Faster Algorithms for Subgraph Isomorphism of k-connected Partial k-trees

Anders Dessmark — Lund University

Andrzej Lingas — Lund University

Andrzej Proskurowski — University of Oregon

Abstract

The problem of determining whether a k-connected partial k-tree is isomorphic to subgraph of another partial k-tree is shown to be solvable in time $O(n^{k+2})$. The presented time-bounds considerably improve the corresponding bounds known in the literature. They rely in part on a new characterization of width-k tree-decomposition of k-connected partial k-trees.

1 Introduction

The *subgraph isomorphism* problem is to determine whether a graph is isomorphic to a subgraph of another graph. It is a fundamental graph problem with a variety of applications in, for instance, engineering sciences, organic chemistry, biology, and pattern matching.

For instance, if G is an n-vertex cycle and H is an n-vertex planar graph with each vertex of degree 3, then determining whether G is isomorphic to a subgraph of H is equivalent to the NP-complete problem of determining whether a 3-regular planar graph has a Hamiltonian circuit [6]. Thus, the subgraph isomorphism problem is NP-complete even if G and H range only over connected planar graphs with maximum vertex degree at most 3. Subgraph isomorphism also remains NP-complete when the first input graph, the guest G, is a forest and the other input graph, the host H, is a tree (see page. 105 in [6]).

Roughly speaking, the only known polynomial-time algorithms for subgraph isomorphism are those for k-connected partial k-trees and partial k-trees of bounded degree. Weakening any of the two assumptions yields NP-completeness [16, 8]. Besides general polynomial-time algorithms for the two problems [3, 16, 7], there are several specific polynomial-time algorithms for special cases of these problems, e.g., for trees [15], two-connected outerplanar graphs [12], two-connected series-parallel graphs [14], and k-connected partial k-paths [8]. In [16], Matoušek and Thomas have derived $O(n^{k+4.5})$ and $O(n^{k+2})$ bounds on the time complexity of the subgraph isomorphism problem for k-connected partial k-trees and for bounded degree partial k-trees, respectively. In [7], Gupta and Nishimura confirmed both bounds by using a different method based on width-k tree-decomposition of partial k-trees.

Partial k-trees are known for their good algorithmic properties (see [2]), especially for problems that can be described in the language of extended Monadic Second Order Logic, [1]. Since graph isomorphism is not one of these problems, it is doubtful that the general techniques of [1] can be used to construct an efficient subgraph isomorphism algorithm for the feasible cases of partial k-trees.

In this paper, first we show how to improve the efficiency of the method for k-connected partial k-trees presented in [7] to $O(n^{k+2})$. As in [7], only the guest graph G is required to be k-connected in our method. Our improvement has been possible due to three factors. One follows from our theorem on an invariant of k-tree embedding of a k-connected partial k-tree: if such a graph, G, has a minimal separator, S, yielding at least $\max\{k, 3\}$ connected components, then S induces a clique in every k-tree embedding of G. In consequence, we obtain a linear upper bound on the total number of connected components resulting from the removal of such separators, one at a time. This non-trivial combinatorial fact enables us to decrease the time taken by the embedding subproblems induced by multi-component separators in the method from [7].

Another source of improvement is adopting the technique of jointly solving groups of related matching problems associated to the subproblems (originally applied by Matula to subtree isomorphism [15]). The third source is a more refined analysis. All this contributes to the development of an $O(n^{k+2})$ subgraph isomorphism algorithm for k-connected partial k-trees*.

The structure of our paper is as follows. In Section 2, we introduce some basic graph theoretic concepts and define the bounded-width tree decomposition of a graph and its so called normalized version. In Section 3, we derive our main results on the presence of multi-component separators of k-connected partial trees in their width-k tree decompositions, and their combinatorial consequences. In Section 4, we provide a rough sketch of the method for subgraph isomorphism for k-connected partial k-trees from [7]. In Section 5, we present improved algorithms and analysis for this problem. Finally we discuss various possible extensions and improvements of our results.

