The (weighted) metric dimension of graphs : hard and easy cases

DOI:
10.1007/s00453-014-9896-2

Document status and date:
Published: 01/01/2015

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Algorithmica (2015) 72:1130–1171
DOI 10.1007/s00453-014-9896-2

The (Weighted) Metric Dimension of Graphs: Hard and Easy Cases

Leah Epstein · Asaf Levin · Gerhard J. Woeginger

Received: 6 February 2013 / Accepted: 20 May 2014 / Published online: 30 May 2014

Abstract Given an input undirected graph $G = (V, E)$, we say that a vertex ℓ separates u from v (where $u, v \in V$) if the distance between u and ℓ differs from the distance from v to ℓ. A set of vertices $L \subseteq V$ is a feasible solution if for every pair of vertices, $u, v \in V$ ($u \neq v$), there is a vertex $\ell \in L$ that separates u from v. Such a feasible solution is called a *landmark set*, and the metric dimension of a graph is the minimum cardinality of a landmark set. Here, we extend this well-studied problem to the case where each vertex v has a non-negative cost, and the goal is to find a feasible solution with a minimum total cost. This weighted version is NP-hard since the unweighted variant is known to be NP-hard. We show polynomial time algorithms for the cases where G is a path, a tree, a cycle, a cograph, a k-edge-augmented tree (that is, a tree with additional k edges) for a constant value of k, and a (not necessarily complete) wheel. The results for paths, trees, cycles, and complete wheels extend known polynomial time algorithms for the unweighted version, whereas the other results are the first known polynomial time algorithms for these classes of graphs even for the unweighted version. Next, we extend the set of graph classes for which computing the

An extended abstract of this paper appears in Proc. WG2012, pages 114–125.

L. Epstein (✉)
Department of Mathematics, University of Haifa, 31905 Haifa, Israel
e-mail: lea@math.haifa.ac.il

A. Levin
Faculty of Industrial Engineering and Management, The Technion, 32000 Haifa, Israel
e-mail: levinas@ie.technion.ac.il

G. J. Woeginger
Department of Mathematics and Computer Science, TU Eindhoven, P.O. Box 513,
5600 MB Eindhoven, The Netherlands
e-mail: gwoegi@win.tue.nl

unweighted metric dimension of a graph is known to be NP-hard. We show that for split graphs, bipartite graphs, co-bipartite graphs, and line graphs of bipartite graphs, the problem of computing the unweighted metric dimension of the graph is NP-hard.

Keywords Graph algorithms · Metric dimension · Graph classes

1 Introduction

Let $G = (V, E)$ be a simple, loopless, undirected graph. Denote the number of vertices in G by $n = |V|$ and the number of its edges by $m = |E|$. A vertex $\ell \in V$ is called a *separating landmark* for two vertices $u, v \in V$ with $u \neq v$, if the length of a shortest path from u to ℓ differs from the length of a shortest path from v to ℓ; sometimes we will then also say that vertex ℓ *separates* or *distinguishes* u from v. A subset $L \subseteq V$ is a *landmark set* (or a feasible landmark set) for the graph G, if for any two vertices $u, v \in V$ with $u \neq v$ there exists a separating landmark $\ell \in L$ that distinguishes u from v. We also refer to such a set as a *solution*. The *metric dimension* $\mathtt{md}(G)$ of the graph G is the smallest cardinality of any landmark set in G. Note that $\mathtt{md}(G)$ is well-defined, as $L = V$ trivially forms a landmark set for G. Additionally, $\mathtt{md}(G) = 0$ holds if and only if $|V| = 1$. In this paper, we consider the problem of computing the metric dimension of an input graph G. Applications of this optimization problem arise in diverse areas. See [2] for an application of this problem in network verification, [6] for an application in strategies for the Mastermind game, [9] for an application in metric geometry, [15] for an application in digital geometry, namely in digitizing of images, [14] for an application in robot navigation, and [4] for an application in drug discovery. Many of these applications are relevant for weighted graphs.

This metric dimension problem was introduced by Harary and Melter [9] and by Slater [18], and studied widely in the combinatorics literature. In this line of research, the exact values of the metric dimension or bounds on it for specific graph classes are obtained. We refer to [1,3,4] for results additional to those mentioned here (see also the survey [5]). Khuller et al. [14] showed that $\mathtt{md}(G) = 1$ if and only if G is a path. Tree input graphs that are not paths were considered in [4,9,14,18]. It turns out that it is possible to characterize the feasibility of a landmark set for a tree using a notion of *legs*, which are paths of vertices of degree at most 2 connected to a vertex of a higher degree. For the case that the input graph is a cycle, it holds that $\mathtt{md}(G) = 2$ [9]. Complete wheel graphs were mentioned in [9], and further studied in [17]. Melter and Tomescu [15] considered the problem for grid-graphs induced by lattice points in the plane when the distances are measured in the L_1 norm or in the L_∞ norm. Khuller et al. [14] generalized one result of [15] to lattice points contained inside a d-dimensional rectangle, where the distance is according to the L_1 norm (that is, the grid-graph over points in a d-dimensional rectangle). The grid-graph with Euclidean metrics was studied in the paper [16], where the authors relate the problem to a combinatorial coin weighing problem.

As for the complexity of the problem, Khuller et al. [14] proved that the problem is NP-hard for general graphs, and showed that one can obtain a landmark set for a given instance by applying the greedy algorithm to a certain set cover instance (where we would like to cover the pairs of vertices in a graph using sets which are defined using a single landmark). Thus, there is an $(2 \ln n + O(1))$-approximation algorithm for general graphs. Beerliova et al. [2] showed that if $P \neq NP$ then there is no $o(\log n)$-approximation algorithm for the problem. In [11], Hauptmann, Schmied, and Viehmann strengthened these results and showed that if $NP \not\subseteq DTIME(n^{\log \log n})$, then there is no $((1-\varepsilon) \ln n)$-approximation algorithm for any $\varepsilon > 0$. They give an improved $(1 + (1 + o(1)) \ln n)$-approximation algorithm. Diaz et al. [8] showed that the problem is NP-hard even when restricted to planar input graphs (but solvable in polynomial time for outerplanar graphs), and inspired by the last result, Hoffmann and Wanke announced a reduction showing that it is NP-hard even for Gabriel unit disk graphs [12]. Very recently, Hartung and Nichterlein [10] proved even stronger results as follows. They showed that the problem is W[2]-hard with respect to the parameter $\mathtt{md}(G)$. It implies that the problem of computing the metric dimension of a graph cannot be solved in $n^{o(\mathtt{md}(G))}$ time, unless FPT = W[1], and under the assumption FPT $\neq$ W[1], the trivial $n^{O(\mathtt{md}(G))}$-time algorithm that tests all possible subsets of the vertices sorted by non-decreasing cardinality is asymptotically optimal. They also show that $\mathtt{md}(G)$ on maximum degree three graphs is inapproximable by a factor of $o(\log n)$, unless P = NP.

Here we generalize the problem of computing $\mathtt{md}(G)$ to a weighted variant. More precisely, given a non-negative cost function $c : V \to \mathbb{R}_+$, the goal is to compute a landmark set L such that $\sum_{\ell \in L} c(\ell)$ is minimized. This function is given as input together with the graph G. We let $\mathtt{wmd}(G)$ denote this minimum cost and we say that $\mathtt{wmd}(G)$ is the weighted metric dimension of G. The $\mathtt{wmd}$ problem is to compute a landmark set L of minimum total cost, and to find $\mathtt{wmd}(G)$. Our polynomial time algorithms for special classes of graphs will be for solving $\mathtt{wmd}$ while our NP-hardness proofs will hold even for the unweighted version of computing $\mathtt{md}(G)$ and thus the same holds for the weighted variant as well. We are not aware of any previous work on the weighted version. We say that a landmark set is minimal if it is minimal with respect to inclusion. Note that there is always an optimal solution (a minimum cost landmark set) that is also minimal (since the cost function is non-negative), and thus we sometimes characterize the set of minimal solutions.

We note that the problem is clearly in NP because given a subset of vertices,we can verify its feasibility as a landmark set in polynomial time. To do this, we find the vectors of the distances of each vertex in V from each of the landmarks. Afterwards, we check that there is no pair of identical vectors. In some of our algorithms we perform an exhaustive enumeration of landmark sets (among a restricted family of vertex subsets), and we can always find a cheapest landmark set among such a restricted family of subsets (we first ensure that it contains at least one optimal solution).

1.1 Our Results

We generalize the known polynomial time algorithms for $\mathtt{md}$ (the problem of finding a landmark set of minimum cardinality) to $\mathtt{wmd}$ for the cases where G is a path, a tree, a cycle, or a complete wheel. We develop polynomial time algorithms for the weighted problem when G is a cograph, a k-edge-augmented tree (that is, a tree with additional

k edges) for a constant value of k, and a (not necessarily complete) wheel. These results are the first polynomial time algorithms even for the unweighted version when G belongs to these classes of graphs. One of our approaches for designing polynomial time algorithms is as follows. For a given graph class we identify the structure of minimal (with respect to inclusion) landmark sets. This leads, in several considered graph families, to an exploration of a polynomial number of candidate solutions or to a dynamic programming approach. Thus the algorithm seeks for a minimum cost set among the minimal landmarks sets. Other methods include defining (and solving) a related auxiliary problem, and preprocessing. Next, we extend the set of graph classes for which md is known to be NP-hard. Specifically, we show that md(G) is NP-hard when the input graph G is a split graph, bipartite graph, co-bipartite graph, or a line graph of a bipartite graph.

1.2 Definitions and Notation

Given a graph $G = (V, E)$, we say that a vertex $v \in V$ is a *leaf* if its degree is 1, it is an *isolated vertex* if its degree is 0, if its degree is 2 it is a *path vertex*, and higher degree vertices are called *core vertices*. For a pair of vertices u, v we denote by $d_{u,v}$ the length of a shortest path (i.e., the number of edges in the path) in G from u to v. Recall that $n = |V|$ and $m = |E|$, we use these definitions throughout the paper.

