The Strong Dimension of Distance-Hereditary Graphs

Teresa R. May and Ortrud R. Oellermann *
The University of Winnipeg, 515 Portage Avenue
Winnipeg, MB R3B 2E9, CANADA

Abstract

Let G be a connected graph. A vertex r resolves a pair u, v of vertices of G if u and v are different distances from r. A set R of vertices of G is a resolving set for G if every pair of vertices of G is resolved by some vertex of R. The smallest cardinality of a resolving set is called the metric dimension of G. A vertex r strongly resolves a pair u, v of vertices of G if there is some shortest $u - r$ path that contains v or a shortest $v - r$ path that contains u. A set S of vertices of G is a strong resolving set for G if every pair of vertices of G is strongly resolved by some vertex of S; and the smallest cardinality of a strong resolving set of G is called the strong dimension of G. The problems of finding the metric dimension and strong dimension are NP-hard. Both the metric and strong dimension can be found efficiently for trees. In this paper, we present efficient solutions for finding the strong dimension of distance-hereditary graphs, a class of graphs that contains the trees.

Key words: metric dimension, strong dimension, distance-hereditary graphs
AMS Subject Classification Codes: 05C12, 05C85

1 Introduction

For graph theory terminology not defined here we follow [4]. A vertex r of a graph G *resolves* two vertices u and v of G if the distance, $d(r, u)$, from r to u does not equal the distance, $d(r, v)$, from r to v. A set R of vertices of G is a *resolving set* for G if for every pair u, v of vertices of G, there is some $r \in R$ that resolves u and v. The minimum cardinality of a resolving set for G is called the *metric dimension* of G and is denoted by $dim(G)$. A minimum resolving set is called a *metric basis* for G. So, for example, for the complete

*Supported by an NSERC Grant, Canada

graph K_n, $dim(K_n) = n - 1$; for the path P_n of order n, $dim(P_n) = 1$ and for the cycle C_n of order n, $dim(C_n) = 2$. Alternatively, an (ordered) set $R = \{r_1, r_2, \dots, r_k\}$ is a resolving set for G if for every two distinct vertices u and v of G, the two $k-vectors$ $r(v|R) = (d(v,r_1), d(v,r_2), \dots, d(v,r_k))$ and $r(u|R) = (d(u,r_1), d(u,r_2), \dots, d(u,r_k))$, called the *representations* of v and u with respect to R, are distinct.

Slater in [14] and [15] and independently Harary and Melter in [10] introduced and studied this in variant, although resolving sets in hypercubes were studied earlier under the guise of coin weighing problems (see [2]). Slater referred to the metric dimension of a graph as its location number and motivated the study of this invariant by its application to the placement of a minimum number of sonar/loran detecting devices in a network so that the position of every vertex can be uniquely described in terms of its distances to the devices in the set. It was noted in [7] that the problem of finding the metric dimension of a graph is NP-hard. Khuller, Raghavachari and Rosenfeld [12] gave a proof of this result. Their interest in this invariant was motivated by the navigation of robots in a graph space. A resolving set for a graph corresponds to the presence of distinctly labeled 'landmark' nodes in the graph. It is assumed that a robot navigating a graph can sense the distance to each of the landmarks, and thereby uniquely determine its location in the graph.

Several other applications of the metric dimension of a graph are discussed in [2] and [3] and an integer programming formulation is described and studied in [6].

A more restricted invariant than the metric dimension was introduced in [16]. The authors of this article considered the following problem:

Problem: Suppose H is an *isometric subgraph* of G, i.e., $d_H(u,v) = d_G(u,v)$ for all pairs of vertices u, v in H. Under what conditions does H allow us to determine all distances in G?

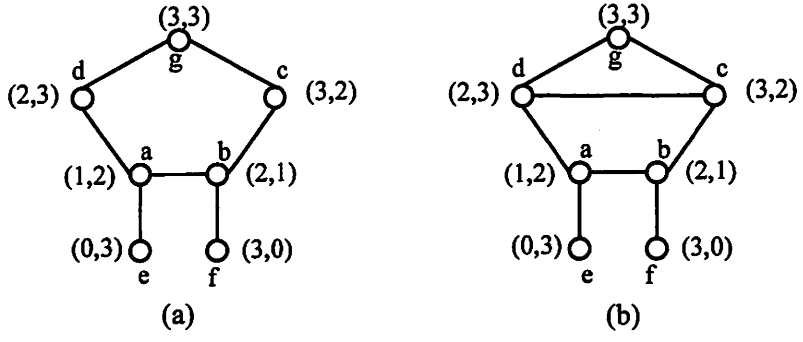

Figure 1: Nonisomorphic graphs with the same metric basis

Even though a metric basis uniquely determines the vertices of a graph it does not uniquely determine the graph. For example, the graphs of Fig 1(a) and Fig 1(b) both have metric basis $B = \{e, f\}$ and corresponding vertices have the same 2-vectors with respect to B but these graphs are not isomorphic.

The following definitions give rise to sets of vertices that are sufficient to guarantee the sought after uniqueness. A *vertex* r *strongly resolves* a pair u, v of vertices in a connected graph G if there is a shortest $u - r$ path that contains v or a shortest $v - r$ path that contains u. A *set* S of vertices of G *strongly resolves* G if every pair of vertices is strongly resolved by some vertex of S. The smallest cardinality of a strong resolving set of G is called the *strong dimension* of G and is denoted by $sdim(G)$. Clearly if a vertex r strongly resolves u and v of G, then r resolves u and v. Hence $dim(G) \leq sdim(G)$. It was shown in [13] that the problem of finding the strong dimension of a graph is NP-hard.

Both the metric dimension and strong dimension can be found efficiently for trees. Indeed it is not difficult to see that the strong dimension of a non-trivial tree is one less than the number of leaves. Efficient procedures for finding the metric dimension of trees were described independently in [3], [10], [12], and [14]. The metric dimension of a path is 1 since a leaf resolves the path. Suppose now that T contains a vertex of degree at least 3. A vertex v of degree at least 3 is an *exterior vertex* if there is some leaf u in T such that the $u - v$ path in T contains no vertices of degree exceeding 2 except v. In that case u is an *exterior leaf* of v. Let $ex(T)$ denote the number of exterior vertices of T and $l(T)$ the number of leaves (vertices of degree 1) of T. It turns out that a metric basis for a tree can be found by selecting, for each exterior vertex, all but one of its exterior leaves. Hence

$$dim(T) = l(T) - ex(T).$$

1.1 Distance-Hereditary Graphs

In this paper we develop an efficient algorithm for finding the strong dimension of distance-hereditary graphs, a class of perfect graphs that properly contains all trees. A graph G is *distance-hereditary* if every connected induced subgraph H of G is isometric, i.e., if for every pair $u, v \in V(H)$, $d_H(u, v) = d_G(u, v)$. Howorka [11] first defined and studied these graphs.