*For $k = 2$, and $k = 3$, the stronger time-bound of $O(n^{3.5})$ can be obtained in the case when both graphs are k-connected [5].

2 Preliminaries

We shall adhere to standard notations for undirected graphs as in, for instance, [9].

In particular, a graph is *isomorphic* to another graph if there is a one-to-one correspondence between vertices of the two graphs that preserves adjacency. Such a correspondence is called an *isomorphism* between the two graphs.

For a graph G, we shall denote the set of vertices of G by $V(G)$ and the set of edges of G by $E(G)$. The degree of a vertex v of G will be denoted by $\deg_G(v)$.

For a subset of vertices $S \subset V(G)$, the graph resulting from removing S with all incident edges in G will be denoted by $G \setminus S$. Such a subset will be called a *separator* of G, if $G \setminus S$ has at least two connected components. For a positive integer constant $k \geq 1$, G is *k-connected* if $|V(G)| > k+1$ and each separator of G has at least k vertices. A completely connected set of k vertices (one that induces a subgraph isomorphic to the complete graph K_k) is called a *k-clique*.

A *k-tree* is a graph which can be reduced to a k-clique by a sequence of eliminations of degree k vertices whose neighbors induce a k-clique. A *partial k-tree* is a subgraph of a k-tree. Following Gupta and Nishimura's presentation, we will refer to vertices of partial k-trees as "nodes".

Following [4], a *tree-decomposition* of a graph G is a pair (T, χ) where T is a tree and χ is a mapping from $V(T)$ to the set of subsets of $V(G)$ satisfying:

1. for each $(u, v) \in E(G)$ there is an $x \in V(T)$ such that $\{u,\ v\} \subset \chi(x)$;

2. for any $x,\ y,\ z \in V(T)$, if y is on the path from x to z in T then $\chi(x) \cap \chi(z) \subset \chi(y)$.

For $v \in V(T)$, the subset $\chi(v)$ of $V(G)$ is often identified with v and called a *bag*. The maximum size of a bag in (T, χ), minus one is called the *width* of (T, χ). The *treewidth* of a graph is the minimum width of its tree-decomposition. The following fact is well known (see, for instance, [11].)

Lemma 2.1 *The class of partial k-trees is exactly the class of graphs of treewidth at most k.*

It is easy to observe that the intersection of any two bags adjacent in a minimal tree-decomposition of a graph G is a separator of G. For the purpose of solving the subgraph isomorphism problem for partial k-trees, Gupta and Nishimura have introduced the so called *normalized tree-decomposition* where such separators explicitly occur as bags (separator vertices) [7].

A tree-decomposition (T, χ) of a graph H is a normalized tree-decomposition of H if $V(T)$ can be partitioned into the set of *separator vertices* and the set of *clique vertices* and T can be rooted such that the following conditions hold. The root of T is a separator vertex, the leaves of T are clique vertices, the child of each separator vertex is a clique vertex and *vice versa*. Furthermore, for each child y of a separator vertex x, $\chi(x) \subset \chi(y)$, and in turn for each child z of y, $\chi(z) \subset \chi(y)$ and, for two clique vertices u and v with a common parent, the symmetric difference of $\chi(u)$ and $\chi(v)$ has exactly two elements. Finally, for any two separator vertices $x,\ y$ adjacent to the same clique vertex in T, $\chi(x),\ \chi(y)$ are required to be different to ensure the linear size of T.

Lemma 2.2 *([7]) Let H be a partial k-tree on n nodes. A normalized tree-decomposition for H of width k can be constructed in time $O(n^2)$.*

3 Tree-decompositions of k-connected partial k-trees

The general setup of this section is a given k-connected partial k-tree G with a separator $S = \{v_1, \cdots, v_k\}$ such that $G \setminus S$ has at least k connected components, $C_1, \cdots, C_q$ where $q \geq k$. Additionally, we assume that C_1 has at least two nodes, $x_{1,1}$ and $x_{1,2}$. We will consider an arbitrary width-k tree-decomposition (T, χ) of G and show that there is a bag of T that includes S.