2 Extending Known Polynomial Cases to the Weighted Variant

In this section we generalize some polynomially solvable cases of md(G) to the weighted case. These simple cases emphasize some differences between the weighted and the unweighted cases.

2.1 Paths

First, consider the case where G is a path. Khuller et al. [14] showed that a landmark set consisting of one vertex, positioned at one of the end-vertices of the path, is a landmark set. Our algorithm for computing wmd(G) (and a corresponding landmark set) for a path G is defined as follows. The algorithm finds two alternative solutions (landmark sets) and outputs a solution of minimum cost. The first candidate solution of the algorithm has a single vertex as a landmark: a minimum cost end-vertex v of the path (breaking ties arbitrarily). The second solution picks the cheapest pair of distinct vertices v and v' of the path (none of which is an end-vertex of the path).

Proposition 2.1 *Given a path $G = (V, E)$, the above algorithm solves* wmd *in linear time $O(n)$.*

Proof If the returned set has one vertex, then feasibility of L was established by [14]. Otherwise, the feasibility follows from the property that out of two vertices ℓ_1, ℓ_2 on the path, a pair u, v of vertices on the path can have equal distances to at most one of ℓ_1 and ℓ_2. □

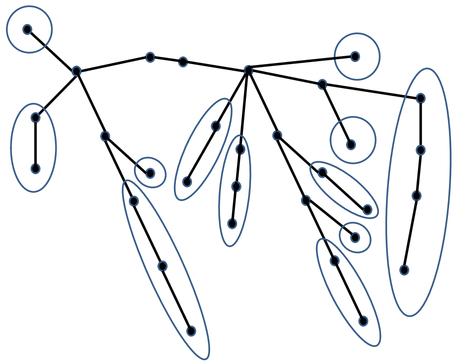

Fig. 1 An example of a tree and all legs, where the vertices of each leg are *circled*

2.2 Trees

Next, we assume that the input graph G is a tree, but not a path, (and so it has at least one core vertex). A leg is a (non-empty) path in the tree between a leaf v and a core vertex u of the graph that is closest to this leaf (the leg does not contain the core vertex u but it contains the leaf v). We say that this leg is *a leg of* u. Note that a leg contains one leaf, and possibly some path vertices (see Fig. 1). For a vertex u, denote the number of legs of u by leg_u.

Consider the following algorithm. Compute leg_u for every core vertex u. Each core vertex u with $leg_u \geq 2$ is allocated $leg_u - 1$ landmarks. To place these landmarks, we find a minimum cost set of $leg_u - 1$ vertices in the legs of u (none of which will be u, as u does not belong to its legs), such that each leg has at most one selected vertex. It was shown [14] that in every landmark set there is a landmark located at each of the legs of u except for at most one such leg. In fact, summing up the values $leg_u - 1$ over all core vertices u with at least two legs, gives a tight bound on $\mathtt{md}(G)$. Here, we generalize their approach.

Note that the running time of the algorithm is $O(n)$ since we can compute the set of legs of each core vertex in $O(n)$ time by running DFS from a core vertex, and for each core vertex u, choose the cheapest set of $leg_u - 1$landmarks such that each leg has at most one selected vertex. Moreover, by the lower bound proof of [14], any feasible solution (a valid landmark set) must place landmarks in at least $leg_u - 1$ legs of u, it is clear that the cost of the landmark set which the algorithm returns is minimal. The algorithm of [14] places the landmarks at the leaves, whereas our algorithm may place them at internal vertices (path vertices) of legs. It remains to argue that the resulting set is feasible. Denote the landmark set returned by our algorithm by L. The proof of the following lemma resembles the proof of Lemma 2.3 of [14].

Lemma 2.2 *Given $L \subseteq V$ as defined above, if $v \neq v'$ is a pair of vertices of V, then there is a landmark $\ell \in L$ which distinguishes v from v'.*

Proof Root the tree at an arbitrary landmark vertex r. We will use the following two properties. Consider a core vertex w, then in the rooted subtree of w there is a core vertex (perhaps w) with at least two legs, and hence this subtree contains a landmark. The second property is that if a vertex w is not on a leg, then either it is a core vertex or there is a core vertex in its rooted subtree. In both cases, the rooted subtree contains a landmark.

If v and v' are at different distances from r, then r distinguishes v from v' and we are done. Thus, assume that v and v' are at the same depth. If at least one of them has a landmark inside its rooted subtree, then this landmark separates v from v' (since, for example, the path from v' to any vertex in the rooted subtree of v traverses v), and we are done. Thus, in the remainder of the proof we assume that v and v' belong to legs. Denote by w the lowest common ancestor of v and v'. If w is the unique vertex of degree at least 3 along the path from v to v' (in which case v, v' belong to legs of w), then among the vertices of the two legs that contain v and v' there must be a landmark (and this landmark cannot be positioned at w, because w is not part of these legs). If this last landmark is on the path from v to v', then it distinguishes v from v', since the vertex in the middle of this path is w. If this vertex is not on this last path, then the path from one of the two vertices v, v' to the landmark traverses the other one, and therefore the landmark distinguishes v from v' in this case as well. Otherwise, if there is another core vertex $\tilde{w} \neq w$ on the path from v to v', then assume without loss of generality that $\tilde{w}$ is on the path from v to w. By the second property, in the subtree rooted at $\tilde{w}$ there is a landmark at a vertex ℓ. Since $d_{v,\tilde{w}} < d_{v',\tilde{w}}$ (as w has the same distance from v and v' while $\tilde{w}$ is on the path from w to v), we conclude that $d_{v,\ell} \leq d_{v,\tilde{w}} + d_{\tilde{w},\ell} < d_{v',\tilde{w}} + d_{\tilde{w},\ell} = d_{v',\ell}$ where the last equality holds because the shortest path from v' to ℓ must traverse $\tilde{w}$. Therefore, ℓ separates v from v'.

Thus, we have established the following result.

Proposition 2.3 *Given a tree $G = (V, E)$, there exists an algorithm that solves* `wmd` *in linear time $O(n)$.*

2.3 Cycles

We assume that the input graph G is a cycle (and thus $n \geq 3$). We next characterize a minimal (with respect to inclusion) feasible landmark set L. We say that a pair of distinct vertices u, v are opposite if their distance is exactly $\frac{n}{2}$, and otherwise they are non-opposite (in which case the shortest path from u to v is unique). Note that if G is an odd-length cycle, then there are no pairs of opposite vertices.

Lemma 2.4 *Let L be a minimal feasible landmark set of a cycle G. Then L consists of a pair of non-opposite vertices v, v' ($v \neq v'$) of G.*

Proof Every feasible landmark set consists of at least two vertices (if a cycle has a single landmark, then the two vertices of distance 1 from it cannot be distinguished). Moreover, a pair of opposite vertices v,v' is not a feasible landmark set, as the two neighbors of v cannot be distinguished.

We now show that any pair of non-opposite vertices $L = \{v, v'\}$ ($v \neq v'$) is a feasible landmark set. Obviously L separates any vertex of L from any vertex. Let $u \neq u'$ be a pair of vertices such that $u, u' \neq v, v'$. If u and u' have equal distances to both v and v', then the shortest paths of u and u' to any vertex of L do not have common edges. We find that $d_{v,v'} = \frac{n}{2}$, so v and v' are opposite, a contradiction.

Every landmark set with at least three distinct vertices contains a pair of non-opposite vertices, and thus it cannot be minimal. Thus, the only type of a minimal landmark set is a pair of non-opposite vertices. □

Our algorithm for solving wmd for a cycle G simply finds the cheapest pair of (distinct) non-opposite vertices in the cycle. By the above lemma, it finds an optimal solution. Note that the running time of the algorithm is $O(n)$ by first identifying the cheapest set of any three vertices (breaking ties arbitrarily) as a first step, and finding the cheapest pair of distinct non-opposite vertices among them as a second step (breaking ties arbitrarily).

Proposition 2.5 *Given a cycle $G = (V, E)$, there exists an algorithm that solves* wmd *in linear time $O(n)$.*

Proof First, note that the algorithm outputs two vertices that are a landmark set. This holds since $n \geq 3$, so any set of three vertices must contain at two or three (non-disjoint) subsets of two vertices that are not opposite, and by Lemma 2.4, every such set is a landmark set. Let $\{v_1, v_2\}$ such that $c(v_1) \leq c(v_2)$ be the output of the algorithm. Obviously, v_1 and v_2 are among the three vertices selected by the algorithm in the first step.

Consider a minimum cost landmark set. By Lemma 2.4, this set consists of a pair $\{u_1, u_2\}$ of non-opposite vertices such that $c(u_1) \leq c(u_2)$. If both $c(v_1) \leq c(u_1)$ and $c(v_2) \leq c(u_2)$ hold, then we are done. Assume by contradiction that this is not the case. If $c(u_2) < c(v_2)$, then u_2 must be one of the three vertices selected by the algorithm in the first step, as it selected a vertex (v_2) of larger cost in this step. Since $c(u_1) \leq c(u_2) < c(v_2)$, it also selected u_1 in the first step. As u_1 and u_2 are non-opposite, the algorithm must have $c(v_1) + c(v_2) \leq c(u_1) + c(u_2)$. The remaining case is $c(u_1) < c(v_1)$ and $c(u_2) \geq c(v_2)$ (and in particular $v_1 \neq u_1$). The algorithm selected both u_1 and v_1 in the first step. If they are non-opposite, then $c(v_1) + c(v_2) \leq c(u_1) + c(v_1) \leq c(u_1) + c(v_2) \leq c(u_1) + c(u_2)$. Otherwise, u_1, v_1 are opposite, $v_2 \neq u_1$, and the three vertices selected by the algorithm in the first step are v_1, v_2, u_1. However, in this case, u_1 and v_2 are non-opposite, and $c(u_1) + c(v_2) < c(v_1) + c(v_2)$, contradicting the action of the algorithm. □

3 Dealing with Disconnected Input Graphs

Consider a disconnected graph $G = (V, E)$. A connected component of G is called a *non-trivial connected component* if it contains at least two vertices, and otherwise it is an isolated vertex. Let $(V_1, E_1), \ldots, (V_p, E_p)$ be the non-trivial connected components of G (for $p \geq 0$), and let $v_1, \ldots, v_t$ be its isolated vertices (for $t \geq 0$), where if $p = 0$ or $t = 0$ then there is no non-trivial connected component or an isolated vertex, respectively. Without loss of generality, we assume that $c(v_t) = \max_{i:1 \leq i \leq t} c(v_i)$. In this section we show that it is sufficient to solve the weighted metric dimension problem for each non-trivial connected component of G. The time complexity of solving wmd is $O(n+m)$ plus the total running times of solving wmd for each of the non-trivial connected components of G.

