Discrete Mathematics 108 (1992) 343–364
North-Holland

On the complexity of finding iso- and other morphisms for partial k-trees

Jiří Matoušek

Department of Computer Science, Charles University, Malostranské nám. 25, 11800 Praha 1, Czechoslovakia

Robin Thomas*

School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332, USA

Received 4 January 1991

Abstract

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

The problems to decide whether $H \leqslant G$ for input graphs H, G where $\leqslant$ is 'isomorphic to a subgraph', 'isomorphic to an induced subgraphs', 'isomorphic to a subdivision', 'isomorphic to a contraction' or their combination, are NP-complete. We discuss the complexity of these problems when G is restricted to be a partial k-tree (in other terminology: to have tree-width $\leqslant k$, to be k-decomposable, to have dimension $\leqslant k$). Under this restriction the problems are still NP-complete in general, but there are polynomial algorithms under some natural restrictions imposed on H, for example when H has bounded degrees.

We also give a polynomial time algorithm for the n disjoint connecting paths problem restricted to partial k-trees (with n part of input).

1. Introduction

In this paper graphs are finite, without loops and multiple edges. We need to consider many graph-theoretic inclusions and it is convenient to have a consistent notation for them. We propose the following one. We write $H \leqslant_h G$ if G is isomorphic to a graph that can be obtained from H by subdividing its edges and $H \leqslant_m G$ if H is isomorphic to a graph that can be obtained from G by contracting edges and deleting loops and multiple edges thus produced (here h stands for homeomorphic and m for minor embedding). We write $H = G$ if H is isomorphic to G, $H \subseteq G$ if H is isomorphic to a subgraph of G and $H \subseteq_i G$ if H is isomorphic to an induced subgraph of G. Finally, we define $H \subseteq_h G$ $(H \subseteq_{ih} G$, $H \subseteq_m G$,

* This research was carried out at Department of Mathematical Analysis, Charles University, Sokolovská 83, 186 00 Praha 8, Czechoslovakia.

$H \subseteq_{im} G$, resp.) if there is a graph G' such that $H \leqslant_h G' \subseteq G$ ($H \leqslant_h G' \subseteq_i G$, $H \leqslant_m G' \subseteq G$, $H \subseteq_m G' \subseteq_i G$, resp.).

Let $\leqslant$ be any of the above defined relations. We introduce the following.

Problem 1.1 (*$\leqslant$-decision problem*).
Instance: Graphs H, G.
Question: Is $H \leqslant G$?

As this is easily seen to be NP-complete (or isomorphism-complete for '$\leqslant$' = '$=$') in general, we introduce the following restriction on G. We say that G is a partial k-tree if G is a subgraph of a chordal graph which contains no K_{k+2}, the complete graph on $k+2$ vertices. Our results concern the case when the graph H is connected and the graph G is constrained to be a partial k-tree.

It turns out that the isomorphism of partial k-trees is decidable in polynomial time, while for all the remaining relations the problem remains NP-complete. We investigate further conditions implying polynomial-time solvability. One such condition is very natural—bounding the maximum degree of the graph H by a constant. Another (maybe surprising) restriction is the requirement that H be (vertex) k-connected (and, as before, G be a partial k-tree). This limits the form of possible embeddings and yields a polynomial solvability for all relations except for the minor embeddability (the intuitive reason why minor embedding behaves differently is that there are too many possibilities how to blow up a vertex of high degree in H).

The results are summarized in Table 1. The entries contain the complexity result (P stands for polynomial time solvable) and the location in this paper where we state and prove that result (and also the references to results known before, modulo our knowledge).

The problem indicated by '?' leads to a matching-type problem whose complexity is unknown (only probabilistic polynomial-time algorithms for such type of problems were given by Lovász). The isomorphism for general graphs of bounded degree was solved by Luks [7]. The polynomial-time test for subgraph relation among trees has been known [4]. A polynomial algorithm for the

Table 1
$\leqslant$-decision when H is connected and G is a partial k-tree

$\leqslant H$	$\Delta(H)$ bounded	k-connected	arbitrary
$=$	P [7], (6.14)	P (6.14)	P (6.14)
$\subseteq$ or $\subseteq_i$	P (5.14)	P (6.13), (6.14)	P for $k=1$ [4], (6.14)
			NP-complete $k>1$ (4.4)
$\leqslant_h$	P (5.14)	P (6.14)	P (6.14)
$\leqslant_m$	P (5.14)	NP-complete (4.3)	NP-complete (4.1)
$\subseteq_h$ or $\subseteq_{ih}$	P (5.14)	P for $k=1,2$ (6.14)	P for $k=1$
		? for $k>2$	NP-complete $k>1$ (4.4)
$\subseteq_m$ or $\subseteq_{im}$	P (5.14)	NP-complete (4.3)	NP-complete (4.2)

$\subseteq$-decision problem when H is connected, $\Delta(H) \leq$ const and G is a partial k-tree was independently found by Bodlaender [3]. We suspect that also other results might have been known, but so far we could not find any references. Let us also mention that the situation changes drastically if the graph H is fixed. Polynomial time algorithms in this case (without any restriction on G) were recently produced by Robertson and Seymour [10].

In the next section we introduce another definition of a partial k-tree, more convenient from the algorithmic point of view. In Section 3 we give a basic algorithm which, roughly speaking, computes some specified information about the whole partial k-tree G by a knowledge of this information on smaller pieces of G. We also illustrate the use of this algorithm.

Section 4 contains all our NP-completeness results, while in Sections 5 and 6 we use the basic algorithm to obtain polynomial time algorithms for the results indicated in Table 1.

Our methods also imply a polynomial time algorithm to solve the following problem, widely discussed in the literature for various classes of graphs:

Problem 1.2 (*Disjoint connecting paths problem for partial k-trees*).

Instance: Graph G, which is a partial k-tree and vertices $s_1, t_1, \ldots, s_n, t_n$ of G.

Question: Do there exists n vertex disjoint paths $P_1, \ldots, P_n$ in G in such a way that P_i connects s_i and t_i ($i = 1, \ldots, n$)?

The algorithm is given in Section 7. If we drop the partial k-tree assumption, then this problem turns out to be NP-complete [6], but a polynomial algorithm exists when n is fixed (without any restriction on G) [10].

Let us introduce some terminology. Throughout the paper, w, Δ will be arbitrary but fixed integers. If A is a set, then $|A|$ denotes the cardinality of A and 2^A the set of all subsets of A. A graph is a pair $G = (V, E)$, where V, the set of vertices, is a finite set and E, the set of edges, is a collection of two-element subsets of V. We write $V(G) = V$ and $E(G) = E$. If $\{u, v\} \in E(G)$ is an edge, we say that u, v are its endpoints. If A, B are disjoint subsets of $V(G)$ we say that A, B are *adjacent* if there is an edge $\{a, b\} \in E(G)$ such that $a \in A$ and $b \in B$. We say that $u, v \in V(G)$ are *adjacent* if $\{u, v\} \in E(G)$ (i.e., if $\{u\}$, $\{v\}$ are adjacent). If $A \subseteq V(G)$, then $G \mid A$ denotes the graph induced (or spanned) by the set A and $G \backslash A$ denotes the graph resulting from removing all vertices from A and all edges with at least one endpoint in A. The *degree* of a vertex v in a graph G, denoted, by $\deg_G(v)$, is the number of other vertices adjacent to it. For a graph G, $\Delta(G)$ denotes the maximal degree of G, i.e., the maximum of degrees of its vertices.