For a tree-decomposition (T, χ) of G, and a node v of G, denote by T_v the subtree of T induced by bags containing v. For a subset R of nodes of G, let T_R denote the union of T_v, over all nodes $v \in R$. The following fact can be established readily.

Fact 1 For any clique in a graph G, there is a bag in its tree-decomposition (T, χ) that contains all the nodes of the clique.

Also the following fact is a direct consequence of the definition of tree-decomposition.

Fact 2. For any connected subgraph G' of G, $T_{V(G')}$ is a connected subtree of T.

Claim 1. Let v_i and v_j be two nodes of a separator S of G. If the two subtrees T_{v_i} and T_{v_j} of T have empty intersection, then the bags of T on the path between T_{v_i} and T_{v_j} contain at least one node from each of the connected components $C_l, 1 \leq l \leq q$.

Proof: If an element of C_l is absent from a bag on the path, then $T_{v_i} \cap T_{C_l} = \emptyset$ or $T_{v_j} \cap T_{C_l} = \emptyset$ by Fact 2. Consequently, $S \setminus \{v_j\}$ or $S \setminus \{v_i\}$ is a separator of G with $k-1$ nodes. $\square$

Claim 2. If there is no bag in T that contains a given pair of nodes, v_i and v_j of the separator S, then no bag in T contains any pair of nodes in S.

Proof: Consider a third node v_l in S. If the corresponding subtree of T, T_{v_l}, has a non-empty intersection with, say, T_{v_i}, then there is a bag with $q+2$ vertices, namely, q nodes from the connected components (by Claim 1) and the two nodes from S. By symmetry, there is no pair of nodes in S sharing a bag. $\square$

Claim 3. If for every pair of nodes $v_i,\ v_j$ in the separator S there is a bag of T containing v_i and v_j then there is a bag of T containing S.

Proof: The tree T with the same mapping χ remains a tree-decomposition when the graph G is augmented by all edges (v, w) (not in $E(G)$) such that there is a bag of T including $\{v,\ w\}$. Now, it is sufficient to observe that S forms a clique in the resulting partial k-tree and use Fact 1 to ascertain the existence of a bag of T that includes all nodes of S. $\square$

For a set $\mathcal{T}$ of subtrees of T, let us call a subtree T_x *extreme* if it has empty intersection with the interior of any path in T between two subtrees of the set $\mathcal{T}$.

We are now ready to state and prove the basic theorem.

Theorem 3.1 *Let $k \geq 3$, and let $S = \{v_1, \cdots, v_k\}$ be a separator in a k-connected partial k-tree G such that $G \setminus S$ has at least k connected components, $C_1, \cdots, C_q$ where $q \geq k$. Then, if C_1 has at least two nodes, then S is in a bag of every width-k tree-decomposition of G. (Alternatively, S induces a k-clique in every k-tree embedding of G).*

Proof: By Claim 3, it is sufficient to show that for every pair of nodes in S there is a bag of T containing both nodes. The proof is by contradiction with the assumption about either the k-connectivity of G, or the width of the tree-decomposition T.

Let us thus assume that there are two nodes in S that do not appear together in any bag of T. By Claim 2, all subtrees T_{v_i} are then pairwise disjoint in T. Let T_{v_1} and T_{v_k} be extreme in T and let X_1 and X_k be their closest bags (such that v_1 and v_k are in no other bag on the path between them in T). By Claim 1, we have $C_1 \cap X_1 \neq \emptyset$ and $C_1 \cap X_k \neq \emptyset$. We may assume without loss of generality that $x_{1,1} \in C_1 \cap X_1$. Observe that $x_{1,1}$ cannot also be the element of C_1 in X_k: if that were the case, the other nodes of C_1 would be separated from other components by fewer than k nodes. Thus, there is a bag X_l on the path between X_1 and X_k in T with both $x_{1,1}$ and $x_{1,2}$. X_l contains no separator nodes, since Claim 1 would then imply $q+2$ nodes in that bag. Depending on the location of X_l with respect to the other subtrees T_{v_j}, either $x_{1,1}$ is separated from the other components $C_i, 1 < i \leq q$, by $v_1, x_{1,2}$ and at most $k-3$ other separator nodes, or $x_{1,2}$ is separated from the other connected components of G by $v_k, x_{1,1}$ and at most $k-3$ other separator nodes, contradicting the connectivity assumption.