Proposition 3.1 *An optimal solution (a minimum cost landmark set) L for* wmd *of a graph G is achieved by the union of an optimal solution L_i for* wmd *for each non-trivial connected component (V_i, E_i) of G and the (isolated) vertices $\{v_1, v_2, \ldots, v_{t-1}\}$ of G (the most expensive isolated vertex v_t is not a landmark).*

Proof We first argue that the totalcost of L is at least $\sum_{v \in L_1 \cup L_2 \cup \cdots \cup L_p} c(v) + \sum_{i=1}^{t-1} c(v_i)$. To see this claim, first note that if there exists a pair of isolated vertices $v, v' \notin L$ $(v \neq v')$, then L does not separate v from v' (as their distance from any vertex in L is ∞). Thus, L must contain at least $t-1$ isolated vertices of cost at least $\sum_{i=1}^{t-1} c(v_i)$. Next, consider a non-trivial connected component (V_i, E_i). It suffices to show that $L \cap V_i$ must be a feasible landmark set for the induced subgraph (V_i, E_i) (and in particular it cannot be empty). Assume for contradiction that there is a pair of vertices $v, v' \in V_i$ such that $L \cap V_i$ does not separate v from v'. Since the distances satisfy $d_{v,\ell} = d_{v',\ell} = \infty$ for all $\ell \in L \setminus V_i$, we conclude that L does not separate v from v' (in G), and this contradicts the assumption that L is a feasible landmark set.

It remains to show that the solution obtained by concatenating the solutions for every non-trivial connected component with$\{v_1, v_2, \ldots, v_{t-1}\}$ gives a feasible landmark set L' for G. To see this claim consider a pair of vertices $v, v' \notin L'$. Every connected component, except for at most one component that is an isolated vertex, has at least one landmark. This holds since any non-trivial connected graph must have at least one landmark, and all isolated vertices but one are defined to be landmarks. On one hand, if v and v' belong to distinct connected components of G, then in at least one of these connected components there is a landmark $\ell \in L'$, and ℓ separates v from v' (as it has a finite distance from one of them and infinite distance from the other). On the other hand, if v and v' belong to the same connected component, then it is a non-trivial connected component, and we know that there is a landmark $\ell \in L'$ which separates v from v'. □

We will use these properties in the next section, but it should be noted that Proposition 3.1 can be applied to any disconnected graph, and in particular implies that there exists a polynomial time algorithm for solving wmd on forests, collections of vertex disjoint cycles, etc.

4 Cographs

For two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ with $V_1 \cap V_2 = \emptyset$, the disjoint union $G_1 \cup G_2$ is the graph $(V_1 \cup V_2, E_1 \cup E_2)$. The product $G_1 \times G_2$ of these two graphs is obtained by first taking the disjoint union of G_1 and G_2 and then adding all the edges $\{v_1, v_2\}$ with $v_1 \in V_1$ and $v_2 \in V_2$. A graph G is a *cograph*, if (i) G consists of a single vertex or (ii) G is the disjoint union of two cographs, or (iii) G is the product of two cographs. An equivalent characterization states that G is a cograph if and only if it does not contain the path P_4 on four vertices as an induced subgraph. This implies, in particular, that the distances between pairs of distinct vertices in the graph are in $\{1, 2, \infty\}$. Note that the complement graph $G' = (V, \{\{u, v\} | u, v \in V, \{u, v\} \notin E\})$ of a cograph $G = (V, E)$ is a cograph as well.

The *cotree* of a cograph G is a rooted binary tree whose leaves correspond to single-vertex graphs and whose inner nodes correspond to subgraphs of G. Every inner node of the cotree is labelled either by $\cup$ (union) or by $\times$ (product) and has exactly two children: if it is labelled by $\cup$ then it corresponds to the disjoint union of the two cographs that correspond to its two children, and if it is labelled by $\times$ then it corresponds to the product of the two cographs that correspond to its children. Corneil et al. [7] showed how to compute a cotree for a given cograph in linear time $O(m+n)$. Note that the number of inner nodes of the cotree is $n-1$.

By Proposition 3.1, we conclude that we may restrict ourselves to connected cographs. Since a connected cograph with at least two vertices must be a product of two non-empty cographs, the distance between a pair of vertices v, v' ($v \neq v'$) of a connected cograph G is either 1 or 2. We define a *binary landmark set* L of an arbitrary cograph G (not necessarily a connected one) to be a set of vertices such that for every pair of distinct vertices $v, v' \in V \setminus L$ there is a landmark $\ell \in L$ such that either both $\{v, \ell\} \in E$ and $\{v', \ell\} \notin E$ or both $\{v, \ell\} \notin E$ and $\{v', \ell\} \in E$. In this case we say that ℓ separates v from v'.

Claim 4.1 *Given a connected cograph, a set of vertices is a landmark set if and only if it is a binary landmark set.*

Proof We first prove that for any graph $G' = (V', E')$, a given binary landmark set L of G' is also a feasible landmark set of G'. Let $v, v' \in V'$. If $v \in L$ (or $v' \in L$), then this vertex separates v from v'. Otherwise, since L is a binary landmark set, there is $\ell \in L$ such that either both $d_{v,\ell} = 1$ and $d_{v',\ell} \geq 2$ or both $d_{v,\ell} \geq 2$ and $d_{v',\ell} = 1$. In both cases ℓ separates v from v'.

Assume that L is a landmark set of a connected cograph $G = (V, E)$. Let $v, v' \in V \setminus L$, then there is a landmark $\ell \in L$ such that $d_{v,\ell} \neq d_{v',\ell}$. Note that these two distinct distances are not zero (as $v, v' \notin L$) so they are either 1 or 2. Thus, one of these vertices is adjacent to ℓ and the other is not adjacent. Therefore, L is a binary landmark set. □

In the remainder of this section we will present a linear time algorithm for computing a binary landmark set of a minimum total cost. The next observation (which holds by definition) considers the relation between binary landmark sets of G and its complement G'.

Observation 4.2 *Let $G = (V, E)$ be a cograph on $n \geq 2$ vertices, and let G' be its complement graph. Then a set $L \subseteq V$ is a binary landmark set for G if and only if L is a binary landmark set for G'.*

Proof It suffices to show that if L is a feasible binary landmark set for G, then it is also a feasible landmark set for G'. Denote by E' the edge set of G'. Assume by contradiction that there exists a pair of vertices $u, v \in V \setminus L$ such that for every $\ell \in L$ either both $\{u, \ell\}, \{v, \ell\} \in E'$ or both $\{u, \ell\}, \{v, \ell\} \notin E'$. In the first case both $\{u, \ell\}, \{v, \ell\} \notin E$, and in the second case both $\{u, \ell\}, \{v, \ell\} \in E$. This contradicts the assumption that L is a feasible binary landmark set of G. □

We next adapt our decomposition of the problem for disconnected graphs to the problem of computing a binary landmark set. We consider the case $G = G_1 \cup G_2$, and show the following.

Lemma 4.3 *Assume that G is a disjoint union of G_1 and G_2 where $G_i = (V_i, E_i)$.*

(i) If L is a feasible binary landmark set for G, then $L_i = L \cap V_i$ is a feasible binary landmark set for G_i, for $i = 1, 2$.
(ii) Assume that L_1 and L_2 are feasible binary landmark sets for G_1 and G_2, respectively. Then, $L = L_1 \cup L_2$ is a feasible binary landmark set for G if and only if there exists $i \in \{1, 2\}$ such that in G_i every vertex $v \in V_i \setminus L_i$ is adjacent to a vertex of L_i.

Proof Consider claim (i). For $i = 1, 2$, consider a pair of vertices $u, v \in V_i \setminus L_i = V_i \setminus L$. Note that for every $\ell \in L_{3-i}$ there is no edge connecting ℓ to u or to v. Hence, since L is a feasible binary landmark set, we conclude that there is a landmark $\ell \in L_i$ that separates u from v, that is, it is adjacent to exactly one of these vertices. Thus, L_i is a feasible binary landmark set for G_i.

Next, consider claim (ii). Assume that $L_1 \cup L_2$ is a feasible binary landmark set for G. Assume by contradiction that for $i = 1, 2$, G_i contains a vertex u_i that is not adjacent to L_i. Clearly, u_i is not adjacent to L_{3-i} as well and hence L does not separate u_1 from u_2, contradicting the assumption that L is a feasible binary landmark set for G.