Several polynomial recognition algorithms for distance-hereditary graphs exist. To describe one of these that we will use we define two vertices v and v' to be *true (false) twins* if they have the same closed (respectively, open) neighbourhood. (The *open neighbourhood* of a vertex v is defined as $N(v) = \{u | uv \in E(G)\}$ and the *closed neighbourhood* of v is defined by $N[v] = N(v) \cup \{v\}$.) Vertices that are either true or false twins will be referred to as *twins*. False twins that are not leaves will be referred to as *proper false twins*. The following characterization of distance-hereditary graphs was discovered independently in [1] and [9].

Theorem 1.1 *A graph G is distance-hereditary if and only if every induced subgraph of G contains an isolated vertex, a leaf, or a pair of twins.*

Suppose G is a graph and that v' is some vertex of G. If we add a new vertex v to G and join it to:

1. only v', we say v was *added as a leaf to* v';
2. every vertex in the open neighbourhood of v', we say v was *added as a false twin of* v';
3. every vertex in the closed neighbourhood of v', we say v was *added as a true twin of* v'.

From Theorem 1.1, it follows that a connected graph G of order n is distance-hereditary if and only if there is a sequence of graphs $G_2, G_3, \ldots, G_n$ such that
(i) $G_2 \cong K_2$, $G_n \cong G$ and
(ii) for $i = 3, 4, \ldots, n$, G_i is obtained from G_{i-1} by adding some vertex v_i as a leaf, true twin or false twin of some vertex v_i' in G_{i-1}.

Let v_1 and v_2 be the vertices of G_2 and for $i \geq 3$, let v_i be as described in (ii). Then we call $v_1, v_2, \ldots, v_n$ a *DH sequence* of G.

Another useful characterization of distance-hereditary graphs due to Howorka [11] states:

Theorem 1.2 *A graph G is distance-hereditary if and only if every cycle of length at least* 5 *contains a pair of crossing chords.*

1.2 Background on the Strong Dimension

It was shown in [13] that the problem of finding the strong dimension of a graph can be transformed to the vertex covering problem. A *vertex cover* of a graph is a set S of vertices of G such that every edge of G is incident with at least one vertex of S. The *vertex covering number* of G, denoted by $\alpha(G)$, is the smallest cardinality of a vertex cover of G. We say a vertex u is *maximally distant* from v (denoted by $u\ MD\ v$) if for every $w \in N(u)$, $d(v, w) \leq d(u, v)$. If u is maximally distant from v and v is maximally distant from u, we say that u and v are *mutually maximally distant* and denote this by $u\ MMD\ v$.

Let G be a connected graph. Then the *strong resolving graph* G_{SR} of G has the same vertex set as G and $uv \in E(G_{SR})$ if and only if $u\ MMD\ v$. The following result was established in [13].

Theorem 1.3 *For any connected graph G, $sdim(G) = \alpha(G_{SR})$. Moreover, a minimum vertex cover of G_{SR} is a minimum strong resolving set of G.*

We use this result when developing an efficient algorithm for finding the strong dimension and a strong resolving graph for G.

2 Algorithm for Finding the Strong Dimension of a Distance-Hereditary Graph

Let G be a distance-hereditary graph of order n. Let $v_1, v_2, \ldots, v_n$ be a DH sequence for G and let $G_i = \langle\{v_1, v_2, \ldots, v_i\}\rangle$ for $1 \leq i \leq n$. Then v_i is a leaf, proper false twin or a true twin of some vertex v_j in G_i $1 \leq j < i$.

To describe the algorithm, we will distinguish five types of vertices in a distance-hereditary graph:

1. leaf
2. proper false twin
3. true twin
4. cut-vertex
5. other (if it does not belong to any of the above four types)

The algorithm begins with G_2, which necessarily is isomorphic to K_2. It proceeds by adding at each step the next vertex in the DH sequence until all vertices are added. After the addition of the next vertex in the DH sequence, the strong resolving graph G_{SR} is modified and the minimum strong resolving set S_{SR} is modified. The algorithm also keeps track of the vertex type of each vertex after each step is completed. Moreover, if a vertex belongs to a set of proper false twins or true twins, then the algorithm keeps record of these sets.

Algorithm (for finding the strong dimension of a distance-hereditary graph $G = (V, E)$ with vertex set V and edge set E.)

1. Initially G_{SR} consists of v_1, v_2 and the edge v_1v_2 and $S_{SR} = \{v_1\}$. Assign type leaf to both v_1 and v_2.
2. For $i = 3, \ldots, n$
   (a) If v_i is a leaf in G_i with neighbour v_i' then assign type leaf to v_i and type cut-vertex to v_i'.
      i. If v_i' is of type leaf in G_{i-1}, then add v_i to G_{SR} and for every u in G_{SR} such that $uv_i' \in E(G_{SR})$ delete uv_i' and add uv_i. If $v_i' \in S_{SR}$, let $S_{SR} \leftarrow (S_{SR} \setminus \{v_i'\}) \cup \{v_i\}$; otherwise, S_{SR} is unchanged.
      ii. If v_i' is of type proper false twin or a true twin in G_{i-1}, let $S_{v_i'}$ be the collection of twins to which v_i' belongs in G_{i-1}. If $|S_{v_i'}| = 2$, then the vertex of $S_{v_i'} \setminus \{v_i'\}$ is assigned type other; otherwise, the vertices of $S_{v_i'} \setminus \{v_i'\}$ retain their proper false twin status and $S_{v_i'} \setminus \{v_i'\}$ is the set of proper false twins for each of the vertices contained in it. Determine a BFS tree T rooted at v_i in G_i. Add v_i to $V(G_{SR})$. For every leaf $u \neq v_i$ in T such that no neighbour of u in G_i is further from v_i than u, add uv_i to $E(G_{SR})$. For every vertex w in G_{SR} such that $wv_i' \in E(G_{SR})$ delete wv_i' from G_{SR}. Let G_{SR} be the resulting graph. If $v_i' \in S_{SR}$, let $S_{SR} \leftarrow (S_{SR} \setminus \{v_i'\}) \cup \{v_i\}$. If $v_i' \notin S_{SR}$, let $u \in S_{v_i'} \setminus \{v_i'\}$ and let $S_{SR} \leftarrow (S_{SR} \setminus \{u\}) \cup \{v_i\}$.
      iii. If v_i' is of type cut-vertex in G_{i-1}, determine a BFS tree T rooted at v_i in G_i. Add v_i to G_{SR} and for every leaf $u \neq v_i$ of T such that no neighbour of u in G_i is further from v_i than u, add uv_i to $E(G_{SR})$. If every neighbour of v_i in G_{SR} belongs to S_{SR}, then S_{SR} is unchanged; otherwise, $S_{SR} \leftarrow S_{SR} \cup \{v_i\}$.

iv. If v_i' is of type other in G_{i-1}, construct a BFS tree T rooted at v_i in G_i. For every u in G_{SR} such that $uv_i' \in E(G_{SR})$ (if any) delete uv_i' from G_{SR}. Add v_i to $V(G_{SR})$. If u is a leaf of T such that no neighbour of u in G_i is further from v_i than u add uv_i to $E(G_{SR})$. If $v_i' \in S_{SR}$, $S_{SR} \leftarrow (S_{SR} \setminus \{v_i'\}) \cup \{v_i\}$. If $v_i' \notin S_{SR}$, and every neighbour of v_i in G_{SR} belongs to S_{SR}, then S_{SR} is unchanged; otherwise, $S_{SR} \leftarrow S_{SR} \cup \{v_i\}$.