Let G be a graph and $Z \subseteq V(G)$. A *Z-separation* is a pair (G_1, G_2) of induced subgraphs of G such that $V(G_1) \cap V(G_2) \subseteq Z$, $V(G_1) \cup V(G_2) \cup Z = V(G)$ and $E(G_1) \cup E(G_2) \cup E(G \mid Z) = E(G)$. A *partial Z-separation* is a pair (H_1, H_2) such that there exists a Z-separation (G_1, G_2) such that $V(H_i) = V(G_i)$, ($i = 1, 2$) and $E(H_1) \cup E(H_2) = E(G_1) \cup E(G_2)$. A Z-component of G is a connected component

C of $G \backslash Z$ together with all edges from C to Z and all ends of these edges. The ends of these edges are called *vertices of attachment* of the Z-component. If A is the set of vertices of attachment of a Z-component C, we say that C is *attached at* A and that C has *order* $|A|$.

If H and G are graphs and $V \subseteq V(H) \cap V(G)$, we say that H is *V-isomorphic* to G if there exists an isomorphism between H and G whose restriction to V is the identity map.

Definition 1.3. If G is a graph then G^* denotes the graph obtained from G by subdividing every edge exactly once.

Lemma 1.4. *Let H, G be graphs, let $\leqslant$ be any of $\subseteq$, $\subseteq_h$, $\subseteq_m$ and let $\leqslant_i$ be its induced counterpart, i.e., $\subseteq_i$, $\subseteq_{ih}$ or $\subseteq_{im}$, respectively. Then $H \leqslant G$ if and only if $H^* \leqslant_i G^*$.*

Proof. The proof is elementary and is left to the reader. □

2. Tree-decompositions

Definition 2.1. A *tree-decomposition* of a graph G is a pair (T, X), where T is a tree and $X = (X_t \mid t \in V(T))$ is such that:

(a) $\bigcup_{t \in V(T)} X_t = V(G)$,

(b) for every edge $\{u, v\}$ of G there exists $t \in V(T)$ such that $u, v \in X_t$, and

(c) whenever t' is on the path between t and t'' in T, then $X_t \cap X_{t''} \subseteq X_{t'}$.

The *width* of a tree decomposition (T, X) is.

$$\max_{t \in V(T)} (|X_t| - 1).$$

The *tree-width* of a graph G is the least integer w such that G admits a tree-decomposition of width $\leqslant w$.

The following two well-known results summarize some of the properties of tree-width.

Theorem 2.2. *Let G be a graph and k an integer. Then G is a partial k-tree if and only if it has tree-width $\leqslant k$.*

Proof. (Sketch) ($\Leftarrow$) Let H be the graph obtained from G by adding all edges $\{u, v\}$ such that $u, v \in X_t$ for some $t \in V(T)$. Then H is chordal and contains no K_{k+2}.

($\Rightarrow$) Let G be a subgraph of a chordal graph H, which contains no K_{k+2}. Every chordal graph contains a vertex whose neighbors induce a complete subgraph. By successive elimination of such vertices we can construct the desired tree-decomposition (T, X). It also follows that this tree-decomposition satisfies $|V(T)| \leqslant |V(G)|$. □

Proposition 2.3. (i) *G has tree-width $\leqslant 1$ iff G is a forest.*

(ii) *G has tree-width $\leqslant 2$ iff G is series-parallel.*

(iii) *The complete graph K_n has tree-width $n-1$.*

(iv) *If $H \subseteq_m G$, then the tree-width of H is at most the tree-width of G.*

(v) *The tree-width of the adjacency graph of the $n \times n$ chessboard is n.*

Proof. We shall use only (iii) and (iv), and these are easily seen. □

Since we shall assume that input graphs come already equipped with their tree-decomposition, it is worth mentioning how such a tree-decomposition can be found.

Theorem 2.4. *Let w be fixed.*

(i) *There is an $O(|V(G)|^{w+2})$ algorithm which, for an input graph G, decides whether its tree-width is $\leqslant w$, and if yes, it also constructs a corresponding tree-decomposition,* [1].

(ii) *There is an $O(|V(G)|^2)$ algorithm which for an input graph G either finds a tree-decomposition of tree-width $\leqslant 3w$ or finds out that the tree-width of G is $\geqslant w$,* [10].

(iii) *There exists an $O(|V(G)|^2)$ algorithm which decides whether the tree-width of G is $\leqslant w$ (but does not construct the tree-decomposition if so),* [10]. *(This is just an existence statement and its proof gives no hint how to construct such an algorithm.)*

(iv) *There is an $O(|V(G)| \log^2 |V(G)|$ probabilistic algorithm which for the input graph G either finds a tree-decomposition of width $\leqslant 6w$ or finds out that the tree-width of G is $\geqslant w$,* [8].

(v) *There is a linear algorithm for the problem described in (i) if $w \leqslant 3$,* [8].

(vi) *If w is variable part of the input instance, then the problem to decide if tree-width of G is $\leqslant w$ is* NP-*complete,* [1].

Definition 2.5. Let G be a graph. We say that a tree-decomposition (T, X) of G is standard, if every vertex of T has degree either 1 or 3 and $|V(T)| \leqslant 2|V(G)|$.

Theorem 2.6. *Every graph G admits a standard tree-decomposition.*

Proof. It is clear that the tree-decomposition (T, X) obtained in the proof of Theorem 2.2 satisfies $|V(T)| \leqslant |V(G)|$. By adding some vertices t with $X_t = \emptyset$ we can arrange that T has no vertices of degree 2. By 'splitting' vertices of degree >3 we can arrange that every vertex has degree 3 or 1. □

One can also show that an arbitrary tree-decomposition can be modified to a standard one in linear time, and hence there is no harm in supposing that all tree-decompositions are standard.

3. The basic algorithm

This section gives a generic form of an algorithm computing on a graph G equipped with a tree-decomposition of bounded width. We formulate it as the computation of some function of the whole graph, provided that certain assumptions on computability and behavior of this function on subgraphs of G hold.

Definition 3.1. A *transition function* is a function P assigning a set $P(G, Z)$ to each pair (G, Z), where G is a graph and $Z \subseteq V(G)$ is such that $|Z| \leq w+1$ (recall that w is a fixed integer).

Definition 3.2. A transition function P is called *feasible* if there are functions T_0, T_R, T_M of $|V(G)|$, such that for every G and every $Z \subseteq V(G)$, $|Z| \leq w+1$ the following holds:

(a) $P(G, Z)$ can be determined in time T_0 for every G such that $|V(G)| \leq w+1$.

(b) For $Z \supseteq Z'$, $P(G, Z')$ can be determined by a knowledge of $P(G, Z)$ in time T_R.

(c) If (G_1, G_2) is a Z-separation of G, then $P(G, Z)$ can be determined by a knowledge of $P(G_j, Z \cap V(G_j))$ $(j = 1, 2)$ in time T_M.

Let a feasible transition function P be given.

Algorithm 3.3

Input: A graph G and its standard tree-decomposition (T, X) of width $\leq w$.

Output: $P(G, \emptyset)$.

Description. We use the ideas of [2], which appear in [10, Algorithm (4.1)]. Let r be a leaf of T, and number the vertices of T as $t_1, \ldots, t_m = r$, where $m = |V(T)|$, so that the numbers on every path leaving r are decreasing. For $i = 1, \ldots, m$ let B_i be the set of all vertices t of T such that t_i lies on the path between t and r in T and let G_i be the graph spanned by the set $\bigcup \{X_t \mid t \in B_i\}$. Let also $Z_i = X_{t_i}$. Note that $G_m = G$. We shall compute $P(G_i, Z_i)$ for each i by recursion as follows. At the start of the ith iteration, $P(G_j, Z_j)$ has been determined for $1 \leq j < i$.

(a) If $i < m$ and t_i has degree 1, then $V(G_i) = Z_i = X_{t_i}$ and we use Definition 3.2(a).