On the other hand, assume without loss of generality that for every vertex $w \in V_1 \setminus L_1$ there is a landmark in L_1 adjacent to w. Consider a pair of vertices $u, v \in V \setminus L$. If $u, v \in V_i$ (for $i = 1, 2$), then since L_i is a feasible binary landmark set for G_i, we conclude that there is $\ell \in L_i$ (and thus $\ell \in L$) such that ℓ is adjacent to exactly one of the vertices u or v, and we are done. Next, assume without loss of generality that $u \in V_1$ and $v \in V_2$. By the assumption, u is adjacent to a landmark $\ell \in L$. Then since $\ell \in V_1$ we conclude that $\{v, \ell\} \notin E$, and we are done in this case as well. □

In a nutshell, our algorithm for solving `wmd` of a cograph uses the cotree structure. We use the term *point* to refer to a vertex that is not a landmark. If a subgraph is a disjoint union of two subgraphs, then we treat recursively each of the two subgraphs but we keep track of the existence of a point that is not adjacent to any landmark in its subgraph. If a subgraph is the product of two subgraphs, then we first transform our problem to the complement of the subgraph, and then apply the first case. Since by moving from a graph to its complement, the role of a point that is not adjacent to any landmark switches with the role of a point that is adjacent to all landmarks, we will keep track of the number of points (zero or one) of each of these types. Note that there cannot be two points that are adjacent to all landmarks, as these two points cannot be distinguished by the binary landmark set, and similarly, there cannot be two points that are not adjacent to any landmark. Thus, every subgraph in the cotree will correspond to four optimization problems.

Theorem 4.4 *Given a cograph $G = (V, E)$,* `wmd` *can be solved in linear time $O(m + n)$.*

Proof By Proposition 3.1, it suffices to consider a connected cograph G. By Claim 4.1, it is sufficient to compute a minimum cost binary landmark set for G, and give it as an output, since for connected graph the two concepts (of a landmark set and of a binary landmark set) are equivalent. Our algorithm computes the cotree of G, and computes a minimum cost binary landmark set for every internal node of the cotree, using dynamic programming on a tree. We now define the dynamic programming formulation. Consider a cograph $\tilde{G}$ containing at least two vertices that corresponds to an internal node of the cotree. Let $a, b \in \{0, 1\}$, and let $F_{a,b}(\tilde{G})$ be a minimum cost feasible binary landmark set L of a cograph $\tilde{G}$ such that exactly a of its points are adjacent to all vertices of L, and exactly b of its points are not adjacent to any vertex of L. Our algorithm computes $F_{a,b}(\tilde{G})$ for every $\tilde{G}$ consisting of at least two vertices in the cotree (i.e., for every internal node of the cotree) and for every $a, b \in \{0, 1\}$. The number of such problems is $O(n)$, and thus it suffices to show how to compute each of these values in $O(1)$ using the solutions for the two subgraphs $\tilde{G}_1 = (\tilde{V}_1, \tilde{E}_1)$ and $\tilde{G}_2 = (\tilde{V}_2, \tilde{E}_2)$ corresponding to the children in the cotree of $\tilde{G}$ (if such solutions are defined, that is, the corresponding graphs contain at least two vertices). Recall that $F_{a,b}$ cannot be defined for $a > 1$ or for $b > 1$ since this would imply that the binary landmark set is infeasible. Note that while G is connected, some of the graphs corresponding to nodes of the cotree of G may be disconnected.

Let $\tilde{G}'$ be the complement of $\tilde{G}$. Similar to Lemma 4.2 we can find the relation between the functions F for $\tilde{G}$ and $\tilde{G}'$. Here, we have $F_{a,b}(\tilde{G}') = F_{b,a}(\tilde{G})$, as if we move from a graph to its complement then a point which was adjacent to all landmarks becomes non-adjacent to all landmarks and vice versa. Therefore, it suffices to consider an inner node of the cotree that is a union of two disjoint subgraphs (that is, $\tilde{G} = \tilde{G}_1 \cup \tilde{G}_2$). Throughout the computation we encode infeasible problems by allocating them infinite cost. It remains to define the values $F_{a,b}$ for the disconnected graph $\tilde{G}$ whose children in the cotree are $\tilde{G}_1$ and $\tilde{G}_2$, either directly or using recurrences, depending on the case.

We now define the base cases for the dynamic programming. First, assume that each of $\tilde{G}_1$ and $\tilde{G}_2$ consists of a single isolated vertex denoted as v_1 and v_2 respectively. $F_{a,b}(\tilde{G})$ is defined as follows. If $a = 1$, then the instance is infeasible, because for any instance with $|V| \geq 2$ we must place at least one landmark, and in this case the graph does not contain a point adjacent to all vertices in L. Therefore, $F_{1,0}(\tilde{G}) = F_{1,1}(\tilde{G}) = \infty$. If $a = b = 0$, placing just one landmark we would get that the other vertex is not adjacent to any of the landmarks, so we must place a landmark at each vertex, and hence $F_{0,0}(\tilde{G}) = c(v_1) + c(v_2)$. Finally, if $b = 1$, then we must place exactly one landmark, while the other vertex will be the one that is not adjacent to any landmark, and any such solution is a feasible binary landmark set. Hence, $F_{0,1}(\tilde{G}) = \min\{c(v_1), c(v_2)\}$.

In the remaining cases we define the recurrence relations. Next, assume that $\tilde{G}_1$ has more than one vertex, and hence any feasible binary landmark set for $\tilde{G}_1$ has at least one vertex, and assume that $\tilde{G}_2$ consists of a single vertex v. If $a = 1$, then it is impossible to place a landmark at v (because there is no point of $\tilde{V}_1$ that is adjacent to v) and v will not be adjacent to any of the landmarks, so in the remaining instance (i.e., in $\tilde{G}_1$), there cannot be an additional point of distance at least 2 from every landmark (and we must have a point of $\tilde{V}_1$ that is adjacent to all landmarks). Therefore, $F_{1,1}(\tilde{G}) = F_{1,0}(\tilde{G}_1)$, and $F_{1,0}(\tilde{G}) = \infty$. In the remaining cases $a = 0$. If we do not place a landmark at v, then v is not adjacent to any landmark, and $b = 1$. If there is a landmark at v, then there cannot be a point adjacent to all landmarks, even if there is such a point for $\tilde{G}_1$. Thus,we have the following $F_{0,0}(\tilde{G}) = c(v) + \min\{F_{0,0}(\tilde{G}_1), F_{1,0}(\tilde{G}_1)\}$, and $F_{0,1}(\tilde{G}) = \min\{F_{0,1}(\tilde{G}_1) + c(v), F_{1,1}(\tilde{G}_1) + c(v), F_{0,0}(\tilde{G}_1)\}$.

In the remaining case each of $\tilde{G}_1$ and $\tilde{G}_2$has more than one vertex, and has at least one landmark in any feasible binary landmark set. Hence, no point in $\tilde{G}$ can be adjacent to all landmarks. Thus, if $a = 1$, the problem is infeasible and we have $F_{1,0}(\tilde{G}) = F_{1,1}(\tilde{G}) = \infty$. If $a = 0$, then there can be at most one point in either G_1 or G_2 that is not adjacent to a landmark. Therefore, we have $F_{0,0}(\tilde{G}) = \min\{F_{0,0}(\tilde{G}_1), F_{1,0}(\tilde{G}_1)\} + \min\{F_{0,0}(\tilde{G}_2), F_{1,0}(\tilde{G}_2)\}$, and finally

$$F_{0,1}(\tilde{G}) = \min\Big\{\min\{F_{0,1}(\tilde{G}_1), F_{1,1}(\tilde{G}_1)\} + \min\{F_{0,0}(\tilde{G}_2), F_{1,0}(\tilde{G}_2)\}, \\ \min\{F_{0,0}(\tilde{G}_1), F_{1,0}(\tilde{G}_1)\} + \min\{F_{0,1}(\tilde{G}_2), F_{1,1}(\tilde{G}_2)\}\Big\}.$$

□

5 *k*-Edge-Augmented Trees

In this section we consider the class of connected graphs for which a removal of at most k edges results in a spanning tree. We call this class of graphs *k-edge-augmented trees*. We note that this class of graphs is related to the almost-k-trees discussed in [13], however some of the literature on almost-k-trees used an alternative definition in which every edge biconnected component has at most k edges whose removal results in a tree over the vertices of this biconnected component. Thus, we will use the terminology of k-edge-augmented trees to refer to this class of graphs that we study. Our polynomial time algorithm for computing `wmd` first applies a preprocessing step that handles the tree-like part, and then uses an exhaustive enumeration approach for selecting an optimal landmark set in a reduced problem. Thus, the methods used in this section are recursion and preprocessing, and for the graphs resulting from preprocessing, once again we find properties of minimal landmark sets that allow us to enumerate a sufficiently small number of subsets as candidate landmark sets. Clearly our algorithm is polynomial only if k is a constant and it is unlikely that there is an algorithm which is polynomial in n and k for solving this problem. The reason for this statement is that any connected graph is also a k-edge-augmented tree for a sufficiently large value of k, and already the problem `md` is NP-hard for general graphs.