□

The conditions of the theorem are fairly tight. For a k-connected partial k-tree with a separator whose removal gives only trivial (one vertex) k connected components, a counterexample is provided by the complete bipartite graph $K_{k,k}$, where there are two symmetric tree-decompositions. For a separator yielding fewer than k connected components, counter-examples can be obtained for relatively small sizes of the components, also based on $K_{k,k}$.

Theorem 3.1 can be refined to the following statement in terms of normalized tree-decomposition.

Theorem 3.2 *Let $k > 2$ and let $S = \{v_1, \cdots, v_k\}$ be a separator in a k-connected partial k-tree G such that $G \setminus S$ has at least k connected components, $C_1, \cdots, C_q$ where $q \geq k$. Then, if C_1 has at least two elements, S forms a bag corresponding to a separator vertex in every width-k normalized tree-decomposition of G.*

Proof: By the definition of normalized tree-decomposition (T, χ), if there are two clique vertices x, y such that $S \subset \chi(x) \cap \chi(y)$ then the thesis holds. By Theorem 3.1, we know that there is at least one such clique vertex x. Suppose that for no other clique vertex y, $\chi(y)$ includes S. Let v be the node in $\chi(x) \setminus S$, and let C be the component in $G \setminus S$ not containing v. There has to be a separator vertex z in T_C adjacent to x in the tree-decomposition T and a node $s \in S$ outside $\chi(z)$. It follows that s doesn't occur in any bag containing nodes from C which contradicts the fact that s is adjacent to at least one node in C by the k-connectivity property. $\square$

For $k = 2$, we can analogously obtain a slightly weaker version of Theorem 3.2 requiring at least $k+1$, i.e., 3 components. This version also follows from Lemma 7 in [17].

Theorem 3.3 *Let $S = \{v_1, \cdots, v_k\}$ be a separator in a 2-connected partial 2-tree G such that $G \setminus S$ has at least 3 connected components, $C_1, \cdots, C_q$ where $q \geq 3$. Then S forms a bag corresponding to a separator vertex in every width-2 normalized tree-decomposition of G.*

4 Sketch of the algorithm of Gupta and Nishimura

The algorithm of Gupta and Nishimura starts by finding a width-k normalized tree-decomposition $TD(H)$ of the host partial k-tree H and the so called tree-decomposition graph $TDG(G)$ of the guest k-connected partial k-tree G [7]. It uses $TDG(G)$ to determine the existence of a width-k normalized tree-decomposition of G similar to a part of $TD(H)$.

A vertex in the directed acyclic graph $TDG(G)$ corresponds to a potential vertex in a width-k normalized tree-decomposition of G. Thus, there are two kinds of vertices, separator vertices and clique vertices, similarly as in $TD(H)$.

A separator vertex v in $TDG(G)$ is a triple (S_v, δ_v, P_v) where S_v is a k node separator of G, δ_v is a permutation of nodes in S_v and P_v is a distinguished connected component of $G \setminus S$, so called parent component [†]. (The permutation δ_v is used in the algorithm from [7] to prespecify a potential embedding of a subgraph of G induced by v into the subgraph of H induced by a similar vertex of $TD(G)$ on the subset S_v.)

Similarly, a clique vertex w is a triple (O_w, γ_w, R_w) where O_w is a $(k+1)$-element subset of $V(G)$, γ_w is a permutation of O_w and R_w is a sequence R^0_w, R^1_w, ...R^k_w of subsets of $V(G)$ satisfying the following properties. Let $O_w = (o_1, ..., o_{k+1})$ under the permutation γ_w. R^0_w is the set of all connected

[†] We simplify here somewhat the original definitions from [7].

components of $G \setminus (O_w \setminus \{o_{k+1}\})$ not containing o_{k+1}, and for $i = 1, .., k$, R_w^i is the set of all connected components of $G \setminus (O_w \setminus \{o_i\})$ not containing o_i.