(b) If t_i has degree > 1, let $j, k < i$ be such that $\{t_i, t_j\} \in E(T)$, $\{t_i, t_k\} \in E(T)$. Using Definition 3.2(b) we can determine $P(G_j, Z_j \cap Z_i) = P(G_j, V(G_j) \cap Z_i)$ and

$P(G_k, Z_k \cap Z_i) = P(G_k, V(G_k) \cap Z_i)$, and since (G_j, G_k) is a Z_i-separation of G_i (this is a basic property of tree-decompositions—see e.g. [10]), we can use Definition 3.2(c) to determine $P(G_i, Z_i)$.

(c) If $i = m$ then $(G_{m-1}$, empty graph) is a Z_m-separation of G_m and so having determined $P(G_{m-1}, Z_m \cap V(G_{m-1}))$ using Definition 3.2(b) we can use Definition 3.2(c) to determine $P(G_m, Z_m)$. Finally, we use Definition 3.2(b) to determine $P(G, \emptyset) = P(G_m, \emptyset)$ from $P(G_m, Z_m)$.

Theorem 3.4. *The worst-case running time of the above algorithm is*

$$O(T_0 + T_R + T_M) \cdot |V(G)|.$$

Proof. There are $|V(T)| = O(|V(G)|)$ steps to be done and each takes time at most $T_0 + T_R + T_M$. $\square$

We end this section by a list of examples which illustrate the use of Algorithm 3.3.

Example 3.5. Let $P(G, Z)$ consist of one element, namely the mapping

$$f: 2^Z \to \{0, 1, 2, \ldots\}$$

defined by saying that for $V \subseteq Z$, $f(V)$ is the size of maximum independent set M in G such that $M \cap Z = V$ (if such a M exists; otherwise the value is irrelevant and we may set it e.g. to 0). Then from $P(G, \emptyset)$ we can read off the maximum independent set size of G.

Example 3.6 [2]. Let $|Z|^Z$ be the set of all functions $Z \to |Z|$. Let $P(G, Z)$ consist of one element, namely the mapping

$$f: |Z|^Z \to \{0, 1, 2, \ldots\},$$

which to each $r \in |Z|^Z$ assigns the least n such that there is a coloring of G by n colors, which extends r. Then from $P(G, \emptyset)$ we can read off the chromatic number of G.

The last example assumes familiarity with [10]:

Example 3.7 [10]. Let $X \subseteq V(G)$ be given. With

$$P(G, Z) = \text{the } \delta\text{-folio of } G \text{ relative to } Z \cup X,$$

Algorithm 3.3 reduces to Algorithm (4.1) of [10]. This can be used to solve the $\subseteq_m$-decision problem for fixed H or the disjoint paths problem for fixed n (provided that G has bounded tree-width).

4. NP-completeness

Theorem 4.1. *The $\leqslant_m$-decision problem is* NP-*complete even if we impose one of the following restrictions on G and H*:

(i) *H and G are trees of bounded diameter,*

(ii) *H and G are trees all whose vertices but one have degree* $\leqslant 5$.

(By a somewhat more complicated construction we could have degrees $\leqslant 3$ in (ii).)

Proof. We proceed by reduction from a 3-matching problem of the following form [4]: $X = \{x_1, \ldots, x_{3n}\}$ is a set and $\mathcal{T} = \{t_1, \ldots, t_m\}$ is a set of triples of points of X (we may assume $m > n+1$), and we ask whether all points of X can be covered by a collection of n disjoint triples of $\mathcal{T}$. We construct trees H and G such that $H \leqslant_m G$ iff $(X, \mathcal{T})$ has a solution.

Let $\tau_1, \ldots, \tau_{3n}$ be pairwise $\leqslant_m$-incomparable rooted trees, i.e., trees with a specified vertex of degree 1, called the root for which the $\leqslant_m$-relation is defined with the additional requirement that root is contracted onto root. Such a family of trees is easy to construct; moreover, we may assume that none of the τ_i is a path.

The height of a rooted tree is the length of the longest path starting from the root. Let h be the maximum of the heights of $\tau_1, \ldots, \tau_{3n}$. Let H arise by gluing $\tau_1, \tau_2, \ldots, \tau_{3n}$ (by their roots) and $(m-n)$ paths of length $h+2$ (by their endpoints), see Fig. 1. Let G arise by gluing trees $T_1, \ldots, T_m$ by the root, where T_j is obtained by gluing a path of length $h+1$, τ_{i_1}, τ_{i_2}, τ_{i_3} and an extra edge e_j, see Fig. 2. Here i_1, i_2, i_3 are such that $t_j = \{x_{i_1}, x_{i_2}, x_{i_3}\}$.

Suppose that $H \leqslant_m G$. Then the root of H must be mapped onto the root of G, since they are both unique middle points on a longest path in H and G. Now at most n of the edges $e_1, \ldots, e_m$ may be contracted since H contains $m-n$ paths of length $h+2$ starting from the root. These contracted edges define a solution to

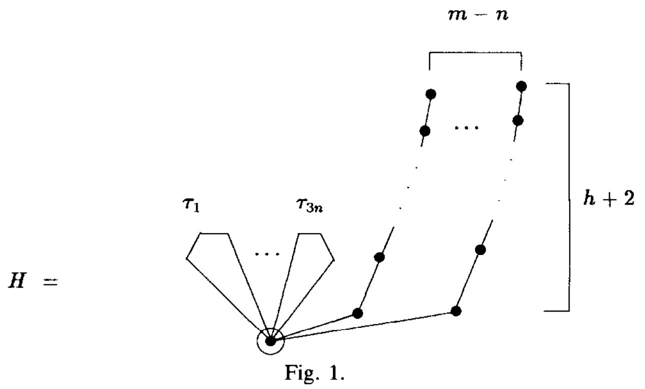


Fig. 1.

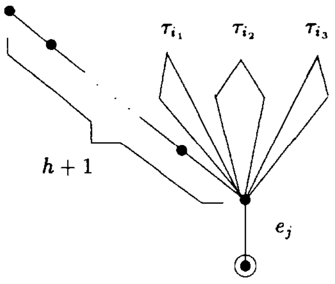
$T_j =$ 

Fig. 2.

$(X, \mathcal{T})$, because each τ_i in H must be embedded onto the τ_i of G by the incomparability of the τ_i.

Conversely it is easy to see that a solution of $(X, \mathcal{T})$ gives rise to a contraction of G isomorphic to H. Now the trees $\tau_1, \ldots, \tau_{3n}$ can be chosen to have either bounded height (then we get (i)) or all degrees $\leqslant 3$ (and we get (ii)). □

Theorem 4.2. *The $\subseteq_m$- and $\subseteq_{im}$-decision problems are also* NP*-complete under either of the restrictions of Theorem* (4.1).