(b) If v_i is a true twin of v_i' in G_i, then both v_i and v_i' are assigned type true twin.

i. If v_i' is a leaf, then add v_i to G_{SR} and join v_i to v_i', as well as to all the neighbours of v_i' in G_{SR} and let G_{SR} be the resulting graph. Let $S_{SR} \leftarrow S_{SR} \cup \{v_i\}$. Let $\{v_i, v_i'\}$ be the set of true twins for v_i and v_i'.

ii. If v_i' is a proper false twin, then let $S_{v_i'}$ be the collection of false twins of v_i' in G_{i-1}. If $|S_{v_i'}| = 2$, then the vertex of $S_{v_i'} \setminus \{v_i'\}$ is assigned type other; otherwise, the vertices of $S_{v_i'} \setminus \{v_i'\}$ retain their false twin status and $S_{v_i'} \setminus \{v_i'\}$ is the set of proper false twins for each of the vertices contained in it. Let $\{v_i', v_i\}$ be the set of true twins for v_i and v_i'. Add v_i to G_{SR} and join v_i to v_i' and to all the neighbours of v_i' in G_{SR}. Let $S_{SR} \leftarrow S_{SR} \cup \{v_i\}$.

iii. If v_i' is a true twin, then add v_i to G_{SR} and join v_i to v_i' and all the neighbours of v_i' in G_{SR}. Let $S_{SR} \leftarrow S_{SR} \cup \{v_i\}$. Add v_i to the set of true twins that contains v_i'.

iv. If v_i' is a cut-vertex, then add v_i to G_{SR} and join v_i to v_i' and let $S_{SR} \leftarrow S_{SR} \cup \{v_i\}$. Let $\{v_i, v_i'\}$ be the set of true twins for v_i and v_i'. If v_i' is adjacent with exactly one leaf in G_{i-1}, then that leaf is assigned type other. If v_i' is adjacent with two or more leaves in G_{i-1}, then those leaves become a set of proper false twins.

v. If v_i' is of type other, then add v_i to G_{SR} and join v_i to v_i' and to every neighbour of v_i' in G_{SR} (if any). Let $\{v_i, v_i'\}$ be the true twin set for v_i and v_i'. Let $S_{SR} \leftarrow S_{SR} \cup \{v_i\}$.

(c) If v_i is a proper false twin of v_i' in G_i, then both v_i and v_i' are assigned type proper false twin. (Note that in this case v_i' is not a leaf.)

i. If v_i' is a proper false twin, add v_i to G_{SR} and join v_i to v_i' and all the neighbours of v_i' in G_{SR}. Let G_{SR} be the resulting graph. Let $S_{SR} \leftarrow S_{SR} \cup \{v_i\}$ and add v_i to the set of twins containing v_i'.

ii. If v_i' is a true twin in G_{i-1}, then let $S_{v_i'}$ be the collection of true twins in G_{i-1} to which v_i' belongs. If $|S_{v_i'}| = 2$, then the vertex of $S_{v_i'} \setminus \{v_i'\}$ is assigned type other; otherwise, the vertices of $S_{v_i'}$ retain their true twin status and $S_{v_i'} \setminus \{v_i'\}$ is the set of true twins for each of the vertices contained in it. Let $\{v_i, v_i'\}$ be the set of proper false twins for v_i and v_i'. Add v_i to G_{SR} and join v_i to v_i' and all neighbours of v_i' in G_{SR} that do not belong to $S_{v_i'}$. For

every $u \in S_{v_i'} \setminus \{v_i'\}$, delete the edge uv_i' from G_{SR}. Let G_{SR} be the resulting graph.
Suppose $N_{G_{SR}}(v_i) \setminus \{v_i'\} \subseteq S_{SR}$. If $v_i' \in S_{SR}$, then S_{SR} remains unchanged. If $v_i' \notin S_{SR}$, let $u \in S_{v_i'} \setminus \{v_i'\}$ and let $S_{SR} \leftarrow (S_{SR} \setminus \{u\}) \cup \{v_i\}$.
Suppose $N_{G_{SR}}(v_i) \setminus \{v_i'\} \not\subseteq S_{SR}$. Then $N_{G_{SR}}(v_i) \setminus \{v_i'\} \neq \emptyset$. If $|N_{G_{SR}}(v_i) \setminus S_{SR}| \geq 2$, then $S_{SR} \leftarrow S_{SR} \cup \{v_i\}$. If $|N_{G_{SR}}(v_i) \setminus S_{SR}| = 1$, let $u \in S_{v_i'} \setminus \{v_i'\}$ and $S_{SR} \leftarrow (S_{SR} \cup N_{G_{SR}}(v_i)) \setminus \{u\}$.

iii. If v_i' is a cut-vertex in G_{i-1}, add v_i to G_{SR} and join v_i to v_i' by an edge and let $S_{SR} \leftarrow S_{SR} \cup \{v_i\}$. If v_i' is adjacent with exactly one leaf in G_{i-1}, then that leaf is assigned type other. If v_i' is adjacent with two or more leaves in G_{i-1}, then those leaves become a set of proper false twins.

iv. If v_i' is of type other, add v_i to G_{SR} and join v_i to v_i' by an edge and for every neighbour u of v_i' in G_{SR} add the edge v_iu to G_{SR}. Let $S_{SR} \leftarrow S_{SR} \cup \{v_i\}$.

3. Output S_{SR} as it is a minimum strong resolving set of G.

Theorem 2.1 *The set S_{SR} output by the Algorithm is a minimum strong resolving set for $G = (V, E)$.*

Proof: To simplify the proof, we will let G_{iSR} be the 'strong resolving' graph of G_i constructed by the algorithm and S_{iSR} the 'strong resolving' set for G_i constructed by the algorithm for $2 \leq i \leq n$.

We proceed by induction on $i \geq 2$ to show that after v_i is added to G_{i-1} to produce G_i, the algorithm correctly modifies S_{SR} so that it is a minimum strong resolving set for G_i. Moreover, G_{SR} is correctly modified to be the strong resolving graph of G_i and the vertex types are correctly described.

For $i = 2$ the algorithm correctly specifies a minimum strong resolving set of G_2 and correctly describes the strong resolving graph G_{2SR} for G_2 and the vertex types of v_1 and v_2 in G_2.

Suppose now that $i \geq 3$ and that the algorithm correctly determines a minimum strong resolving set, namely $S_{(i-1)SR}$, for G_{i-1}, that $G_{(i-1)SR}$ correctly describes the strong resolving graph of G_{i-1}, and that the algorithm correctly describes the vertex types of all vertices in G_{i-1}.

Case 1: v_i is added as a leaf to v_i'.

