers 23500001.

References