Proof. The relations $\leqslant_m$, $\subseteq_m$ and $\subseteq_{im}$ coincide on the set of trees. Thus the theorem follows from Theorem 4.1. □

Theorem 4.3. *For every fixed k, the $\leqslant$-decision problem is* NP*-complete for $\leqslant \in \{\leqslant_m, \subseteq_m, \subseteq_{im}\}$ if we restrict G on partial k-trees and H on k-connected graphs.*

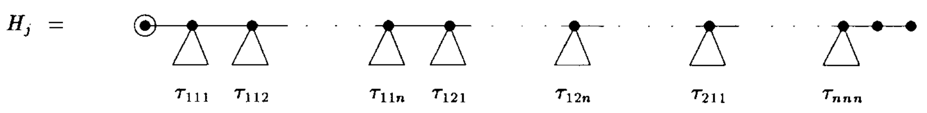
Proof. For $k = 1$ the theorem holds by Theorems 4.1, 4.2. If $k > 1$, we use the following simple reduction to the case $k = 1$: Take the trees G, H constructed in the reduction from the given 3-matching problem. To both these trees, add $k - 1$ new vertices, each joined by an edge to all the vertices of the original tree and also to all of the other new vertices. Now one can prove that the new graphs are in the $\leqslant$ relation iff the old ones were (for all the meaning of $\leqslant$). We omit this proof. □

Theorem 4.4. *For $\leqslant \in \{\subseteq_h, \subseteq_{ih}, \subseteq, \subseteq_i\}$, the $\leqslant$-decision problem is* NP*-complete even if we impose the following two restrictions on H and G:*

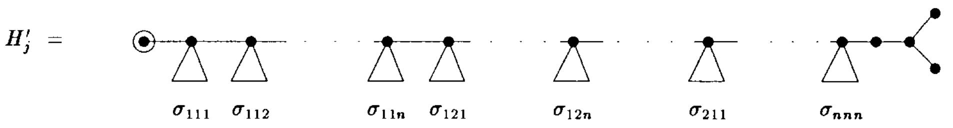
(i) *H is a tree all whose vertices but one have degree $\leqslant 3$,*

(ii) *G has tree-width 2 and all vertices but one have degree $\leqslant 3$.*

Proof. We shall assume that $\leqslant \in \{\subseteq_h, \subseteq\}$, since the other two cases follow from Lemma 1.4. Consider a 3-matching problem $(X, \mathcal{T})$, where $X = R \cup S \cup T$, $R = \{r_1, \ldots, r_n\}$, $S = \{s_1, \ldots, s_n\}$, $T = \{t_1, \ldots, t_n\}$ and $\mathcal{T} \subseteq R \times S \times T$. We



where $\tau_{kpq} = \begin{cases} \bullet & \text{if } p \neq j \\ \bullet\!-\!\bullet & \text{if } p = j \end{cases}$



where $\sigma_{kpq} = \begin{cases} \bullet & \text{if } \{r_k, s_p, t_j\} \in \mathcal{T} \text{ or } j \neq q \\ \bullet\!-\!\bullet & \text{otherwise} \end{cases}$

Fig. 3.

construct H and G in such a way that $H \leq G$ if and only if $(X, \mathcal{T})$ has a solution. Let H arise by identifying roots of the trees $H_1, \ldots, H_n$ and $H'_1, \ldots, H'_n$, where H_j and H'_j are depicted in Fig. 3. Let G arise by identification of the roots of $G_1, \ldots, G_n$, where G_i is depicted in Fig. 4.

Now if $H \subseteq G$ or $H \subseteq_h G$ then the root of H must go onto the root of G, every G_i must contain some H_p and some H'_q, and the 'horizontal' paths must be mapped bijectively. If H_p and H'_q are embedded into G_i, then $\{r_i, s_p, t_q\} \in \mathcal{T}$ since otherwise both H_p and H'_q would contain a tail at position (i, p, q), which is impossible since they are embedded into G_i. Hence the assignment $i \mapsto (p, q)$

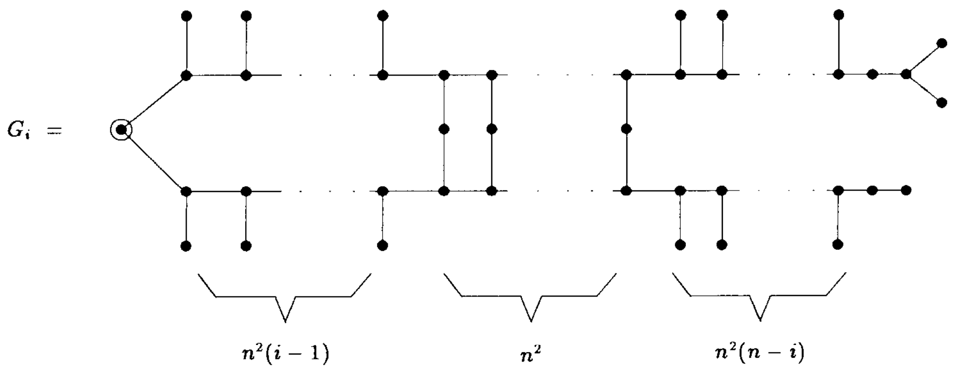


Fig. 4.

where H_p and H'_q are embedded into G_i, defines a solution

$$\{\{r_i, s_p, t_q\}: i = 1, \ldots, n\} \subseteq \mathcal{T}$$

of $(X, \mathcal{T})$.

Conversely, it is easy to see that every solution to $(X, \mathcal{T})$ gives rise to a subgraph of G isomorphic to H in a similar way. □

5. Expansions

Let k be a fixed integer. In this section we shall deal with the following.

Problem 5.1. (*(Δ, k)-restricted $\leqslant$-decision problem*).

Instance: Graphs H, G, here H is connected and $\Delta(H) \leqslant \Delta$, and a standard tree-decomposition (T, X) of G of width $\leqslant k$.

Question: Is $H \leqslant G$?

We shall describe the algorithm in full detail for the induced minor embedding case. The case of induced subgraph relation will be a trivial simplification of the induced minor embedding case. The homeomorphic embedding is slightly more different, and we shall sketch the approach at the end of this section.

Let us describe the idea of the algorithm first. It turns out that $H \leqslant G$ is equivalent to the existence of a mapping $\varphi: V(H) \to 2^{V(G)}$ with certain properties, which we call an expansion. If such an expansion is chosen, Z is a (small) cutset of $V(G)$, and (G_1, G_2) is a Z-separation of G, then this turns out to induce a cutset $W \subseteq V(H)$, such that each component of $H \setminus W$ is mapped either into G_1 or into G_2. Conversely, if we have an expansion φ_1 of some collection of W-components of H into G_1 and an expansion φ_2 of some collection of W-components of H into G_2, we may ask whether these expansions can be merged into an expansion φ of the union of both collections of W-components. It turns out that for this decision we need not know the whole expansions φ_1 and φ_2, but only some bounded size information how these behave on W.

Of course, there are many ways how to choose this information; we have attempted to select one which is easy to describe. For every pair (G_j, Z_j) occurring in the computation of the basic algorithm (of Section 3) we will compute a suitable description of all possible partial embeddings φ of H into G_j 'relative to' Z_j: this information describes the behavior of the embedding on Z_j exactly, and specifies which components of $H \setminus \varphi^{-1}(Z_j)$ are embedded into G_j (and also the components of $G_j \setminus Z_j$ they are embedded into). This is essentially how the algorithm works for induced subgraph embeddings; for the minor case we must add the description how the vertices of $\varphi^{-1}(Z_j)$ are 'blown up'. Here again we cannot afford to describe the whole subgraph of G_j corresponding to such a vertex exactly (we would get too many possibilities), but we give only the vertices belonging to Z_j and the way these are connected in G_j. To make all this precise, we must introduce several technical definitions.

Definition 5.2. Let H be a graph and $W \subseteq V(H)$. A *full W-subgraph* of H is an induced subgraph K of H such that $W \subseteq V(K)$ and for every u, $v \in V(H) \setminus W$, if u, v belong to the same component of $H \setminus W$ and $u \in V(K)$, then $v \in V(K)$.

A *W-subgraph* of H is any subgraph of H which can be obtained from a full W-subgraph of H by (possibly) deleting some edges of H with both endpoints in W.

Since there are at most $\Delta(H) \cdot |W|$ components of $H \setminus W$, there are at most

$$2^{\Delta(H) \cdot |W|}$$

full W-subgraphs of H and thus at most

$$2^{\Delta(H) \cdot |W|} \cdot 2^{\binom{|W|}{2}}$$

W-subgraphs of H.