Clearly v_i's type is a leaf and v_i' is a cut-vertex. Moreover, if v_i' has exactly one proper false twin or exactly one true twin, then that twin is of type other in G_i and the vertex types of all other vertices remain unchanged. Since G_{i-1} and G_i are connected induced subgraphs of a distance-hereditary graph, $d_{G_{i-1}}(x, y) = d_G(x, y)$ for all $x, y \in V(G_{i-1})$. Moreover, if $x, y \in V(G_{i-1}) \setminus \{v_i'\}$, then $x\ MMD\ y$ in G_{i-1} if and only if $x\ MMD\ y$ in G_i. Since, for all $u \in V(G_{i-1})$, $d_{G_i}(v_i, u) = d_{G_i}(v_i', u) + 1 = d_{G_{i-1}}(v_i', u) + 1$, v_i' is not incident with any edges in the strong resolving graph for G_i and since v_i is maximally distant from all vertices of G_{i-1}, it is mutually maximally distant with those vertices that are maximally distant from v_i in G_i. If v_i' is a leaf, these vertices are precisely the neighbours of v_i' in the strong resolving graph $G_{(i-1)SR}$ for G_{i-1}. If v_i' is a twin, cut-vertex or of type other in G_{i-1}, the vertices maximally distant from v_i can be determined using a BFS tree rooted at v_i in G_i. Hence G_{iSR} is the strong resolving graph of G_i in this case. It remains to be shown that S_{iSR} is a minimum strong resolving set for G_i.

Subcase 1.1: Suppose v_i' is a leaf in G_{i-1}. Note that v_i' is isolated in the strong resolving graph for G_i. Moreover, v_i MMD u in G_i if and only if v_i' MMD u in G_{i-1}. So the neighbours of v_i' in $G_{(i-1)SR}$ are precisely the neighbours of v_i in G_{iSR}. Hence, $G_{iSR} \cong G_{(i-1)SR} \cup K_1$, and thus a minimum vertex cover S_{iSR} of G_{iSR} can be obtained from a minimum vertex cover of $G_{(i-1)SR}$, namely $S_{(i-1)SR}$, by either replacing v_i' with v_i in $S_{(i-1)SR}$ if $v_i' \in S_{(i-1)SR}$ or by letting $S_{iSR} = S_{(i-1)SR}$ if $v_i' \notin S_{(i-1)SR}$. By Theorem 1.3, S_{iSR} is thus a minimum strong resolving set for G_i.

Subcase 1.2: Suppose v_i' is a proper false twin or a true twin in G_{i-1}. Then v_i' is no longer a twin in G_i. The neighbours of v_i in G_{iSR} contain as a subset the neighbours of v_i' in $G_{(i-1)SR}$. From an earlier observation, $G_{(i-1)SR} - v_i' = G_{iSR} \setminus \{v_i, v_i'\}$. By assumption, $S_{(i-1)SR}$ is a minimum vertex cover of G_{i-1}, and by the algorithm, v_i' is isolated in G_{iSR}. From these observations, it follows, if $S_{(i-1)SR}$ contains v_i', that $S_{iSR} = (S_{(i-1)SR} \setminus \{v_i'\}) \cup \{v_i\}$ must be a minimum vertex cover of G_{iSR}. Hence, by Theorem 1.3, S_{iSR} is a minimum strong resolving set for G_i. If $v_i' \notin S_{(i-1)SR}$, then necessarily all the twins of v_i' in G_{i-1} belong to $S_{(i-1)SR}$ since twins of G_{i-1} are pairwise adjacent in $G_{(i-1)SR}$.

If u is a twin of v_i', then $S' = (S_{(i-1)SR} \setminus \{u\}) \cup \{v_i'\}$ is still a vertex cover of $G_{(i-1)SR}$, since u and v_i have the same closed neighbourhood in $G_{(i-1)SR}$. So, as in the previous situation, $(S' \setminus \{v_i'\}) \cup \{v_i\} = (S_{(i-1)SR} \setminus \{u\}) \cup \{v_i\}$ is a minimum strong resolving set of G_i.

Subcase 1.3: Suppose v_i' is a cut-vertex of G_{i-1}. Then v_i' is isolated in $G_{(i-1)SR}$ and G_{iSR}. So all the edges of $G_{(i-1)SR}$ are also edges of G_{iSR}. In addition, v_i is incident with some edges of G_{iSR}. Clearly, if all neighbours of v_i in G_{iSR} are in $S_{(i-1)SR}$, then $S_{(i-1)SR}$ is a minimum vertex cover of G_{iSR}.

Suppose now that some neighbour of v_i in G_{iSR} does not belong to $S_{(i-1)SR}$. We will show that $S_{(i-1)SR} \cup \{v_i\}$ is a minimum cover for G_{iSR}.

The set $S_{(i-1)SR} \cup \{v_i\}$ is certainly a vertex cover for G_{iSR}. If it is not a minimum cover, then there is a minimum vertex cover T_{i-1} of $G_{(i-1)SR}$ that contains all the neighbours of v_i in G_{iSR}. Since T_{i-1} is a minimum vertex cover of $G_{(i-1)SR}$, each $x \in T_{i-1}$ has a neighbour x' in $G_{(i-1)SR}$ that is not adjacent with v_i in G_{iSR}. So x' is not maximally distant from v_i' in G_{i-1}. (Note x and x' belong to the same component of $G_{i-1} - v_i'$.) Hence x' has a neighbour y_1 in G_{i-1} such that $d_{G_{i-1}}(y_1, v_i',) > d_{G_{i-1}}(x', v_i')$. If x' is not maximally distant from v_i' in G_{i-1}, it has a neighbour y_2 such that $d_{G_{i-1}}(y_2, v_i') > d_{G_{i-1}}(y_1, v_i')$. Let $x' = y_0$. We continue in this manner constructing a sequence $y_0, y_1, y_2, \ldots$ of distinct vertices such that $d_{G_{i-1}}(y_j, v_i') < d_{G_{i-1}}(y_{j+1}, v_i')$ for $j \geq 0$. This sequence must terminate with some y_s such that y_s is maximally distant from v_i'.

Note that if H is any $x' - v_i'$ geodesic, then $y_j y_{j-1} \ldots y_1 H$ is a $y_j - v_i'$ geodesic for $1 \leq j \leq s$. Since $x' = y_0$ is maximally distant from x in G_{i-1}, x' cannot lie on any $v_i' - x$ geodesic in G_{i-1}. Hence y_j can also not belong to any $x - v_i'$ geodesic for $1 \leq j \leq s$. Let $Q : (v_i' =)u_0u_1 \ldots u_d(= x)$ be a $v_i' - x$ geodesic in G_{i-1}. Let $P : (v_i' =)w_0w_1 \ldots w_k(= x')$ be a $v_i' - x'$ geodesic in G_{i-1} that has a maximum number of vertices in common with Q. Let $l \geq 1$ be the smallest integer such that w_l does not belong to Q. Since $x' = w_k$ is not on Q, such an l exists. Also by our choice of P, $u_j = w_j$ for $0 \leq j < l$. We now show that $l = k$. Observe that the subgraph induced by the vertices on the path $X : (x =)u_du_{d-1} \ldots u_{l-1}w_lw_{l+1} \ldots w_k(= x')y_1$ must contain a $x - y_1$ geodesic. Since x' is MD from x, X is not a $x - y_1$ geodesic.