The edges in $TDG(G)$ correspond to potential edges in a width-k normalized tree decomposition of G. To keep the representation of $TDG(G)$ linear in its number of vertices, Gupta and Nishimura group children of a vertex into clusters. In result, for each cluster only one edge directed from the parent to a minimum number child in the cluster is explicitly given.

A clique vertex w is in the j-th cluster of a separator vertex v if $S_v = (o_1, ..., o_k)$ under the permutation δ_v, $O_w = S_v \cup \{o_{k+1}\}$, o_{k+1} is in the j-th non-parent connected component C_j of $G \setminus S_v$, γ_w is an extension of δ_v, R_w^0 is the union of all connected components of $G \setminus S_v$ different from C_j and C_j is the union of R_w^1 through R_w^k augmented by $\{o_{k+1}\}$.

Similarly, a separator vertex u is in the j-th cluster of a clique vertex w iff $O_w = (o_1, ..., o_{k+1})$ under the permutation γ_w, $S_u = (o_1, ...o_{j-1}, o_{j+1}...o_{k+1})$ under the permutation δ_u, the parent component P_u of u is the union of R_w^i where $i \neq j$, and R_w^j is the union of all non-parent connected components of $G \setminus S_u$.

After computing a width-k normalized tree-decomposition $TD(H) = (T, \chi)$ and the tree-decomposition graph $TDG(H)$ of G, the algorithm of Gupta and Nishimura searches for a tree-decomposition of G similar to a subtree of $TD(H)$. The search proceeds by solving constrained subgraph isomorphism subproblems respectively induced by pairs of separator vertices and pairs of clique vertices from $TD(H)$ and $TDG(G)$.

Consider a separator vertex w of $TDG(G)$ and a separator vertex v of $TD(H)$. Let $G_{(w)}$ be the subgraph of G resulting from removing the parent component P_w from G. Next, let $H_{(v)}$ be the subgraph of H induced by the union of the bags of the complete subtree of T rooted at v. *The subgraph* *isomorphism subproblem induced by w, v* consists in determining whether there is an isomorphism f between $G_{(w)}$ and a subgraph of $H_{(v)}$ satisfying the following conditions.

1. f maps S_w onto $\chi(v)$ such that the i-th node of $\chi(v)$ is the image of the ${\delta_w}^{-1}(i)$ node of S_w.

2. Each child cluster of w contains a representative clique vertex u for which there is a unique child y of v such that f restricted to $G_{(u)}$ solves the subproblem induced by u and y.

The definition of the subproblem induced by a pair of clique vertices is analogous.

Following the above definitions, the subgraph isomorphism subproblems are solved by reduction to the subproblems for child vertices already solved using a matching technique [7, 10]. The maximum maximum cardinality matching in a bipartite graph with girls corresponding to the so called clusters of w and boys corresponding to the child clique vertices of v is computed. There is an edge between the i-th cluster and the j-th child of v iff there exists a subgraph isomorphism between a subgraph of G induced by a clique vertex in the i-th cluster and the subgraph of H induced by the j-th child of v. If the cardinality of the maximum matching equals the number of child clusters of w then the subgraph isomorphism subproblem has a solution‡. Ultimately, there exists an isomorphism between G and a subgraph of H if there is a vertex w in $TDG(G)$ with empty parent component and a vertex in $TD(H)$ such that the subproblem induced by w and v has a solution.

‡Additionally, the algorithm from [7] passes some information from grandchildren to grandparents which seems superfluous by the properties of normalized tree decompositions.

5 Subgraph isomorphism for k-connected partial k-trees

The following combinatorial consequence of Theorems 3.2, and 3.3 will be useful in the analysis of our algorithm for subgraph isomorphism for k-connected partial k-trees based on that due to Gupta and Nishimura [7].