5.1 Preprocessing Step

Our preprocessing step uses the following procedure that can be applied on a core vertex u with p legs. Recall that a leg is a path consisting of at least one vertex, starting at a leaf and ending just before a core vertex (at a neighbor of a core vertex), that is, a path from a leaf to a core vertex where all vertices except for the leaf are path vertices, for $p \geq 2$. Similarly to trees, consider subsets of $p - 1$ vertices belonging to the legs of u, at most one vertex per leg. We choose L_u to be a set of minimum cost among the sets that satisfy these constraints. We will place landmarks at L_u and remove from the graph every vertex belonging to a leg of u that contains a vertex of L_u (that is, the leg that does not contain a vertex of L_u is not removed). In the remaining graph we change the cost of u to zero (i.e., we define a new cost function), and thus allow the solution to place a landmark at u without increasing its cost. Since we already

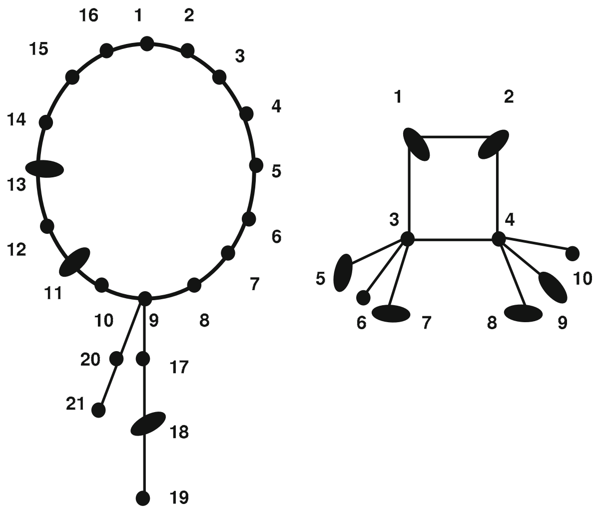

Fig. 2 An example of two graphs, where *ellipse shaped vertices* are candidate landmarks. For the graph on the *left* hand side, if both legs of vertex 9 are removed, the landmark set that was obtained for the cycle together with one landmark that was selected for one leg are not a landmark set, as $d_{8,\ell} = d_{20,\ell}$ for each one of the three landmarks. For the graph on the *right* hand side, there is no need to augment a solution obtained in a similar way by any landmarks, as it is already a landmark set

dealt with trees in Sect. 2.2, we assume that the input graph is not a tree. As we saw in the case of trees, the remaining leg does not always require a landmark, however, since we will consider different graph structures, in some cases removing this leg and finding a landmark set for the remaining graph results in a set of vertices that is not a landmark set. This is the reason for not removing the remaining leg, and our algorithm will continue to search fora landmark set for the remaining graph including this leg (see Fig. 2 for examples). Note that one core vertex may be considered multiple times (see Fig. 3).

The following lemma holds in fact also for trees that are not paths (but in order to prove it for trees as well, the special case of a spider graph, which is a graph with exactly one core vertex, should be treated separately, and since we already discussed trees, we will not discuss them in this section).

Lemma 5.1 *Denote by $G = (V, E)$ the input graph that is not a tree, and by $G' = (V', E')$ (where $V' \subsetneq V$) the graph obtained after applying the procedure above on a core vertexu (that is, V' is the set of vertices after removing all the legs of u but one). Let L_u denote the set of landmarks that we placed on the removed legs of u, and let L' denote the set of landmarks in an optimal solution of the remaining graph. Then, $L = L_u \cup L' \setminus \{u\}$ is an optimal landmark set in G.*

Proof We first show that L is a feasible landmark set. $L' \setminus \{u\} \neq \emptyset$ because G' is not a path (and hence $|L'| \geq 2$). Moreover, $L_u \neq \emptyset$ since u has at least two legs. Let $v, v' \in V$. We will show that L contains a landmark which separates v from v'. First, consider the case where $v, v' \in V'$. Note that v or v' (or both of them) may belong to the remaining leg of u. For every landmark $\ell \in L'$, the shortest paths in G from ℓ to either v or v' are exactly the shortest paths in G' from ℓ to v and v', respectively. Since L' is a feasible landmark set in G', we conclude that either there exists $\ell' \in L'$ ($\ell' \neq u$) that separates v from v' (and in this case $\ell' \in L$), or $u \in L'$ and u separates v from v'. In the last case, let $\ell \in L_u$ be an arbitrary landmark of L_u. Then, ℓ separates v from v' because $d_{\ell,v} = d_{\ell,u} + d_{u,v} \neq d_{\ell,u} + d_{u,v'} = d_{\ell,v'}$.

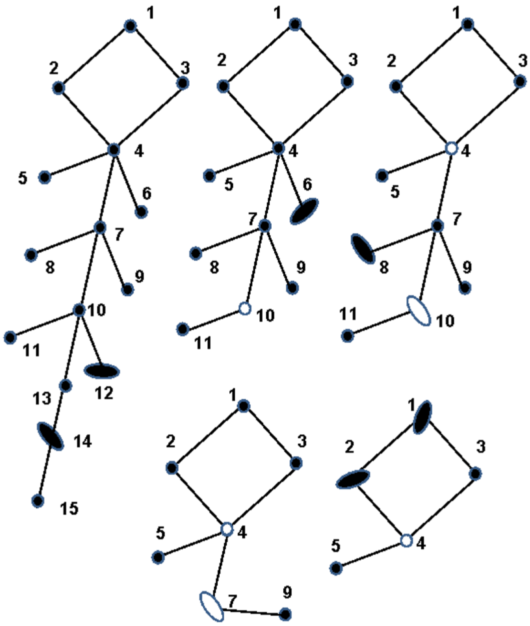

Fig. 3 The preprocessing step applied to the graph on the left hand side. *Ellipse shaped vertices* are landmarks selected in the current iteration. *White vertices* are those whose weights was set to zero in previous iterations, while *black* ones still have their original weights. The preprocessing is applied as follows. First, the legs of vertex 10 are considered, and vertices 12 and 14 are chosen as landmarks, their legs are removed, and the weight of vertex 10 is set to zero. In the graph in the *middle* (*top*) the legs of vertex 4 are considered, a landmark is placed at vertex 6 and the leg is removed. The weight of vertex 4 is set to zero. In the graph on the *right* hand side (*top*), the legs of vertex 7 are considered, and landmarks are placed at vertices 8 and 10 (but the landmark at 10 will be removed when the process returns from the recursive call that started after the legs of 10 were removed). Two legs of 7 are removed and the weight of vertex 7 is set to zero. In the next graph (*bottom middle* graph) the only core vertex is 4, and we place a landmark at vertex 7 (so vertex 4 was considered again now). In the last graph, just one core vertex with one leg remains (vertex 4). After finding a landmark set for it, the recursive calls end one by one, and the landmark set is {1,2,6,8,12,14}

Next consider the case where $v, v' \in V \setminus V'$. Note that L_u is a feasible landmark set for the spider graph consisting of u and its legs (even including the remaining leg of u, which is part of V') since it is a tree (see Sect. 2.2). Thus, there exists a landmark $\ell \in L_u$ that separates v from v'.

Finally, consider the case where $v' \in V'$ whereas $v \in V \setminus V'$. Assume by contradiction that v and v' are not separated by the landmarks in $L_u \cup L' \setminus \{u\}$. First note that the leg of v contains a landmark $\ell \neq u$. Since ℓ does not separate v from v', $d_{\ell,v} = d_{\ell,v'}$. We show that v is closer to the leaf of its leg than ℓ, that is, the path from u to ℓ does not traverse v. If this is not the case, then we get $d_{\ell,v'} = d_{\ell,v} + d_{v,u} + d_{u,v'} > d_{\ell,v}$, since $v \neq u$, contradicting the assumption that $d_{\ell,v} = d_{\ell,v'}$. Therefore, since $u \neq \ell$, $d_{\ell,v} < d_{u,v}$, and since the path from v' to ℓ is via u, $d_{\ell,v'} > d_{u,v'}$. Let $w \in L' \setminus \{u\}$. Then w separates v from v' because (using the triangle inequality and the derived inequalities for the distances) $d_{w,v'} \leq d_{w,u} + d_{u,v'} < d_{w,u} + d_{\ell,v'} = d_{w,u} + d_{\ell,v} < d_{w,u} + d_{u,v} = d_{w,v}$. Thus L is indeed a feasible landmark set.

To show that L is optimal, first note that we must place landmarks of total cost at least $\sum_{\ell \in L_u} c(\ell)$ in the legs of u, because placing a landmark at a vertex that is not in a leg of u cannot separate a pair of vertices from legs of u that are neighbors of u. Thus, any feasible solution must contain at least one landmark in at least $p-1$ of the legs of u, and we chose the cheapest such set.

First consider the case where the optimal solution for G places a landmark in each one of the legs that we removed. Assume that by paying a total cost of $\sum_{\ell \in L_u} c(\ell)$ we could place a landmark at each of the vertices of $V \setminus V'$ as well as placing a landmark at u. This is clearly a super-optimal scenario, and in order to show the optimality of L it suffices to show that we must place landmarks of additional total cost $\sum_{\ell \in L' \setminus \{u\}} c(\ell)$. Note that if a pair of vertices $v, v' \in V'$ cannot be separated by u, then they cannot be separated by any of the vertices in the removed legs. Therefore, by the optimality of L' in G', and since L' is computed under the assumption $c(u) = 0$, we must pay at least $\sum_{\ell \in L' \setminus \{u\}} c(\ell)$ for the additional landmarks.

Next, consider the case where there is a removed leg in which the optimal solution for G (denoted by OPT) does not place a landmark. Let x denote the neighbor of u along this leg. We create a new solution SOL from OPT by removing a landmark ℓ from the leg that we did not remove and placing a landmark at the member of L_u along the leg in which OPT does not place any landmarks. By the choice of the algorithm of landmarks on the legs of u, the cost of SOL is at most the cost of OPT. Therefore, using the first case, it suffices to show that SOL is a feasible landmark set. Using the first part of the claim, it suffices to show that $SOL' = SOL \cap V' \cup \{u\}$ is a feasible landmark set for G'. Assume by contradiction that SOL' does not separate v from v' (where $v, v' \in V'$). If both v and v' are on the (unique) leg of u in G', then they are separated by u. If both v and v' are not on the leg of u, then since u does not separate v from v', any landmark placed on a leg of u does not separate v from v', and by the feasibility of OPT, we conclude that $OPT \cap SOL'$ contains a landmark that separates v from v'. Thus, without loss of generality, we assume that v is on the leg of u and v' is not on the leg of u. For every landmark $\ell \in OPT$ that is not on the leg of u that contains v, we have $d_{v,\ell} = d_{v,u} + d_{u,\ell}$, and $d_{v',\ell} \leq d_{v',u} + d_{u,\ell}$. Since no such inequality for a landmark $\ell \in OPT \cap SOL'$ can be a strict inequality, we conclude that for every landmark ℓ not on the leg of u, there is a shortest path from ℓ to v' that traverses u (clearly this holds also for a landmark on a leg of u). Denote by x' the neighbor of u along a shortest path from u to v'. We conclude that for every $\ell \in OPT$, we have $d_{x',\ell} = d_{u,\ell} + 1$ and since x is along a leg in which OPT does not have a landmark, $d_{x,\ell} = d_{u,\ell} + 1$. Therefore, OPT does not separate x from x' and this contradicts the feasibility of OPT. □

We apply this preprocessing on one vertex at a time until there is no vertex which has at least two legs, and in the remaining graph every vertex has at most one leg.