Since $P' : w_{l-1}w_l \ldots w_ky_1$ and $Q' : u_{l-1}u_l \ldots u_d(= x)$ are both geodesics, the only edges of $\langle V(X) \rangle$ that are not on X must join vertices of $P' - w_{l-1}$ and $Q' - u_{l-1}$. For an integer j ($l \leq j < min\{d, k+1\}$), the only vertices of Q' to which w_j can possibly be adjacent are u_{j+1}, u_j and u_{j-1} (since $d_{G_{i-1}}(v_i', u_j) = d_{G_{i-1}}(v_i', w_j)$). But if w_ju_{j-1} is an edge, we have a contradiction to our choice of P unless $j = l$. Also $x' = w_k$ is not adjacent with u_{k+1} (if $k+1 \leq d$); otherwise, x' lies on an $x - v_i'$ geodesic which is not possible. So the only vertex of $Q' - \{u_{l-1}\}$ to which x' is possibly adjacent is u_k. Since x' is on no $x - v_i'$ geodesic, the only vertices to which y_1 is possibly adjacent are u_{k+1} and u_k. If $y_1u_{k+1} \in E$, then $\langle \{u_{l-1}, u_l, \ldots, u_{k+1}\} \cup \{w_l, \ldots, w_k, y_1\} \rangle$ contains a cycle of length at least five without crossing chords; this is not possible in a distance-hereditary graph. Hence $y_1u_{k+1} \notin E$ and the only vertex of Q' to which x' may be adjacent is u_k. Since $d_{G_{i-1}}(y_1, x) \leq d_{G_{i-1}}(x', x)$ it is necessarily the case that $y_1u_k \in E$. So $u_k \neq x$; otherwise, x is not maximally distant from v_i'. Hence $k < d$. Also $Y : w_lw_{l+1} \ldots w_ky_1u_ku_{k-1} \ldots u_{l-1}w_l\}$ is a cycle of length at least four without crossing chords. Since G_{i-1} is distance-hereditary, Y must contain exactly four vertices, namely y_1, x', w_{k-1}, u_k. Hence $l = k$.

Observe that for $2 \leq j \leq s$, y_j is not adjacent with any vertex of Q', otherwise if y_j is adjacent with a vertex of Q' then it is adjacent with one of u_{k+j}, u_{k+j+1} or u_{k+j-1}. But then it is not difficult to see that G_{i-1} contains a 5-cycle without crossing chords. Hence the following is an induced subgraph of G_{i-1}.

We know that for every vertex x that is adjacent with v_i in G_{iSR} there is a vertex x' not in $N_{G_{iSR}}[v_i] \setminus \{x\}$ adjacent with x in G_{iSR} that is not in T_{i-1}. Among all such pairs (x, x') of vertices in G_i let (x_1, x_1') be a pair for which $d(v_i', x_1')$ is as large as possible. Suppose $(v_i' =)u_0u_1 \ldots u_d(= x_1)$ is a shortest $v_i' - x_1$ path. By the previous observation there is a vertex x_{11} adjacent with x_1' such that $d(x_{11}, v_i') > d(x_1', v_i')$. Moreover, if $k = d(v_i', x_1')$, then $1 \leq k < d$ and $x_1'u_{k-1}$, $x_{11}u_k \in E$. If x_{11} is not maximally distant from v_i' in G_{i-1}, it has a neighbour x_{12} such that $d(x_{12}, v_i') > d(x_{11}, v_i')$. We continue in this manner constructing a sequence $(x_1' =)x_{10}, x_{11}, x_{12}, \ldots$ of distinct vertices such that $d_{G_{i-1}}(x_{1j}, v_i') < d_{G_{i-1}}(x_{1(j+1)}, v_i')$ for $j \geq 0$. This sequence will terminate with some x_{1s_1} where x_{1s_1} is MD from v_i'. Also the paths $P_1' : x_1'x_{11} \ldots x_{1s_1}$ and $P_0' : u_{k-1}u_k \ldots u_d(= x)$ are internally disjoint and the only edges from vertices of P_1' to vertices of P_0' are $x_{11}u_k, x_1'u_{k-1}$ and possibly $x_1'u_k$.

Let $x_2 = x_{1s_1}$. Clearly $x_2 \neq x'$. From an earlier observation, there exists a vertex z that is MMD from x_2 but not MD from v_i'. Among all such vertices z let x_2' be one furthest from v_i'. So there is a neighbour x_{21} of x_2' such that $d(x_{21}, v_i') >$

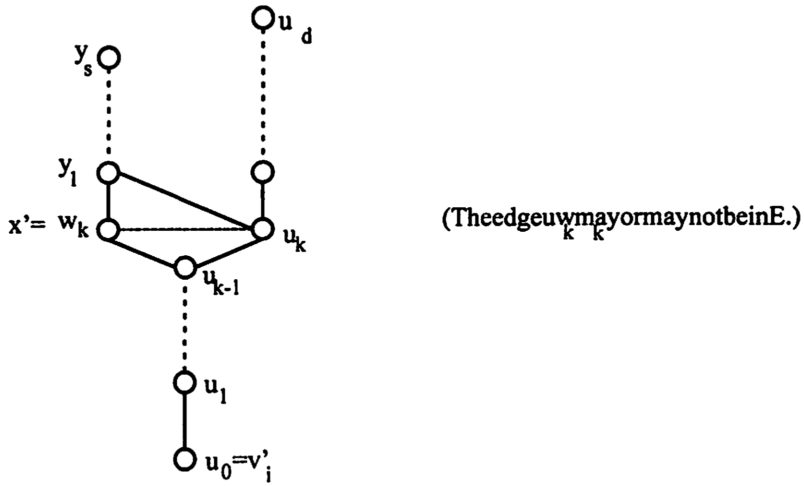

Figure 2: A configuration in Subcase 1.3

$d(x_2', v_i')$. Continuing in this manner we construct a sequence $x_{21}, x_{22} \ldots$ such that $d_{G_{i-1}}(v_i', x_{2j}) < d_{G_{i-1}}(v_i', x_{2(j+1)})$. This sequence must terminate with some x_{2s_2}. It is not difficult to see that the path $P_2' : (x_2' =)x_{20}x_{21} \ldots x_{2s_2}$ is internally disjoint from both $u_{k-1}P_1'$ and the path $P_1'' : u_{k-1}u_kx_{11}x_{12} \ldots x_{1s_1}$. Using the $v_i' - x_2$ geodesic $(v_i' =)u_0u_1 \ldots u_{k-1}u_kx_{11}x_{12} \ldots x_{1s_1}(= x_2)$, it follows as for x_1, x_1' and x_{11} and by the choice of the pair (x_1, x_1') that there is some j $(1 \le j \le k)$ such that $x_2'u_{j-1}$, $x_{21}u_j$ and possibly $x_2'u_j$ are edges and $d(x_2', v_i') = d(u_j, v_i')$. Apart from the edges $x_2'u_{j-1}, x_{21u_j}$ and possibly $x_2'u_j$ there are no edges between vertices of P_2' and those on $u_{j-1}u_j \ldots u_{k-1}P_1''$ or $u_{j-1}u_j \ldots u_{k-1}P_1'$.