Theorem 5.1 *Let $k \geq 2$, $k' = \max\{k, 3\}$ and let G be a k-connected partial k-tree on n nodes. The sum of the number of connected components induced by separators S on k nodes disconnecting G into at least k' components such that at least one of them is non-trivial is $O(n)$.*

Proof: Consider any width-k normalized tree-decomposition (T, χ) of G. Clearly, T has $O(n)$ vertices. By Theorems 3.2, and 3.3, each of the separators S corresponds to a separator vertex in T. The degree of such a vertex in T is clearly not smaller than the number of connected components induced by S by the k-connectivity of G (see p. 176 in [7]). □

The following lemmata will be useful in our refined time analysis.

Lemma 5.2 *Let S be a separator of G on k nodes disconnecting G into l connected components. The total time for reducing the subgraph isomorphism subproblems induced by vertices in $TDG(G)$ having S on their first coordinate is $O(n^2 + l^{1.5}n)$.*

Proof: First, consider such a single subgraph isomorphism subproblem respectively induced by a vertex w in $TDG(G)$ with S on its first coordinate and a separator vertex v in $TD(H)$. It can be solved by finding a maximum cardinality matching in a bipartite graph as explained in the previous section. Since the clusters of w are disjoint they contain $O(n)$ vertices in all. Hence, the construction of the bipartite graph can be accomplished in time $O(n \deg_{TD(H)}(v))$ on the basis of subproblems already determined. Since the number of clusters of w is equal to $l-1$, a maximum cardinality matching in the bipartite graph can be found in time $O(l^{1.5} \deg_{TD(H)}(v))$.

Let δ be the permutation of S which is the second coordinate of w. Consider the set W_δ of all vertices in $TDG(G)$ that have on the first coordinate S and on the second coordinate δ. They differ from each other by the choice of the parent component and consequently by one cluster. Their number is clearly l. Instead of solving the subproblems induced by vertices in W_δ and v separately, we can solve them together by adopting the so called technique of rooted bipartite matching from [15]. So, we extend the bipartite graph constructed for w and v by a girl cluster node corresponding to the "parent cluster" of w and a root boy node adjacent to all girl nodes. Suppose that the extended bipartite graph has a matching incident to all girl nodes. Then, it is easy to see that the subproblem induced by the vertex in W_δ, whose "parent cluster" is matched with the root boy node, and v has a positive solution. Now, it is sufficient to find all other maximum matchings of the extended bipartite graph by alternating breadth-first search on the edges in the matching and outside the matching starting from the root boy node [15]. It takes $O(l \deg_{TD(H)}(v))$ time. Thus, we can solve all the subproblems induced by W_δ and v in time $O(n \deg_{TD(H)}(v) + l^{1.5} \deg_{TD(H)}(v))$.

Since the number of permutations of S is $O(1)$, we conclude that all the subgraph isomorphism subproblems induced by the vertices in $TGD(G)$ with S on their first coordinate take time $O(n \deg_{TD(H)}(v) + l^{1.5} \deg_{TD(H)}(v))$. Now, by taking the sum over all separator vertices v in $TD(H)$ and, because of the linear size of $TD(H)$, we obtain the thesis. $\square$

A clique vertex in $TDG(G)$ has $O(k)$ singleton clusters by the definition of $TDG(G)$ [7]. Hence, we analogously obtain the following simpler lemma for clique vertices.

Lemma 5.3 *Let C be a $(k+1)$-element subset of $V(G)$. The subproblems induced by vertices in $TDG(G)$ having C on their first coordinate can be reduced to the subproblems already solved in time $O(n)$.*

The following lemma will also be useful in limiting the time taken by the subproblems induced by clique vertices.

Lemma 5.4 *For a k-connected partial k-tree, the number of $(k+1)$-element subsets of $V(G)$ that occur on the first coordinates of clique vertices in $TGD(G)$ can be pruned to $O(n^k)$.*

Proof: Consider any width-k normalized tree-decomposition (T, χ) of G.