5.2 The Case $k = 1$

The unweighted version of computing $\mathtt{md}(G)$ for the case of 1-edge-augmented tree is discussed in [4] (where such graphs are called unicycles). Here we consider the weighted case. If $k = 1$, then at the end of the preprocessing phase we are left with a cycle C where some of its vertices have legs (at most one leg for each vertex of C). We call such graphs *extended hairy cycles*.[1] Recall that some of the vertices of G may have zero cost resulting from the preprocessing step, and if we choose to place a landmark at such a vertex u, then the solution returned by the algorithm skips it (but has at least one landmark in the tree-like part connected to u that was removed in the preprocessing step). In what follows we only consider the graph G resulting from the preprocessing. We next characterize a minimal landmark set for extended hairy cycles. This characterization will show that every minimal landmark set for the resulting graph has at most three vertices. Thus by enumerating all subsets consisting of two or three vertices we can choose the cheapest feasible landmark set and solve wmd in polynomial time (as mentioned earlier, at least two landmarks are required for any graph that is not a path).

We denote by n_C the number of vertices in C. For a vertex v, we let v' be its *cycle-vertex*, defined as follows. v' is the closest vertex in C to v, that is if $v \in C$ then $v' = v$ and otherwise$v$ belongs to some leg of a vertex in C, and we let v' denote this vertex of C that is connected to the leg. Consider a path P from u to v, then its C-length is defined as the number of edges of C which belong to P. We say that a path P from u to v is a clockwise path if it traverses the edges in $P \cap C$ in a clockwise order, and otherwise it is a counterclockwise path. We say that two vertices are non-opposite in G if their cycle-vertices are distinct, and the C-length of the shortest path from u to v is not equal to $\frac{n_C}{2}$.

Lemma 5.2 *Let $L \subseteq V$ be a feasible minimal landmark set. There is no cycle-vertex u such that L contains a pair of vertices v_1, v_2 whose cycle-vertex is u.*

Proof Assume by contradiction that there is a minimal landmark set L which contains a pair of vertices v_1 and v_2 with a common cycle-vertex u. Without loss of generality we assume that $d_{u,v_1} < d_{u,v_2}$. We first argue that L contains another vertex w whose cycle-vertex w' is not u. If this is not the case (that is, all vertices of L have the same cycle-vertex, u), then the two neighbors of u on the cycle cannot be separated by a vertex in L.

We show that $L' = L \setminus \{v_1\}$ is a feasible landmark set. Let $x_1, x_2 \in V$. Assume by contradiction that x_1 and x_2 are not separated by L'. Since L is a feasible landmark set, we conclude that $d_{v_1,x_1} \neq d_{v_1,x_2}$. Denote by x_i' the cycle-vertex of x_i for $i = 1, 2$. If $x_1', x_2' \neq u$, then $d_{x_i,v_2} = d_{x_i,v_1} + d_{v_1,v_2}$ for $i = 1, 2$, and this is a contradiction. Next, assume that $x_1' = x_2' = u$. Then, $d_{x_1,u} \neq d_{x_2,u}$, and $d_{w,x_i} = d_{w,u} + d_{u,x_i}$, for $i = 1, 2$, a contradiction as well. Finally, assume that $x_1' = u$ whereas $x_2' \neq u$. Since $d_{v_2,x_1} = d_{v_2,x_2}$, the path from x_1 to u traverses v_2 (as otherwise $d_{x_2,v_2} = d_{x_2,x_1} + d_{x_1,v_2}$, contradicting $d_{v_2,x_1} = d_{v_2,x_2}$ as $x_1 \neq x_2$). Then, $d_{w,x_2} \leq d_{w,u} + d_{u,x_2} < d_{w,v_2} + d_{v_2,x_2} = d_{w,v_2} + d_{v_2,x_1} = d_{w,x_1}$ where the first inequality is the triangle inequality, the second inequality holds because $v_2 \neq u$, the first equality holds because v_2 does not separate x_1 from x_2, and the second equality holds because the shortest path from x_1 to w traverses v_2. Again, we get a contradiction, thus the claim follows. □

[1] In [19], a hairy cycle is defined to be a cycle such that each of its vertices has a leg of one vertex. Here we allow an arbitrary number of vertices in each leg (for each cycle vertex) including the option of zero vertices (in such a case, the considered cycle vertex does not have a leg).

In what follows, we focus on a minimal landmark set, and thus assume that it does not contain two vertices with a common cycle-vertex.

Lemma 5.3 *If $L \subseteq V$ contains a pair of non-opposite vertices u_1, u_2, then for every pair of vertices $x_1, x_2 \in C$, there exists $w \in \{u_1, u_2\}$ that separates x_1 from x_2.*

Proof Let the vertices $u'_1, u'_2 \in C$ be the cycle-vertices of u_1, u_2, respectively. Since u_1 and u_2 are non-opposite vertices, so are u'_1 and u'_2. Then, $d_{u_i,x_j} = d_{u_i,u'_i} + d_{u'_i,x_j}$ for $i, j \in \{1, 2\}$ (this equality holds even if $x_j = u'_i$). Therefore, it suffices to show that either u'_1 or u'_2 separates x_1 from x_2. This claim holds since the shortest paths from u'_i to x_j traverse only edges of C (for $i, j \in \{1, 2\}$), and therefore the claim holds by Lemma 2.4 (applied to C). □

Corollary 5.4 *If $L \subseteq V$ consists of at least three vertices, no pair of which have the same cycle-vertex, then for every pair of vertices $x, y \in C$ there is $\ell \in L$ that separates x from y.*

We next define a *covering of the legs by landmarks*. We say that a landmark ℓ clockwise-covers (counterclockwise-covers) the leg of a vertex u if one of the following conditions hold: Either ℓ is one of the vertices of the leg of u, or the clockwise path (counterclockwise path) from u to ℓ has C-length of at least 1 and at most $\frac{n_C+1}{2}$ (recall that u is not considered to be a part of its leg and indeed it cannot cover its own leg). We say that a leg is *covered* by $L \subseteq V$ if there is a landmark in L that clockwise-covers the leg and (perhaps another) landmark in L which counterclockwise-covers the leg.

Lemma 5.5 *Let L be a set of vertices, such that $|L| \geq 3$ and no two of which have the same cycle-vertex. L is a feasible landmark set if and only if every leg is covered by L.*

Proof First assume that there is an uncovered leg. Without loss of generality assume that the leg of a vertex u of C is not clockwise-covered. That is, the C-length of any clockwise path from u to any landmark in L is either zero or at least $\frac{n_C}{2} + 1$, and there is no landmark at any vertex of the leg of u. Consider a pair of vertices x, y where x is the neighbor of u along its leg, and y is the neighbor of u along C such that the path consisting of a single edge from u to y is a clockwise path. Then, for every $\ell \in L$ we have $d_{x,\ell} = 1 + d_{u,\ell}$ (since the leg of u does not contain a landmark) and $d_{y,\ell} = 1 + d_{u,\ell}$ (since the clockwise path from y to ℓ is at least as long as the counterclockwise path, andthat last path uses the edge from y to u). Therefore, there is no $\ell \in L$ that separates x from y, that is L is infeasible landmark set.

For the other direction, assume that all legs are covered by L. Let $x, y \in V$. First, consider the case that x and y have a common cycle-vertex u. Let $v \in L$ be such that the cycle-vertex of v is not u. Then, $d_{v,x} = d_{v,u} + d_{u,x}$ and $d_{v,y} = d_{v,u} + d_{u,y}$. Since $x \neq y$, $d_{u,x} \neq d_{u,y}$, and therefore v separates x from y. In the remaining cases we assume that x and y have distinct cycle-vertices. If $x, y \in C$, then by Corollary 5.4, there exists a landmark $\ell \in L$ that separates x from y.

Next consider the case where $x \notin C$. Let z be the cycle-vertex of x. In the case that x belongs to a leg that has a landmark ℓ, if ℓ does not separate x from y, then the path from x to the cycle traverses ℓ. Then, any other landmark ℓ' which has a different cycle-vertex from those of x and y, separates x from y because $d_{\ell',x} = d_{\ell',\ell} + d_{\ell,x} = d_{\ell',\ell} + d_{\ell,y} > d_{\ell',z} + d_{z,y} \geq d_{\ell',y}$, where the second equality holds because ℓ does not separate x from y.

Without loss of generality (by possibly swapping the roles of x and y) we are left with the case that if $y \notin C$, then the legs of x and y do not contain a landmark, and if $y \in C$, then the leg of x does not contain a landmark. In the first case $\ell \in L$ separates x from y if and only if ℓ separates the parent x' of x from the parent y' of y (where the parent of a vertex of a leg is its neighbor that is closer to C or on C). Applying this transformation sufficiently many times, we are left (after renaming x and y) with one remaining case in which $x \notin C$ and $y \in C$, the leg of x does not contain a landmark (since the case that $x, y \in C$ was already treated), and $z \neq y$, that is, y is not the cycle-vertex of x.

The leg of z is covered, and hence there is a landmark ℓ that clockwise-covers the leg of z and a landmark ℓ' that counterclockwise-covers this leg. Since the leg of x does not contain a landmark, ℓ, ℓ' are not on the leg of z. We assume that the cycle-vertex of ℓ is the cycle-vertex of a landmark such that the C-length of the clockwise path from z to a landmark is minimized, and similarly for ℓ' (with respect to the counterclockwise paths). Note that by this assumption, the clockwise path from z to ℓ and the counterclockwise path from z to ℓ' do not share any edge, and $\ell \neq \ell'$ (since $|L| \geq 3$).