Suppose first that $j = k$. As before we see that P_2' is vertex disjoint from the two geodesics $u_0u_1 \ldots u_{k-1}x_1'x_{11} \ldots x_{1s_1}$ and $u_0u_1 \ldots u_{k-1}u_kx_{11} \ldots x_{1s_1}$. Moreover, $x_2'u_{k-1}$ and $x_{21}x_1'$ and possibly $x_2'x_1'$ are the only edges joining vertices of P_2' and the first of these two geodesics and $x_2'u_{k-1}$, $x_{21}u_k$ and possibly $x_2'u_k$ are the only edges joining vertices of the second $u_0 - x_{1s_1}$ geodesic (see Fig 3).

We now show that P_2' is vertex disjoint from $u_{k+1} \ldots u_d$. First of all observe that $x_{21} \ne u_{k+1}$, since $x_{21}x_1' \in E$ but $u_{k+1}x_1' \notin E$. We show next that $u_{k+1}x_2' \notin E$. If not, then $x_2'u_{k+1}u_kx_{11}x_1'u_{k-1}x_2'$ is a 6-cycle without chords unless $x_1'x_2' \in E$. However, then $x_2'u_{k+1}u_kx_{11}x_1'x_2'$ is a 5-cycle without crossing chords. So $x_2'u_{k+1} \notin E$. Now it follow that $x_{21}u_{k+1} \notin E$; otherwise, $x_{21}u_{k+1}u_ku_{k-1}x_2'x_{21}$ is a 5-cycle without crossing chords. It is now not difficult to see that P_2' does not intersect the path $u_{k+1}u_{k+2} \ldots u_d$; otherwise, G contains a cycle of length at least 5 without crossing chords. Moreover, no vertex of P_2' is adjacent with any vertex of $u_{k+1}u_{k+2} \ldots u_d$. So $x_{2s_2} \ne x_1$ or x_2.

If $j \le k-1$, then one can argue as before that P_2' is internally disjoint from the

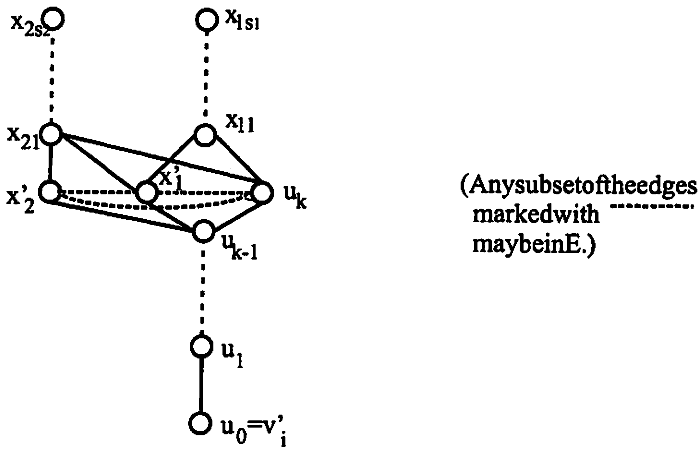

Figure 3: Another configuration in Subcase 1.3

path $u_{j-1}u_j \dots u_{k-1}P_1'$ and from the path $u_{j-1}u_j \dots u_{k-1}P_1''$ and the only edges between vertices of P_2' and those on $u_{j-1}u_j \dots u_{k-1}P_1'$ or $u_{j-1}u_j \dots u_{k-1}P_1''$ are $x_{21}u_j$ and $x_{20}u_{j-1}$ and possibly $x_2'u_j$. Also one can argue since G_i is distance-hereditary that the geodesics $u_{j-1}u_j \dots u_d$ and $x_2'x_{21}x_{22} \dots x_{2s_2}$ are disjoint and that the only edges between vertices from these two geodesics are $x_2'u_{j-1}$ $x_{21}u_j$ and possibly $x_2'u_j$. So $x_{2s_2} \neq x_1$ or x_2.

Let $x_3 = x_{2s_2}$. Hence there exists a vertex $x_3' \notin N_{G_{iSR}}[v_i]$ that is MMD with x_3 but that is not MD from v_i'. So there is a neighbour x_{31} of x_3' in G such that $d_G(v_i', x_{31}) > d_G(v_i', x_3')$. Again we construct a finite sequence $x_{31}, x_{32}, \dots, x_{3s_3}$ such that x_{3s_3} is MD from v_i'. Let $x_4 = x_{3s_3}$. We can argue as before that x_4 is not equal to x_1, x_2, or x_3. Continuing in this manner we construct an infinite sequence $x_1, x_2, \dots$ of distinct vertices of G which is not possible as G_i is a finite graph.

Subcase 1.4: Suppose v_i' is of type other in G_{i-1}. It can be argued similarly as in Case 1.3, that the set S_{iSR} constructed by the algorithm is indeed a minimum resolving set of G_i.

Case 2: v_i is added as a true twin of v_i' in G_{i-1}.

Clearly v_i and v_i' are both of type true twin. As in Case 1, we see that if $x, y \in V(G_{i-1}) \setminus \{v_i'\}$, then x MMD y in G_{i-1} if and only if x MMD y in G_i. Since for all $u \in V(G_{i-1}) \setminus \{v_i'\}$, $d_{G_i}(v_i, u) = d_{G_i}(v_i', u) = d_{G_{i-1}}(v_i', u)$, v_i MMD u if and only if v_i' MMD u. Moreover, v_i is MMD with v_i' in G_i. Thus v_i and v_i' have the same closed neighbourhood in G_{iSR} and this closed neighbourhood consists of v_i, v_i' and all the neighbours of v_i' in $G_{(i-1)SR}$. Hence G_{iSR} is constructed correctly and contains G_{iSR} as an induced subgraph.

We now show that S_{iSR} is a minimum strong resolving set for G_i. Note that in all cases the algorithm lets $S_{iSR} \leftarrow S_{(i-1)SR} \cup \{v_i\}$. By the inductive hypothesis $S_{(i-1)SR}$ is a minimum resolving set for G_{i-1}, i.e., a minimum vertex cover of G_{i-1}. If v_i' is an isolated vertex in $G_{(i-1)SR}$, i.e., if v_i' is not MMD with any vertex in G_{i-1}, then $v_i v_i'$ is the only edge incident with v_i' in G_{iSR} and $S_{(i-1)SR} \cup \{v_i\}$ is thus a minimum vertex cover of G_{iSR} and hence a minimum strong resolving set of G_i. Suppose now that v_i' is not isolated in $G_{(i-1)SR}$. Then either v_i' or all its neighbours in G_{iSR} belong to $S_{(i-1)SR}$ but not both. In order to cover all the edges incident with v_i in G_{iSR} we must necessarily add one more vertex to $S_{(i-1)SR}$ to obtain a minimum vertex cover of G_{iSR}, and adding v_i to $S_{(i-1)SR}$ produces a (minimum) vertex cover for G_{iSR}.