Definition 5.3. Let H, G be graphs. A mapping $\varphi : V(H) \to 2^{V(G)}$ is called an *abstract expansion of H into G* if

(E1) $\varphi(v) \cap \varphi(v') = \emptyset$ for distinct v, $v' \in V(H)$, and

(E2) for any u, $v \in V(H)$, u, v are adjacent in H if and only if $\varphi(u)$, $\varphi(v)$ are adjacent in G.

We denote by $\varphi(H)$ the graph induced by $\bigcup_{v \in V(H)} \varphi(v)$ in G. An abstract expansion φ is said to be *surjective* if $\varphi(H) = G$.

Definition 5.4. Let H, G be graphs, $W \subseteq V(H)$, $Z \subseteq V(G)$ and let $\psi : W \to 2^Z \setminus \{\emptyset\}$ be a mapping. We say that an abstract expansion φ of H into G is a *(minor) extension* of ψ if:

(M1) $\varphi(v) \cap Z = \psi(v)$ for $v \in W$, $\varphi(v) \cap Z = \emptyset$ for $v \in V(H) \setminus W$,

(M2) $G \mid \varphi(v)$ is connected for $v \in V(H) \setminus W$,

(M3) each component of $G \mid \varphi(v)$ intersects $\psi(v)$ for $v \in W$.

We say that φ is a *(minor) expansion* if φ is an extension of the empty map.

Observation 5.5. *Let H, G be graphs. There exists an expansion of H into G if and only if $H \subseteq_{mi} G$.*

We reformulate Problem 5.1 for the induced minor relation as follows.

Problem 5.6. *Instance*: Graphs H, G, where H is connected and $\Delta(H) \leq \Delta$, and a standard tree-decomposition (T, X) of G of width $\leq k$.

Question: Does there exist an expansion of H into G?

Definition 5.7. Let G be a graph and let Z, $V \subseteq V(G)$. A *Z-model of V in G* is an equivalence relation on $Z \cap V$: two vertices are equivalent iff they belong to the same component of $G \mid V$.

Definition 5.8. Let H, G be graphs and $Z \subseteq V(G)$. A *partial expansion of H into G relative to Z is a quadruple* (W, ψ, K, M) such that $W \subseteq V(H)$, $\psi : W \to 2^Z \setminus \{\emptyset\}$ satisfies (E1), K is a W-subgraph of H and $M = (M(v) \mid v \in W)$, where each $M(v)$ is an equivalence relation on $\psi(v)$ (such that each components of $G \mid \psi(v)$ is in a single class).

Let us bound the number of all partial expansions of H into G relative to Z: There are at most $|V(H)|^{|Z|}$ ways to choose W, then (by Definition (5.2)) at most

$$2^{\Delta(H)\cdot|Z|} \cdot 2^{\binom{|Z|}{2}}$$

W-subgraphs of H. The choice of ψ can be viewed as coloring of Z by $|W| + 1$ colors (thus at most $(|Z| + 1)^{|Z|}$ ways), and finally the choices of the equivalences $M(v)$ can be viewed as choosing a single equivalence relation on Z, thus at most $|Z|^{|Z|}$ ways. The essential point for us is that if the size of Z is bounded by a constant k, then there are at most $O(|V(H)|^k)$ partial expansions of H into G relative to Z.

A partial expansion $\mathscr{E} = (W, \psi, K, (M(v) \mid v \in W))$ is called *feasible* if there exists an expansion φ of K into G, called a *feasible extension of $\mathscr{E}$*, which is an extension of ψ such that, for $v \in W$,

$$M(v) \text{ is a } \psi(v)\text{-model of } \varphi(v) \text{ in } \varphi(K). \qquad (*)$$

Conversely, given $Z \subseteq V(G)$ and an expansion φ of K into G we can produce a partial expansion $\mathscr{E} = (W, \psi, K, (M(v) \mid v \in W))$, called the *saturation* of φ at Z, by the rules

$$W = \{w \in V(H) \mid \varphi(w) \cap Z \neq \emptyset\},$$

$$\psi(w) = \varphi(w) \cap Z \quad \text{for } w \in W,$$

and $M(v)$ is defined by $(*)$. By definition, the saturation of φ at Z is always feasible, namely φ is its feasible extension.

Definition 5.9. The set of all feasible partial expansions of H into G relative to Z will be denoted by $P_H(G, Z)$. Then P_H is a transition function in the sense of (3.1). To apply the basic algorithm we must show that it is feasible. To this end, we need two technical lemmas. The first one gives the conditions allowing us to recognize, given a potential member of $P_H(G, Z)$, whether it is feasible, using the knowledge of $P_H(G, Z')$ (where $Z \subseteq Z'$).

Lemma 5.10. *Let $\mathscr{E} = (W, \psi, K, (M(v) \mid v \in W))$ be a partial expansion of H into G relative to Z and let $Z \subseteq Z'$. Then $\mathscr{E}$ is feasible if and only if there exists a* *feasible partial expansion* $(W', \psi', K', (M'(v) \mid v \in W))$ *of* H *into* G *relative to* Z' *such that*:

(a) $W = \{w \in W' \mid \psi'(w) \cap Z \neq \emptyset\}$,

(b) $\psi(v) = \psi'(w) \cap Z$ *for* $w \in W$,

(c) $K = K'$,

(d) *each class of* $M'(v)$ *intersects* $\psi(v)$ $(v \in W)$,

(e) $M(v)$ *is equal to* $M'(v)$ *restricted on* $\psi(v)$, $(v \in W)$,

(f) $M'(v)$ *has a single class for* $v \in W' \setminus W$.

Proof. $(\Rightarrow)$ Let φ be a feasible extension of $\mathscr{E}$, let $K' = K$ and let $\mathscr{E}' = (W', \psi', K', (M'(v) \mid v \in W))$ be the saturation of φ at Z'. Then $\mathscr{E}'$ is a partial expansion and φ is its feasible extension. We must verify (a)–(f).

To prove (a) and (b) we have by (M1)

$$\psi'(w) \cap Z = \varphi(w) \cap Z = \psi(v) \quad \text{for } w \in W,$$

$$\psi'(w) \cap Z = \varphi(w) \cap Z = \emptyset \quad \text{for } w \in W' \setminus W.$$

Conditions (c), (e) are clear. The violation of (d) would mean that $\varphi(v)$ has a component not intersecting Z, contradicting (M3), and similarly for (f).

$(\Leftarrow)$ Let $\mathscr{E}' = (W', \psi', K', (M'(v) \mid v \in W'))$ be a feasible partial expansion of H into G relative to Z' satisfying (a)–(f) and let φ be its feasible extension. We claim that φ is a feasible extension of $\mathscr{E}$. To this end, we must verify that $M(v)$ satisfies $(*)$ and that φ satisfies (M1)–(M3).

Condition $(*)$ follows from (e). To prove (M1) we have for $v \in W'$

$$\varphi(v) \cap Z = \varphi(v) \cap Z' \cap Z = \psi'(v) \cap Z,$$

which gives the result by (a) and (b). Condition (M2) is clear for $v \in V(H) \setminus W'$ and for $W' \setminus W$ follows from (f), condition (M3) follows from (d). □

The next lemma lists the conditions for the merging step of the basic algorithm.

Lemma 5.11. *Let* $\mathscr{E} = (W, \psi, K, (M(v) \mid v \in W))$ *be a partial expansion of* H *into* G *relative to* Z *and let* (G_1, G_2) *be a* Z*-separation of* G. *Then* $\mathscr{E}$ *is feasible if and only if there are feasible partial expansions*

$$\mathscr{E}_i = (W_i, \psi_i, K_i, (M_i(v) \mid v \in W_i))$$

of H *into* G_i *relative to* $Z \cap V(G_i)$ $(i = 1, 2)$ *such that*:

(a) $W_i = \{w \in W \mid \psi(v) \cap V(G_i) \neq \emptyset\}$.