By the k-connectivity, any leaf clique vertex in the tree-decomposition has to contain at least k edges.

Consider a non-leaf clique vertex w and its parent separator vertex v in T. Let b be the node in $\chi(w) \setminus \chi(v)$. Suppose that there is no edge in $E(G)$ connecting b with the nodes in $\chi(v)$. Since w is not a leaf vertex, it has a child separator vertex v' which in turn has a child clique vertex w'. Clearly, there is a node a in $\chi(v) \setminus \chi(w')$ such that $\chi(v') = \chi(v) \cup \{b\} \setminus \{a\}$. There is also a node c in $\chi(w') \setminus \chi(v)$ different from b. We obtain a contradiction since a can be disconnected from c by removing the $k-1$ nodes in $\chi(v') \setminus \{b\}$.

G as a partial k-tree has $m < nk$ edges. Hence, the number of $(k+1)$-element subsets of $V(G)$ that could occur on the first coordinate of a clique vertex in any width-k normalized tree-decomposition of G is $< mn^{k+1-2} = O(n^k)$. $\square$

Now, we are ready to derive our main result in this section.

Theorem 5.5 *Let $k \geq 2$. The subgraph isomorphism problem for a k-connected partial k-tree G and a partial k-tree H is solvable in time $O(n^{k+2})$.*

Proof: Clearly, $TD(H)$ and $TDG(G)$ can be constructed within the hypothesized time bound [7]. We may assume without loss of generality that G has more than $2k+2$ nodes since otherwise the algorithm of Gupta and Nishimura runs in linear time. It follows that any k-separator of G inducing at least $k+1$ connected components has at least one non-trivial connected component or else there can be only one k-separator in G, $TDG(G)$ consequently being of linear size. Hence, by Theorem 5.1, the sum of the numbers l of connected components induced by the aforementioned separators in $TDG(G)$ is $O(n)$. By Lemma 5.2, we conclude that all subgraph isomorphism subproblems for separator vertices in $TDG(G)$ having separators inducing at least $k+1$ components on their first component can be solved in time $O(n^{2.5})$. There are $O(n^k)$ remaining separators on k nodes occurring on the first coordinates of vertices in $TDG(G)$. The subproblems induced by the vertices whose first coordinates belong to the remaining separators can be solved in time $O(n^{k+2})$ by Lemma 5.2. By Lemma 5.4, we can prune $TDG(G)$ so only $O(n^k)$ $(k+1)$-element subsets occur on the first coordinates of clique vertices in $TDG(G)$. Consequently, all the subproblems induced by pairs of clique vertices from $TD(H)$ and $TDG(G)$ can be solved in time

$O(n^{k+1})$ by Lemma 5.3. $\square$

6 Final remarks

Our algorithms for subgraph isomorphism for k-connected partial k-trees can be also extended to produce, if possible, a subgraph of H isomorphic to G by retracing appropriate bipartite matchings in top-down fashion (cf. [15]).

For $k = 2$, and $k = 3$, it can also be proved that any two width-k, normalized tree decompositions of a k-connected partial k-tree restricted to multi-separator vertices are isomorphic. In effect, we can more closely follow the idea of Matula's algorithm for subtree isomorphism. By combining it with some ideas of Gupta-Nishimura's algorithm for subgraph isomorphism for k-connected partial k-paths [8], we can derive even faster algorithms for subgraph isomorphism when both graphs are either 2-connected partial 2-trees or 3-connected partial 3-trees, in both cases running in time $O(n^{3.5})$ [5].

A graph G can be *topologically embedded* in a graph H if H contains a subgraph which after contracting some paths with internal nodes of degree two is isomorphic to G. Gupta and Nishimura extended their algorithm for subgraph isomorphism of k-connected partial k-trees to include topological embedding in [7]. Similarly, our algorithms can probably be extended to include topological embedding, improving their time-bound for topological embedding of k-connected partial k-trees.

Gupta and Nishimura also provided randomized NC algorithms for subgraph isomorphism and topological embedding of k-connected partial k-trees in [7]. By using our approach, it's likely that their processor bounds can also be improved.