Case 3: v_i is added as a proper false twin to v_i' of G_{i-1} (i.e. v_i is a false twin of v_i' that has degree at least 2).

Clearly v_i and v_i' are correctly assigned type false twin. As in Cases 1 and 2, we see that if $x, y \in V(G_{i-1}) \setminus \{v_i'\}$ then $x\ MMD\ y$ in G_{i-1} if and only if $x\ MMD\ y$ in G_i. Moreover, since for every $u \in V(G_{i-1}) \setminus \{v_i'\}$, $d_{G_i}(v_i, u) = d_{G_i}(v_i', u) = d_{G_{i-1}}(v_i', u)$, $v_i\ MMD\ u$ in G_i if and only if $v_i'\ MMD\ u$. Also $v_i\ MMD\ v_i'$ in G_i. For the remainder of the proof, we consider several cases depending on v_i''s type in G_{i-1}. From the case we are in $deg_{G_{i-1}} v_i' \geq 2$. Hence v_i' is not a leaf in G_{i-1}.

Subcase 3.1: Suppose v_i' is a proper false twin.

Since v_i is added as a proper false twin to v_i', it needs to be added to the set of proper false twins to which v_i' belongs. So $N_{G_i}(v_i) = N_{G_i}(v_i') = N_{G_{i-1}}(v_i') = N_{G_{i-1}}(u)$ for all u in the set of proper false twins containing v_i' in G_{i-1}. So $v_i\ MMD\ v_i'$ in G_i and for $x \in V(G_{i-1}) \setminus \{v_i'\}$, $v_i'\ MMD\ x$ in G_{i-1} if and only if $v_i'\ MMD\ x$ in G_i if and only if $v_i\ MMD\ x$ in G_i. Thus G_{iSR} is constructed correctly in this case.

Let $S_{v_i'}$ be the collection of false twins of v_i' in G_{i-1}. Then $S_{v_i} = S_{v_i'} \cup \{v_i\}$ is the collection of false twins of v_i in G_i. Then the subgraph induced by the vertices of S_{v_i} in G_{iSR} is a complete graph. Let S be the neighbours of v_i' in G_{iSR} that do not belong to S_{v_i}. Then every vertex of S_{v_i} is adjacent with every vertex of S in G_{iSR}. So a minimum vertex cover of G_{iSR} either contains all vertices of S and exactly $|S_{v_i}| - 1$ vertices of S_{v_i} or if it does not contain all the vertices of S, it must contain all the vertices of S_{v_i}. Similarly, a minimum vertex cover for $G_{(i-1)SR}$ must contain all vertices of S and exactly $|S_{v_i'}| - 1$ vertices of $S_{v_i'}$, or if it does not contain all the vertices of S, it must contain all the vertices of $S_{v_i'}$. So $S_{iSR} \cup \{v_i\}$ is a minimum vertex cover of G_{iSR} and thus a minimum strong resolving set for G_i.

Subcase 3.2: Suppose v_i' is a true twin in G_{i-1}.

Let $S_{v_i'}$ be the collection of true twins of v_i' in G_{i-1}. Since v_i is a proper false twin each vertex of $S_{v_i'}$ has degree at least 2 in G_{i-1}. Then v_i is adjacent with every vertex of $S_{v_i'} \setminus \{v_i'\}$, but $v_i v_i' \notin E(G_i)$. So v_i' is no longer a true twin of any vertex of $S_{v_i'} \setminus \{v_i'\}$ in G_i. If $|S_{v_i'}| \geq 3$, then the vertices of $S_{v_i'} \setminus \{v_i'\}$ remain true twins in G_i. If $|S_{v_i'}| = 2$, then the vertex of $S_{v_i'} \setminus \{v_i'\}$ becomes a vertex of type other in G_{i-1} since it is not a twin and not a cut-vertex. So the vertex types of G_{iSR} are described correctly by the algorithm. Since $v_i' u \in E(G_i)$ for all

$u \in S_{v'_i} \setminus \{v'_i\}$ and $uv_i \in E(G_i)$ but $v_iv'_i \notin E(G)$, neither v'_i nor v_i is MMD with u in G_i. However, if $x \in V(G_{i-1}) \setminus S_{v'_i}$, then x is MMD with v'_i in G_{i-1}, if and only if x is MMD with v'_i in G_i. Since v_i is a false twin of v'_i in G_i, it follows that if $x \in V(G_{i-1}) \setminus S_{v'_i}$, then x MMD v_i in G_i if and only if x MMD v'_i in G_i if and only if x MMD v'_i in G_{i-1}. Hence G_{iSR} is correctly constructed in this case.

It remains to show that S_{iSR} is a minimum vertex cover for G_{iSR}. Let S be the set of neighbours of v_i in $G_{iSR} - v'_i$. Then S is also the set of neighbours of v'_i in $G_{iSR} - v'_i$ and the set of neighbours of v'_i in $G_{(i-1)SR} - (S_{v'_i} \setminus \{v'_i\})$. So a minimum vertex cover of G_{iSR} either contains all vertices of S and exactly one of v_i and v'_i and all but one of the vertices of $S_{v'_i} \setminus \{v'_i\}$, or it contains at most $|S| - 1$ vertices of S and all the vertices in $S_{v'_i} \cup \{v_i\}$.

Suppose first that $S \subseteq S_{(i-1)SR}$. Since $S_{(i-1)SR}$ is a minimum vertex cover of $G_{(i-1)SR}$, $S_{(i-1)SR}$ contains $|S_{v'_i}| - 1$ vertices of $S_{v'_i}$. If $v'_i \in S_{(i-1)SR}$, then by the above observation, $S_{(i-1)SR}$ is a minimum vertex cover of G_{iSR}. If $v'_i \notin S_{(i-1)SR}$, then $S_{v'_i} \setminus \{v'_i\} \subseteq S_{(i-1)SR}$. In this case if $u \in S_{v'_i} \setminus \{v'_i\}$, then $S_{iSR} = (S_{(i-1)SR} \setminus \{u\}) \cup \{v_i\}$ for some $u \in S_{v'_i} \setminus \{v'_i\}$ is a vertex cover for G_{iSR} and since $S_{(i-1)SR}$ is a minimum vertex cover for $G_{(i-1)SR}$, S_{iSR} is a minimum vertex cover of G_{iSR}.

Suppose next that $S \not\subseteq S_{(i-1)SR}$. Then by the above observation $S_{v'_i} \in S_{(i-1)SR}$. So if $|S \cap S_{(i-1)SR}| \leq |S| - 2$, then $S_{iSR} = S_{(i-1)SR} \cup \{v_i\}$ is a minimum vertex cover for G_{iSR}. Moreover, if $|S \cap S_{(i-1)SR}| = |S| - 1$, then $S_{(i-1)SR} \cup (S \setminus \{u\})$ for some $u \in S_{v'_i} \setminus \{v'_i\}$ must be a minimum vertex cover of G.