(b) $\psi_i(w) = \psi(w) \cap V(G_i)$,

(c) (K_1, K_2) *is a partial* W*-separation of* K *such that* $W_i = V(K_i) \cap W$ *and* $E(K) = E(K_1) \cup E(K_2) \cup \{\{u, v\} \in \binom{W}{2}$, $\psi(u)$ *and* $\psi(v)$ *are adjacent in* $G \mid Z\}$,

(d) $M(v)$ *is the equivalence hull of the relation consisting of the following pairs of vertices*: *those related by* $M_1(v)$, *those related by* $M_2(v)$ *and the pairs* (u, u'), u, $u' \in \psi(v)$, *such that* (u, u') *is an edge of* $G \mid Z$.

The proof of this lemma (similarly as for the previous one) is mainly an untangling of definitions and we leave it to the reader.

We now turn to the algorithmic use of the previous lemmas.

Algorithm 5.12

Input: Grpahs H, G, where H is connected and $\Delta(H) \leqslant \Delta$, and a standard tree-decomposition (T, X) of G of width $\leqslant k$.

Output: $P_H(G, \emptyset)$,

Description: We check if $|V(H)| > |V(G)|$ and if yes, we return $\emptyset$. Otherwise we proceed as follows. First we preprocess the graph H: for each $W \subseteq V(H)$, $|W| \leqslant k+1$ we compute the components of $H \setminus W$, represent each component e.g., by its first vertex (in some ordering of $V(H)$) and prepare a table specifying how the components are inherited when W is replaced by some $W' \subseteq W$. This can be done in time $O(|V(H)|^{k+2}) = O(|V(H)|^{k+1} \cdot |V(G)|)$. Then we can treat the W-subgraphs of H as objects with bounded size representation (a list of components of $H \setminus W$) and perform all necessary operations on them in constant time.

A set $P_H(G, Z)$ will be represented by an array indexed by all possible partial expansions of H into G relative to Z with Boolean entries specifying the feasibility. Then P_H obviously satisfies Definition 3.2(a).

Lemma 5.10 shows that for $Z \subseteq Z'$, $P_H(G, Z)$ can be obtained from $P_H(G, Z')$ in time $O(|V(H)|^{k+1})$ since for each member of $P_H(G, Z')$ we can test in constant time if it gives rise to some member of $P_H(G, Z)$. Hence P_H satisfies Definition 3.2(b) with $T_R = O(|V(H)|^{k+1})$.

Given a Z-separation (G_1, G_2) of G and a partial expansion $\mathscr{E}$ of H into G relative to Z there is only a bounded number of possibilities how $\mathscr{E}$ can be partitioned into partial expansions of H into G, relative to $Z \cap V(G_1)$ and into G_2 relative to $Z \cap V(G_2)$; to having $P_H(G_1, Z \cap V(G_1))$ and $P_H(G_2, Z \cap V(G_2))$ we can test using Lemma 5.11 in constant time whether $\mathscr{E} \in P_H(G, Z)$. Hence P_H satisfies Definition 3.2(c) with $T_M = O(|V(H)|^{k+1})$.

Since p_H satisfies Definition 3.2(a)–(e), we can use Algorithm 3.3.

Theorem 5.13. *The worst-case running time of the above algorithm is*

$$O(|V(H)|^{k+1} \cdot |V(G)|).$$

Proof. This follows immediately from the above description and Theorem 3.4. □

The above algorithm solves the induced minor case and (via Lemma 1.4) the minor case. It is easy to simplify this for the (induced) subgraph case (basically by requiring $|\psi(v)| = 1$ for every $v \in V$ in the definition of expansion). The case of homeomorphic embedding is slightly different. Either we can handle it in the spirit of the minor case, adding more conditions to the definition of expansions and adding more information to Z-models, or (more naturally) we can treat the homeomorphic embedding as blowing up the edges of H into paths in G, and developing a similar machinery for this kind of expansions as we did for the minor ones. This treatment contains no significantly new ideas, and we omit it. Also, all the above work can also be carried out for surjective expansions. We summarize the results in the following.

Theorem 5.14. *For fixed Δ, k there is an* $O(|V(H)|^{k+1} \cdot |V(G)|)$ *algorithm to solve the (Δ, k)-restricted $\leqslant$-decision problem for $\leqslant \in \{\subseteq_i, \subseteq_{ih}, \subseteq_{im}, \subseteq, \subseteq_h, \subseteq_m, \leqslant_h, \leqslant_m\}$.* $\square$

6. Embeddings

This section gives a polynomial algorithm for the $\leqslant$-decision problem for the induced subgraph case, when G is a partial k-tree and H is k-connected. In the sequel we shall deal with induced subgraph embeddings only, so it is not necessary to use the machinery of expansions.

Definition 6.1. A mapping $\varphi : V(H) \to V(G)$ is called an *embedding of H into G* if it induces an isomorphism between H and $G \mid \varphi(H)$. Thus there exists an embedding of H into G iff $H \subseteq_i G$.

Problem 6.2. (*k-connected $\subseteq_i$-problem*).

Instance: A k-connected graph H, graph G and a standard tree-decomposition (T, X) of G of width $\leqslant k$.

Question: is $H \subseteq_i G$?

We shall describe a polynomially bounded algorithm to solve this problem. We shall use the basic algorithm, this time verifying Definition 3.2(a)–(e) only for those pairs (Γ, Z) for which $P(\Gamma, Z)$ is actually computed during execution of the basic algorithm.

Definition 6.3. Let G be a graph. By $\mathcal{V}(G)$ we denote the set of pairs (Γ, Z) such that the value of the transition function is evaluated at (Γ, Z) during some execution of the basic algorithm on the graph G. We have included the word 'some' because there might be several ways of execution depending on the choice of a tree-decomposition (T, X) of G and of a leaf of T. (We could also fix a particular tree-decomposition (T, X) and a leaf of T.)

Lemma 6.4. *Let $(\Gamma, Z) \in \mathcal{V}(G)$ and let Γ' be the graph obtained from Γ by joining all pairs of vertices from Z by an edge. Then Γ is a full Z-subgraph of G and the tree-width of Γ' is $\leqslant w$.*

Proof. The first assertion follows from Definition 2.1(c). For the second, let $(T, X) = (T, X_t \mid t \in V(T)))$ be a tree-decomposition of G of width $\leqslant k$ such that (Γ, Z) appears when executing 'according to (T, X)'. Then $Z \subseteq X_t$ for some $t \in V(T)$ and it follows that $(T, (X_t \cap V(\Gamma) \mid t \in V(T)))$ is a tree-decomposition of Γ' of width $\leqslant k$. □

The following is a basic observation which makes the algorithm work for a w-connected graph H.

Lemma 6.5. *Let H be a k-connected graph and let G be a graph of tree-width $\leqslant k$. Let $(\Gamma, Z) \in \mathcal{V}(G)$, $W \subseteq V(H)$ and let φ be an embedding of some full W-subgraph K of H into Γ such that $W = \varphi^{-1}(Z)$. Then each component of $\Gamma \setminus Z$ contains at most one component of $\varphi(K) \setminus Z$.*

Proof. Let a component C of $\Gamma \setminus Z$ contain components C_1 and C_2 of $\varphi(K) \setminus Z$. Let $Z' \subseteq Z$ be the set of vertices which are joined to C by an edge. It follows from the k-connectivity of H that $|Z'| \geqslant k$ (hence $|Z'| = k$) and that both C_1 and C_2 are joined to every vertex of Z'. There is a path P between C_1 and C_2 in C (since C is connected). Now let Γ' be as in Lemma 6.4; contracting C_1 to a vertex c_1, C_2 to a vertex c_2 and the path P to an edge with endpoints c_1, c_2 we find that Γ' can be contracted to K_{k+2}-contrary to Lemma 6.4, and Proposition 2.3(iii)–(iv). □

Before applying the basic algorithm we need a definition and two lemmas.