Let $\ell'' \in L \setminus \{\ell, \ell'\}$. Assume by contradiction that the three landmarks ℓ, ℓ', ℓ'' do not separate x from y. Let q, q', q'' be the cycle-vertices of ℓ, ℓ', ℓ'', respectively. By assumption, q, q', q'' are three distinct vertices. First assume that y belongs to either the clockwise path from z to q or to the clockwise path from q' to z (possibly $y = q$ or $y = q'$, but $y \neq z$). Note that the clockwise path from z to q is either the shortest path from z to q or the C-length of this clockwise path is exactly $\frac{n_C+1}{2}$. Thus, considering the subpath of this path starting from the vertex after z, we get a shortest path, and all its subpaths are shortest paths as well. Without loss of generality, we can assume that y belongs to the clockwise path from z to q. In this case $d_{\ell,y} = d_{\ell,q} + d_{q,y} \leq d_{\ell,q} + d_{q,z} = d_{\ell,z} < d_{\ell,z} + 1 \leq d_{\ell,x}$ where the first inequality holds because the shortest path from y to q is the clockwise path from y to q (or it is empty) and since $z \neq y, d_{q,y} \leq d_{q,z}$ even if the clockwise path from z to q is not the shortest path (in the only case where it is not the shortest path its length is exactly $\frac{n_C+1}{2}$ and $d_{q,z} = \frac{n_C-1}{2}$ while $d_{q,y} \leq \frac{n_C-1}{2}$), and the second equality holds since if $q \neq \ell$, then any path of ℓ to the cycle traverses q. Therefore ℓ separates x from y.

Thus, in the remainder of the proof y belongs to the path from ℓ to ℓ' that does not traverse z. First consider the case where $\ell'' = z$ (this is the unique remaining option in the case $q'' = z$). Without loss of generality the shortest path from y to z traverses q. Then $d_{q,x} > d_{z,x} = d_{z,y} > d_{q,y}$ and this is a contradiction. Therefore, $q'' \neq q, q'$ is on the clockwise path from q to q' since $q'' \neq z$ and by our choice of ℓ and ℓ' as the closest landmarks to z in the clockwise and counterclockwise directions, respectively (when considering the C-length of a path).

Without loss of generality assume that one of the paths from y to z traverses both q and q'' (note that $y \neq q''$ as otherwise $d_{\ell'',y} = d_{\ell'',q''} < d_{\ell'',x}$), and the other path from y to z traverses q'. First we argue that the shortest paths from x to q, q', q'' do not traverse y as otherwise y is closer to the corresponding landmark than x. Since the shortest path from x to q'' does not traverse y, we conclude that $d_{x,q} < d_{x,q''}$. Therefore, the shortest path from y to q does not traverse q'' (as otherwise $d_{y,q''} < d_{y,q}$ contradicting the assumption that ℓ, ℓ'' do not separate x from y) and hence traverses z. The clockwise path from y to q is the shortest path, and therefore every subpath of it is a shortest path. We find that $d_{y,q'} < d_{y,z} = d_{x,z} < d_{x,q'}$, where $d_{y,z} = d_{x,z}$ as $d_{y,\ell} = d_{y,z} + d_{z,q} + d_{q,\ell}$ and $d_{x,\ell} = d_{x,z} + d_{z,q} + d_{q,\ell}$. This is a contradiction to the assumption that ℓ' does not separate x from y. □

Lemma 5.6 *Let L be a minimal landmark set of an extended hairy cycle. Then $|L| \leq 3$.*

Proof Assume by contradiction that L is a minimal landmark set and that L has at least four landmarks denoted as u_1, u_2, u_3, u_4. We denote the cycle-vertex of u_i by u'_i, for $i = 1, 2, 3, 4$ (by Lemma 5.2, all these vertices are distinct). Without loss of generality we assume that u'_1, u'_2, u'_3, u'_4 appear in this order along the cycle clockwise. Then without loss of generality the length of the clockwise path from u'_1 to u'_3 has C-length of at most $\frac{n_C}{2}$ (i.e., it is a shortest path and contains u'_2). Therefore, every subpath of the clockwise path from u'_1 to u'_3 has at most $\frac{n_C}{2}$ edges.

We next claim that $L' = L \setminus \{u_2\}$ is a feasible landmark set. By Lemma 5.5, we conclude that it suffices to show that every leg that was covered by L is covered by L'. Assume that a vertex $x \in C$ has a leg. First assume that $x = u'_2$. Then, u_1 clockwise-covers the leg of x and u_3 counterclockwise-covers the leg of x. Next, assume that $x \neq u'_2$, and assume that x is clockwise-covered by u_2 in L (the case of counterclockwise-covered is analogous). First assume that x is on the clockwise path from u'_1 to u'_2 (possibly $x = u'_1$). Then, u_3 clockwise-covers the leg of x. Otherwise, the clockwise path from x to u'_1 is a subpath of the clockwise path from x to u'_2, and therefore u_1 clockwise-covers the leg of x. □

To summarize, our algorithm for computing $\texttt{wmd}(G)$ where G is a 1-edge-augmented tree is to apply the preprocessing step, and afterwards try all possibilities of sets L such that $|L| \leq 3$, and for each of them test its feasibility in linear time by creating lists of legs that are covered by each member of L. We pick a cheapest solution (with the smallest possible number of vertices) among the feasible solutions that were found. Clearly, the algorithm runs in polynomial time ($O(n^4)$) and computes a cheapest minimal feasible landmark set. As for the preprocessing step, it is easy to implement it using the same running time; at each time we can run DFS and detect a core vertex of maximum depth. The legs of this vertex are dealt with (i.e., a set of landmarks is computed for them as it is done for trees), and the legs at which landmarks were placed are removed. Obviously, if the graph is an extendedhairy cycle to begin with, then the same algorithm is applied for it, and its running time is $O(n^4)$. Therefore, we have established the following.

Proposition 5.7 *Given an extended hairly cycle $G = (V, E)$,there exists an algorithm that solves* wmd *in time $O(n^4)$.*

Proposition 5.8 *Given an 1-edge-augmented tree $G = (V, E)$,there exists an algorithm that solves* wmd *in time $O(n^4)$.*

5.3 The General Case

Assume that $G = (V, E)$ is the graph resulting from applying the preprocessing step. That is, in G every core vertex has at most one leg. The case of $k = 1$ is already solved, and here we assume that $k \geq 2$ is a fixed constant. We next define a subgraph of G called the base graph $G_b = (V_b, E_b)$ resulting from G by removing the vertices of all legs. We next characterize the structure of this base graph. That is, we will show that it consists of $O(k)$ edge disjoint paths connecting core vertices where all internal vertices are path vertices.

5.3.1 The Structure of the Base Graph

For a vertex $u \in V_b$, we denote by $deg_b(u)$ its degree in G_b. Note that G_b contains no leaves, and thus the degree of every vertex is at least 2. Moreover, since G_b is a connected subgraph of G and every cycle of G belongs to G_b as well. G_b results from a tree T by adding exactly k edges. Thus the following claim follows.

Claim 5.9 $\sum_{u \in V_b} (deg_b(u) - 2) = 2k - 2$.

Proof First note that the tree T has $|V_b| - 1$ edges, and thus the number of edges in G_b is $|V_b| + k - 1$. Therefore, $\sum_{u \in V_b} deg_b(u) = 2|V_b| + 2k - 2$ and the claim follows. □

Given a connected graph with no leaves, we can decompose it into a set of edge disjoint paths where every internal vertex of such a path has degree 2 (in the graph) and every end-vertex of a path is a core vertex (in the graph). The decomposition is done by splitting every core vertex u into $deg_b(u)$ copies, one for each one of its edges. Since for a vertex u such that $deg_b(u) \geq 3$ we have $deg_b(u) \leq 3(deg_b(u) - 2)$, we find $\sum_{u \in V_b : deg_b(u) \geq 3} deg_b(u) \leq 6k - 6$. Therefore, the number of paths that G_b is decomposed into is $\frac{1}{2} \sum_{u \in V_b : deg_b(u) \geq 3} deg_b(u) \leq 3k - 3$ (where a cycle is also considered to be a path), and we conclude the following.

Lemma 5.10 *The base graph G_b is decomposed into $q \leq 3k - 3$ edge disjoint paths, where every internal vertex of a path has degree 2 in G_b, and the end-vertices of such a path are core vertices in G_b.*

Given a vertex $v \in V$, we define its base vertex as the vertex $v' \in V_b$ that is the closest to v. Given the path decomposition defined above, we associate each vertex v in V with one of the paths in the following way. If the base vertex of v belongs to exactly one of the paths, then we associate v with this path. Otherwise, the base vertex v' of v is a core vertex in G_b, and we associate every vertex of G whose base vertex is v' with an arbitrary path that is incident to v'. Later, we will show that a minimal landmark set has at most one landmark that is associated with each base vertex.

5.3.2 Bounding the Number of Landmarks Associated with One Path

Next, we consider a minimal feasible landmark set L, and one specific path P in the path decomposition of G_b. Our goal is to bound the number of landmarks in L which are associated with P. The following lemma generalizes Lemma 5.2 to the case of $k \geq 2$.

Lemma 5.11 *A minimal landmark set L does not have a pair of vertices v, v' with a common base vertex u.*

Proof First we show that L must contain a landmark whose base vertex is not u. Since G_b has no leaves, if L only contains vertices whose base vertex is u, then the (at least two) neighbors of u in G_b cannot be separated by a landmark in L. Let ℓ be a landmark whose base vertex is different from u.