Subcase 3.3: v'_i is a cut-vertex of G_{i-1}.

It is straight forward to observe that in this case G_{iSR} can be obtained from $G_{(i-1)SR}$ by adding the edge $v_iv'_i$ since v_i MMD v'_i in G_i. Since v'_i is an isolated vertex of $G_{(i-1)SR}$, a minimum vertex cover for G_{iSR} can be obtained from a minimum vertex cover for $G_{(i-1)SR}$ by adding either v_i or v'_i but not both. So $S_{iSR} = S_{(i-1)SR} \cup \{v_i\}$ is a minimum vertex cover for G_{iSR}.

Subcase 3.4: v'_i is of type other in G_{i-1}.

It is a straight forward observation that v_i MMD v'_i in G_i. Thus $v_iv'_i \in E(G_{iSR})$. Moreover, if v'_i is MMD with a vertex u in G_{i-1}, then v_i is MMD with u in G_i and conversely. It is now easily seen that G_{iSR} is constructed correctly in this case. It remains to show that S_{iSR} is a minimum vertex cover of G_{iSR}. Let S be the set of neighbours of v'_i in $G_{(i-1)SR}$. A minimum vertex cover for G_{iSR} must contain either all the vertices of S and exactly one of v_i and v'_i, or at most $|S| - 1$ vertices of S and both v_i and v'_i. It is now readily verifiable that $S_{iSR} = S_{(i-1)SR} \cup \{v_i\}$ is a minimum vertex cover for G_{iSR}. □

3 Concluding Remarks

We developed a $O(|V||E|)$ algorithm for finding the strong metric dimension of a distance-hereditary graph. In [8] it is shown that distance-hereditary graphs have clique width at most 3 and that a 3-expression defining it can be obtained in linear time. Moreover, in [5] it is shown that every graph problem expressible in $LinEMSOL(\tau_{1,L})$ (a variation of Monadic Second Order Logic) is solvable on graphs with bounded clique width k if the input graph is given with a k-expression defining it.

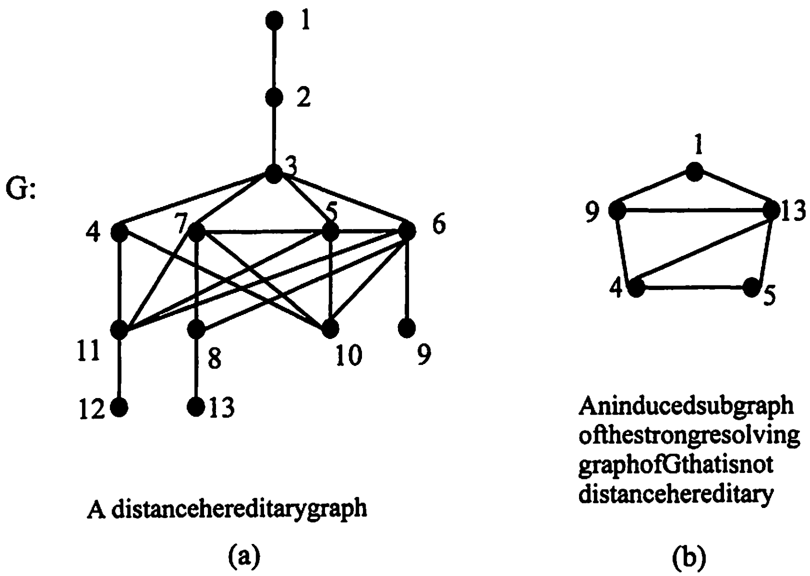

Figure 4: A distance-hereditary graph whose strong resolving graph is not distance-hereditary

Several graph problems such as vertex cover, maximum weight stable set, maximum weight clique, Steiner tree and domination are $LinEMSOL(\tau_{1,L})$ expressible. Since distance-hereditary graphs have bounded clique width, it is natural to ask if the strong dimension problem is $LinMSOL(\tau_{1,L})$ expressible. Since the strong dimension problem for a graph G reduces to a minimum vertex cover problem for the strong resolving graph G_{SR} this question has an affirmative answer if the strong resolving graphs of distance-hereditary graphs have bounded clique width and if a k-expression for these graphs can be found in polynomial time. In particular it is natural to ask if the strong resolving graph of a distance-hereditary graph is also distance-hereditary. The graph of Fig 4(a) illustrates that this is not the case. Since $3, 5, 6, 8, 7, 4, 10, 11, 2, 1, 9, 12, 13$ is a DH sequence for G it is indeed distance-hereditary. Moreover, the graph shown in Fig 4(b) (a 3-fan) is an induced subgraph of the strong resolving graph G_{SR} of G. Since the 3-fan is not distance-hereditary, G_{SR} is not distance-hereditary.

References

[1] H.J. Bandelt and H.M. Mulder, Distance-hereditary graphs. *J. Combin. Theory B* **41** (1986) 183-208.

[2] J. Cáceres, C. Hernando, M. Mora, I.M. Pelayo, M.L. Puertas, C. Seara and D.R. Wood, On the metric dimension of cartesian products of graphs. *SIAM J. Discr. Math* **21** (2007) 423-441.

[3] G. Chartrand, L. Eroh, M. Johnson, and O.R. Oellermann, Resolvability in graphs and the metric dimension of a graph. *Discr. Appl. Math.* **105** (2000) 99-113.

[4] G. Chartrand and L. Lesniak, *Graphs & Digraphs.* Fourth Edition. Chapman and Hall, London (2004).

[5] B. Courcelle, J. Engelfriedt and G. Rozenberg, Handle-rewriting hypergraph grammars. *J. Comput. System Sci.* **46** (1993) 218-270.

[6] J. Currie and O.R. Oellermann, The metric dimension and metric independence of a graph. *J. Combin. Math and Combin. Comput.* **39** (2001) 157-167.

[7] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, New York, 1979.

[8] M.C. Golumbic and U. Rotics, On the clique width of some perfect graph classes. *Intern. J. of Foundations of Computer Science* **11** (2000) 423-443.

[9] P.L. Hammer and F. Maffray, Completely separable graphs. *Discr. Appl. Math.* **27** (1990) 85-100.

[10] F. Harary and R.A. Melter, On the metric dimension of a graph. *Ars Combin.* **2** (1976) 191-195.

[11] E. Howorka, A characterization of distance-hereditary graphs, *Quart. J. Math. Oxford* **28** (1997) 417-420.

[12] S. Khuller, B. Raghavachari, and A. Rosenfeld, Localization in graphs. *Technical Report* CS-TR-3326, University of Maryland at College Park (1994).

[13] O.R. Oellermann and J. Peters-Fransen, The strong metric dimension of graphs and digraphs. *Discr. Appl. Math.* **155** (2007) 356-364.

[14] P.J. Slater, Leaves of trees. *Congr. Numer.* **14** (1975) 549-559.

[15] P.J. Slater, Dominating and reference sets in a graph. *J. Math. Phys. Sci.* **22** (1988) 445-455.

[16] A. Sebö and E. Tannier, On metric generators of graphs. *Mathematics of Operations Research* **29(2)** (2004) 383-393.