Definition 6.6. Let H, G be graphs and $(\Gamma, Z) \in \mathcal{V}(G)$. A component injective embedding (CIE) scheme for H, Γ relative to Z is a triple (W, ψ, B), where $W \subseteq V(H)$, ψ is an embedding of $H \mid W$ into $\Gamma \mid Z$ and B is a bipartite graph defined as follows:

(i) the vertices of the first (second) partite are components of $H \setminus W$ ($\Gamma \setminus Z$, resp.),

(ii) an edge $\{C, K\} \in E(B)$ iff there is an embedding $\varphi_{C,K}$ of $H \mid (W \cup V(C))$ into $\Gamma \mid (Z \cup V(K))$ which extends ψ.

A CIE scheme is fully determined by the choice of W and ψ, and hence there are

$$O(|V(H)|^{|Z|} \cdot |Z|^{|Z|})$$

CIE schemes for H, Γ relative to Z.

Now for $(\Gamma, Z) \in \mathcal{V}(G)$ we define $P_H(\Gamma, Z)$ to be the set of all CIE schemes of H, Γ relative to Z and we wish to apply the basic algorithm.

Observation 6.7. *$H \subseteq_i G$ if and only if $(\emptyset, \emptyset, B)$ is a CIE scheme for H, G relative to $\emptyset$ with $E(B) \neq \emptyset$.*

Lemma 6.8. *Let (Γ, Z), $(\Gamma, Z') \in \mathscr{V}(G)$, let $Z \subseteq Z'$ and let (W, ψ, B) be a CIE scheme for H, Γ relative to Z. Then $\{C, K\} \in E(B)$ iff there exists a CIE scheme (W', ψ', B') for H, Γ relative to Z' with $W \subseteq W'$ and $\psi = \psi' \mid W$ and there exists a matching of size n in $B' \cap (\{C_1, \ldots, C_n\} \times \{K_1, \ldots, K_m\})$, where $C_1, \ldots, C_n$ are all the components of $H \backslash W'$ meeting C and $K_1, \ldots, K_m$ are all the components of $\Gamma \backslash Z'$ meeting K.*

Proof. We observe that a component of $H \backslash W'(\Gamma \backslash Z')$ is either contained in or disjoint from a component of $H \backslash W$ ($\Gamma \backslash Z$, resp.).

($\Rightarrow$) Suppose that $\{C, K\} \in E(B)$ and let $\varphi = \varphi_{C,K}$ be the corresponding embedding. Let $W' = \varphi^{-1}(Z')$ and $\psi' = \varphi \mid W'$, now W' and ψ' determine the CIE scheme (W', ψ', B') for H, Γ relative to Z'. By Lemma 6.5 φ maps the set $\{C_1, \ldots, C_n\}$ of components of $H \backslash W'$ injectively into the set $\{K_1, \ldots, K_m\}$ of components of $\Gamma \backslash Z'$ and this gives the desired matching in $B' \cap \{C_1, \ldots, C_n\} \times \{K_1, \ldots, K_m\}$).

($\Leftarrow$) Having (W', ψ', B') as in the Lemma and matching $\{\{C_i, K_{\rho(i)}\} \mid i = 1, \ldots, n\} \subseteq E(B')$ we define

$$\varphi_{C,K} = \bigcup_{i=1}^{n} \varphi_{C_i}, K_{\rho(i)},$$

where $\varphi_{C_i, K_{\rho(i)}}$ is the embedding corresponding to $\{C_i, K_{\rho(i)}\}$. Then $\varphi_{C,K}$ is an embedding of $H \mid (W \cup V(C))$ into $\Gamma \mid (Z \cup V(K))$ witnessing $\{C, K\} \in E(B)$. $\square$

Lemma 6.9. *Let $(\Gamma, Z) \in \mathscr{V}(G)$, and let (Γ_1, Γ_2) be a Z-separation of Γ. Then (W, ψ, B) is a CIE scheme for H, Γ relative to Z if and only if ψ is an embedding of $H \mid W$ into $\Gamma \mid Z$ and there are CIE schemes (W_i, ψ_i, B_i) for H, Γ_i relative to $Z_i = Z \cap V(\Gamma_i)$, where $W_i = \psi^{-1}(Z_i)$, $\psi_i = \psi \mid W_i$ $(i = 1,2)$ and $B = B_1 \cup B_2$.*

Proof. Straightforward. $\square$

Definition 6.10. Let b' be such that the maximum matching in a bipartite graph on n vertices can be found in time $O(n^{b'})$. By [5], $b' \leq \frac{5}{2}$. We put $b = \max(b', 2)$.

Algorithm 6.11.

Input: Graphs H, G such that H is w-connected and a standard tree-decomposition (T, X) of G of width $\leq k$.

Output: $P_H(G, \emptyset)$

Description: Similarly as in Algorithm 5.12, we assume a suitable preprocessing of components of $H \backslash W$ and a suitable bookkeeping of components of the already processed parts of G.

Again, we would like to apply the basic algorithm for P_H. Condition Definition 3.2(a) is obviously satisfied with $T_0 = O(|V(H)|^{k+1})$, for Definition 3.2(b) it is sufficient to solve $O(|V(H)|^{k+1})$ bipartite matching problems with $\leqslant 2|V(G)|$ vertices each. Thus $T_R = O(|V(H)|^{k+1} \cdot |V(G)|^{b'})$. Finally, for Definition 3.2(c) we can, given W and ψ, compute B from the knowledge of $P_H(\Gamma_1, Z \cap V(\Gamma_1))$ and $P_H(\Gamma_2, Z \cap V(\Gamma_2))$ in time $O(|E(B)| + |V(B)|) = O(|V(H)| \cdot |V(G)|)$, and hence $T_M = O(|V(H)|^{k+1} \cdot |V(G)|^2)$.

So we may apply the basic algorithm.

Theorem 6.12. *The worst-case running time of the above algorithm is* $O(|V(H)|^{k+1}. |V(G)|^{b+1})$.

Proof. Immediate from the description and Theorem 3.4. □

Theorem 6.13. *For fixed k, there is an* $O(|V(H)|^{k+1} \cdot |V(G)|^{b+1})$ *algorithm to solve* 6.2.

Proof. Immediate from Observation 6.7, Algorithm 6.11 and Theorem 6.12). □

Remarks 6.14. (i) The above algorithm can be easily adapted to the $\subseteq$ relation.

(ii) A similar algorithm can be used for deciding an isomorphism of graphs of bounded tree-width (but the bipartite graphs in the corresponding embedding schemes have a very special form—they are disjoint unions of complete bipartite graphs, and so the procedure is faster). We get complexity $O(|V(H)|^{k+1} \cdot |V(G)|^2)$ in this case.

(iii) By a minor modification we can obtain polynomial algorithms for the $\subseteq_h$- and $\subseteq_{ih}$-decision problems where H is 2-connected and G has tree-width $\leqslant 2$ and for the $\leqslant_h$-decision problem when G has bounded tree-width (and no restriction on H).

We shall not give the proofs here because they are straightforward modifications of the presented one.