It is easy to observe that the time complexity of the maximum bipartite matching problem is a lower bound on the time complexity of the subtree isomorphism problem within a logarithmic factor (cf. [13]). In turn, for any positive integer constant k, the time complexity of the subtree isomorphism problem is a trivial lower bound on the time complexity of the subgraph isomorphism problem for k-connected partial k-trees (e.g., consider complete k-trees). It is an interesting open problem whether it is possible to approach the lower bound given by maximum bipartite matching more closely for small k than our upper bounds do? Another interesting problem is whether there exists a polynomial upper bound on the time complexity of subgraph isomorphism for k-connected partial k-trees where the degree of the bounding polynomial is not dependent on k? Such a uniform (cubic) upper bound is known only for k-connected partial k-paths which are a very restricted case of k-connected partial k-trees [8].

Acknowledgments

Thanks go to Srini Arikati for valuable comments on the method of Gupta and Nishimura [7].

References

[1] S. Arnborg, J. Lagergren and D. Seese. Problems easy for tree-decomposable graphs. Journal of Algorithms, 12 (1991), pp. 308-340.

[2] S. Arnborg and A. Proskurowski. Linear time algorithms for NP-hard problems on graphs embedded in k-trees. Discrete Applied Mathematics 23 (1989), pp. 11-24.

[3] H. Bodlaender. Dynamic programming on graphs with bounded tree-width. Proc. ICALP'88, LNCS 317, pp. 105-118, Springer Verlag.

[4] H. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. Proc. 25rd ACM STOC, pp. 226-234.

[5] A. Dessmark, A. Lingas and A. Proskurowski. Faster Algorithms for Subgraph Isomorphism of k-connected Partial k-trees (preliminary version). Proc. ESA'96, LNCS 1136, pp. 501-513.

[6] M.R. Garey and D.S. Johnson. Computers and Intractability. A Guide to the Theory of NP-completeness. Freeman, San Francisco, 1979.

[7] A. Gupta and N. Nishimura. Sequential and Parallel Algorithms for Embedding Problems on Classes of Partial k-Trees. Proc. SWAT'94, LNCS 824, pp. 172-182, Springer Verlag.

[8] A. Gupta and N. Nishimura. Characterizing the Complexity of Subgraph Isomorphism for Graphs of Bounded Path-Width. Proc. STACS'96, LNCS 1046, pp. 453-464.

[9] F. Harary. Graph Theory. Addison-Wesley, Reading, Massachusetts, 1979.

[10] J. Hopcroft and R. Karp. An $n^{5/2}$ algorithm for maximum matching in bipartite graphs. SIAM J.Comput. 2:4 (1973), pp. 225-231.

[11] J. van Leeuwen. Graph Algorithms. Handbook of Theoretical Computer Science A, North Holland, Amsterdam 1990, pp. 527-631.

[12] A. Lingas. Subgraph isomorphism for biconnected outerplanar graphs in cubic time. Theoretical Computer Science 68 (1989), pp. 295-302.

[13] A. Lingas and M. Karpinski. Subtree Isomorphism and Bipartite Perfect Matching are Mutually NC Reducible. Tech. Rept. 856-CS, December 1986, Bonn University.

[14] A. Lingas and M. Sysło. A Polynomial Algorithm for Subgraph Isomorphism of Two-connected Series-Parallel Graphs. Proc. ICALP'88, LNCS 317, pp. 394-409, Springer Verlag.

[15] D. Matula. Subtree isomorphism in $O(n^{5/2})$. Annals of Discrete Mathematics 2 (1978), pp. 91-106.

[16] J. Matoušek and R. Thomas. On the complexity of finding iso- and other morphisms for partial k-trees. Discrete Mathematics 108 (1992), pp. 343-364.

[17] A. Proskurowski, M. Sysło and P. Winter. Plane embeddings of 2-trees and biconnected partial 2-trees. SIAM J. on Discrete Mathematics 9, 4(1996), 277-296.