The condition that the connectivity of H is k was basically needed to guarantee the statement of Lemma 6.5. If we insist on $[(k+1)/2]$-connectivity of H only, then sometimes more components of $\varphi(K)\backslash Z$ (in the situation of Lemma 6.5) can be embedded into a single component of $\Gamma\backslash Z$, but all such components but one must have a bounded size, which gives some hope for a polynomial algorithm.

Open Problem 6.15. What is the complexity of the $\subseteq$, $\subseteq_i$ and $\subseteq_h$-decision problems under the following restriction:

H is $[(k+1)/2]$-connected and G has tree-width $\leqslant k$.

7. The disjoint connecting paths problem

Problem 7.1. *Instance*: *Graph G, a standard tree-decomposition (T, X) of G of tree-width $\leq k$ and n-tuples $(s_1, \dots, s_n)$, $(t_1, \dots, t_n)$ of distinct vertices of G.*

Question: Do there exist disjoint paths $P_1, \dots, P_n$ in G such that P_i connects s_i and t_i?

We shall solve an apparently more general problem, which is easier to handle.

Definition 7.2. A *terminated graph* is a triple (G, V, E), where G and (V, E) are graphs and $V \subseteq V(G)$. We say that (G, V, E) is *feasible* if there are paths $(P_e)_{e \in E}$ in G such that $e = \{v_1, v_2\} \in E$, $V(P_e) \cap V = \{v_1, v_2\}$ and v_1, v_2 are endpoints of P_e, and for $e \neq e' \in E$, P_e and $P_{e'}$ are vertex-disjoint unless $\emptyset \neq e \cap e'$, say $\{v\} = e \cap e'$, in which case $\{v\} = V(P_e) \cap V(P_{e'})$. The system of paths $(P_e)_{e \in E}$ is called a *path-representation of* (V, E) *in* G and (V, E) is said to be *represented* by $(P_e)_{e \in E}$.

Let (G, V, E) be a terminated graph and $Z \subseteq V(G)$. A *transition scheme* of (G, V, E) relative to Z is a set

$$F \subseteq \binom{V \cup Z}{2}$$

such that:

(a) $F \cap \binom{V}{2} \subseteq E$,

(b) $\deg_{(V \cup Z, F)}(v) = \deg_{(V, E)}(v)$ for $v \in V$,

(c) $\deg_{(V \cup Z, F)}(v) \leq 2$ for $v \in Z \setminus V$.

A transition scheme F is called *feasible* if $(V \cup Z, F)$ has a path-representation in G such that every edge of F with both endpoints in Z is represented by a path vertex disjoint from Z except for its endpoints.

The point of transition schemes is that there are at most a polynomial number (namely at most

$$(3 \cdot |E| + 1)^{|Z|} \cdot 2^{\binom{|Z|}{2}})$$

transition schemes of (G, V, E) relative to Z.

Note that E is a transition scheme of (G, V, E) relative to $\emptyset$, and that E is a feasible transition scheme if and only if the terminated graph (G, V, E) is feasible.

Problem 7.3. *Instance*: Terminated graph (G, V, E) and a standard tree-decomposition (T, X) of G of width $\leq k$.

Question: Is (G, V, E) feasible?

Lemma 7.4. *Let F be a transition scheme of (G, V, E) relative to Z and let $Z \subseteq Z'$. Then F is feasible if and only if there exists a feasible transition scheme F' of (G, V, E) relative to Z' such that the graph $(V \cup Z, F)$ is obtained from $(V \cup Z', F')$ by contracting edges not belonging to $\binom{V \cup Z}{2}$.*

Proof. ($\Rightarrow$) Let $(P_f)_{f \in F}$ be a path-representation of $(V \cup Z, F)$ in G and consider the graph G' on vertex set $V(G)$ whose edges are precisely alll edges from P_f $(f \in F)$. Contracting all edges not belonging to $\binom{V \cup Z'}{2}$ and removing vertices other than those in $V \cup Z'$ we get a graph $(V \cup Z', F')$. Clearly F' satisfies Definition 7.2(a)–(c) and $(V \cup Z', F')$ is represented by a certain system of subpaths of $(P_f)_{f \in F}$. Hence F' is feasible and a contraction of $(V \cup Z', F')$ yields $(V \cup Z, F)$.

($\Rightarrow$) Let $(P'_f)_{f \in F'}$ be a path-representation of $(V \cup Z', F')$, let $f = \{v_1, v_2\} \in F$ be obtained by contracting a set $F_1 \subseteq F'$. Clearly there are edges $f_1, \ldots, f_n \in F_1$ forming a path from v_1 to v_2, let $f_1, \ldots, f_n$ be the order in which they appear on this path. Then

$$P'_{f_1} \cup \cdots \cup P'_{f_n}$$

is a path realizing f. Thus F is feasible. □

The straightforward proof of the following lemma is left to the reader.

Lemma 7.5. *Let F be a transition scheme of (G, V, E) relative to Z and let (G_1, G_2) be a Z-separation of G. Then F is feasible if and only if there exist sets F_1, F_2 such that*

$$F_i \subseteq \binom{V(G_i)}{2},$$

$F = F_1 \cup F_2 \cup E(G \mid Z)$, and F_i is a feasible transition scheme of

$$\left(G_i, V \cap V(G_i), E \cap \binom{V \cap V(G_i)}{2}\right)$$

relative to $Z \cap V(G_i)$.

Theorem 7.6. *There is an* $O(|E|^{k+1} \cdot |V(G)|)$ *algorithm to solve Problem* 7.3.

Proof. Let $P_{(V,E)}(G, Z)$ be the set of all feasible transition schemes of (G, V, E) relative to Z. Then Definition 3.2(a) is clearly satisfied with $T_0 = O(1)$ and by Lemmas 7.4 and 7.5, (b) and (c) in Definition on 3.2 are satisfied with $T_R = O(|E|^{k+1})$, $T_M = O(|E|^{k+1})$. (There are $O(|E|^{k+1})$ transition schemes to be tested and each can be done in constant time.) The result follows by Algorithm 3.3 and Theorem 3.4, since $P_{(V,E)}(G, \emptyset) = \{E\}$ if and only if (G, V, E) is feasible. □

Corollary 7.7. *Problem* 7.1 *can be solved in time* $O(n^{k+1} \mid V(G)|)$.

Added in proof. Recently B. Reed found an algorithm which, for every fixed k, finds a tree-decomposition of width k of a given n-vertex graph, provided that one exists, in $O(n \log n)$ time [announced at AMS Summer Research Conference on Graph Minors, Seattle, WA, June 1991].

References

[1] S. Arnborg, D.G. Corneil and A. Proskurowski, Complexity of finding embeddings in a k-tree, SIAM J. Algebraic Discrete Methods 8 (1987) 277–287.

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

[3] H.L. Bodlaender, Polynomial algorithm for graph is isomorphism and chromatic index on partial k-trees, Technical Report, University of Utrecht, Utrecht, 1988.

[4] M.R. Garey and D.S. Johnson, Computers and Intractability (Freeman, San Francisco, CA, 1979).

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

[6] M. Karp, On the complexity of combinatorial problems, Networks 5 (1975) 45–68.

[7] E. Luks, Isomorphism of graphs of bounded valence can be tested in polynomial time, in: Proc. 21st IEEE Symp. on Foundations of Computer Science (1980) 42–49.

[8] J. Matoušek and R. Thomas, Algorithms finding tree-decomposition of graphs, J. Algorithms 12 (1991) 1-22.

[9] N. Robertson and P.D. Seymour, Graph minors II. Algorithmic aspects of tree-width, J. Algorithms 7 (1986) 309–322.

[10] N. Robertson and P.D. Seymour, Graph minors XIII. The disjoint paths problem, Manuscript.