Assume without loss of generality that $d_{v',u} > d_{v,u}$. Let $L' = L \setminus \{v\}$. We argue that L' is a feasible landmark set. Assume by contradiction that there is a pair of vertices x_1, x_2 which are separated by L but not by L'. Then, v separates x_1 from x_2, but v' and ℓ do not. If none of x_1, x_2 belongs to the leg of u then $d_{x_i,v'} = d_{x_i,v} + d_{v,v'}$, and this is a contradiction. If x_1 belongs to the leg of u and x_2 does not belong to this leg, then since $d_{x_1,v'} = d_{x_2,v'}$, we conclude that the shortest path from x_2 to v' does not traverse x_1. Then, $d_{x_1,u} > d_{x_1,v'} = d_{x_2,v'} > d_{x_2,u}$ because $v' \neq u$. We conclude that $d_{\ell,x_2} \leq d_{\ell,u} + d_{u,x_2}$ and $d_{\ell,x_1} = d_{\ell,u} + d_{u,x_1}$ and therefore $d_{\ell,x_1} = d_{\ell,u} + d_{u,x_1} > d_{\ell,u} + d_{u,x_2} \geq d_{\ell,x_2}$ contradicting the assumption that ℓ does not separate x_1 from x_2. In the last case x_1 and x_2 belong to the leg of u. In this case $d_{u,x_1} \neq d_{u,x_2}$ and ℓ separates x_1 from x_2 because for $i = 1, 2$, we have $d_{\ell,x_i} = d_{\ell,u} + d_{u,x_i}$. □

Lemma 5.12 *Let L be a minimal feasible landmark set, and let P be a path in the path decomposition of G_b. Then, the number of vertices in L which are associated with P is at most six.*

Proof Denote by s and t the two end-vertices of P. For a subpath P' of P we say that a vertex x is associated with P' (in addition to being associated with P) if the base vertex of x is in P', and we denote the length of P' by $d(P')$.

Assume by contradiction that L has at least seven vertices associated with P denoted by w_i(for $1 \leq i \leq 7$), and we denote by w'_i the base vertex of w_i. By the previous lemma, $w'_i \neq w'_j$ for $i \neq j$. Without loss of generality, assume that $w'_1, w'_2, \ldots, w'_7$ appear in this order when we traverse P from s to t. If the length of the subpath of P connecting w'_2 to w'_4 is smaller than $\frac{d(P)}{2}$, then we let $u_i = w_i$ for $i = 1, 2, 3, 4$ and $u_5 = w_7$. Otherwise, we must have that the length of the subpath of P connecting w'_4 to w'_6 is smaller than $\frac{d(P)}{2}$, and in this case we let $u_1 = w_1$ and for $i = 2, 3, 4, 5$, we let $u_i = w_{i+2}$. Let u'_i be the base vertex of u_i for $i = 1, 2, 3, 4, 5$. Let P' be the subpath of P starting at u'_2 and ending at u'_4. Therefore, the end-vertices of P' are not s or t, and $d(P') < \frac{d(P)}{2}$. Let $L' = L \setminus \{u_3\}$, and we argue that L' is a feasible landmark set. Assume by contradiction that there is a pair of vertices x, y such that L' does not separate x from y, whereas L does. Thus $d_{x,u_3} \neq d_{y,u_3}$ but for $i = 1, 2, 4, 5$ we have $d_{x,u_i} = d_{y,u_i}$. Denote by x' and y' the base vertices of x and y, respectively. If $x' = y'$, then $d_{x,x'} \neq d_{y,y'}$. Moreover, there exists a value $i \in \{1, 5\}$ such that $u'_i \neq x'$, and therefore $d_{x,u_i} = d_{x,x'} + d_{x',u_i}$ and $d_{y,u_i} = d_{y,y'} + d_{y',u_i}$, contradicting the assumption that $d_{x,u_i} = d_{y,u_i}$. In what follows we assume that $x' \neq y'$.

We first assume that neither x nor y is associated with P'. For $i = 1, 2, 4, 5$, since $d_{x,u_i} = d_{y,u_i}$ and x and y are not associated with P', we have $d_{x,u'_i} = d_{y,u'_i}$ for

$i = 2, 4$. Assume without loss of generality that a shortest path from x to u_3 traverses u'_2 (it must traverse either u'_2 or u'_4). If the shortest path from y to u_3 traverses u'_2 as well, then we have $d_{y,u_3} = d_{y,u'_2} + d_{u'_2,u_3} = d_{x,u'_2} + d_{u'_2,u_3} = d_{x,u_3}$ contradicting the assumption that u_3 separates x from y. Therefore, the shortest path from y to u_3 must traverse u'_4. However, since the shortest path from y to u_3 traverses u'_4, we conclude that $d_{y,u_3} = d_{y,u'_4} + d_{u'_4,u_3} = d_{x,u'_4} + d_{u'_4,u_3} \geq d_{x,u_3}$. Since the shortest path from x to u_3 traverses u'_2, we find $d_{x,u_3} = d_{x,u'_2} + d_{u'_2,u_3} = d_{y,u'_2} + d_{u'_2,u_3} \geq d_{y,u_3}$ and we get a contradiction to $d_{y,u_3} \neq d_{x,u_3}$.

Next, assume that x and y are both associated with P'. We let $x_1 = x$, $x_2 = y$, $x'_1 = x'$, and $x'_2 = y'$. Recall that $d(P') < \frac{d(P)}{2}$ and hence the shortest path between a pair of vertices that are associated with P' does not traverse any other vertices of P (which are not on P'). Assume without loss of generality that x'_1 is closer to u'_2 (along P') than x'_2 (note that it is possible that $x'_1 = u'_2$ or $x'_2 = u'_4$ or both). Therefore, $d_{x_1,u_4} = d_{x_1,x'_2} + d_{x'_2,u_4}$, $d_{x_2,u_4} \leq d_{x_2,x'_2} + d_{x'_2,u_4}$ (this is implied by the triangle inequality and it is not necessarily an equality, as it may be the case that $u'_4 = x'_2$) and since u_4 does not separate x_1 from x_2 (i.e., $d_{x_1,u_4} = d_{x_2,u_4}$), we conclude that $d_{x_1,x'_2} \leq d_{x_2,x'_2}$. Similarly, $d_{x_1,u_2} \leq d_{x_1,x'_1} + d_{x'_1,u_2}$, $d_{x_2,u_2} = d_{x_2,x'_1} + d_{x'_1,u_2}$ and sinceu_2 does not separate x_1 from x_2, we conclude that $d_{x_2,x'_1} \leq d_{x_1,x'_1}$. However, $d_{x_1,x'_1} < d_{x_1,x'_2} \leq d_{x_2,x'_2} < d_{x_2,x'_1} \leq d_{x_1,x'_1}$ where the strict inequalities hold because $x'_1 \neq x'_2$. Thus in this case x and y must be separated.

Finally, assume that x is not associated with P' and y is associated with P'. If the base vertex of y is u'_2 (the case $y' = u'_4$ is similar), then since u_2 does not separate x from y, we conclude that the path from x to u_2 does not traverse y and also $y \neq u'_2$. Moreover, $d_{x,u'_2} \leq d_{x,u_2}$ and $d_{y,u'_2} \geq d_{y,u_2}$, and thus $d_{y,u'_2} \geq d_{x,u'_2}$ and equality holds only if $u_2 = u'_2$. However, since any path from y to u_1 traverses u'_2, the following holds by the triangle inequality $d_{y,u_1} = d_{x,u_1} \leq d_{x,u'_2} + d_{u'_2,u_1} \leq d_{y,u'_2} + d_{u'_2,u_1} = d_{y,u_1}$ where the last inequality holds as an equality only when $u'_2 = u_2$. Thus,$u'_2 = u_2$. Since u_3 separates x from y and $d_{y,u_3} = d_{y,u_2} + d_{u_2,u_3} = d_{x,u_2} + d_{u_2,u_3} \geq d_{x,u_3}$, we conclude that the shortest path from x to u_3 is strictly shorter than $d_{x,u_2} + d_{u_2,u_3}$. We conclude that the shortest path from x to u_3 traverses u'_4, and therefore $d_{x,u'_4} = d_{x,u_3} - d_{u_3,u'_4} < d_{x,u_2} + d_{u_2,u_3} - d_{u_3,u'_4} = d_{y,u_2} + d_{u_2,u_3} - d_{u_3,u'_4} \leq d_{y,u'_4}$ contradicting the assumption that u_4 does not separate x from y. Thus, it remains to consider the case where the base vertex of y is not u'_2 and not u'_4.

Without loss of generality, assume that the shortest path from x to u_3 traverses u'_2. $y' \neq u'_2, u'_4$ and y' is on the subpath P'. Since the unique shortest path from y' to u'_4 is a subpath of P', it does not traverse u'_2, $d_{x,u'_4} = d_{y,u'_4} < d_{y,u'_2} + d_{u'_2,u'_4} = d_{x,u'_2} + d_{u'_2,u'_4}$. Therefore, every shortest path from x to u'_4 does not traverse u'_2. If the shortest path from y to u'_1 traverses u'_2, then $d_{y,u'_1} = d_{y,u'_2} + d_{u'_2,u'_1} = d_{x,u'_2} + d_{u'_2,u'_1}$. However, we argue next that no shortest path from x to u'_1 traverses u'_2. Assume by contradiction that a shortest path from x to u'_1 traverses u'_2. Then, since x' is not on P', we conclude that this shortest path from x' to u'_1 traverses u'_4 and u'_3 before it traverses u'_2 (recall that no shortest path from x to u'_4 traverses u'_2). This is a contradiction to the assumption that the shortest path from x to u'_3 traverses u'_2. Therefore, $d_{x,u'_1} < d_{x,u'_2} + d_{u'_2,u'_1}$ contradicting the assumption that u_1 does not separate x from y. Therefore, it remains to consider the case where the path from y to u'_1 does not traverse u'_2, but it traverses

u'_4 and u'_5 along the way, and so $d_{y,u'_1} = d_{y,u'_4} + d_{u'_4,u'_5} + d_{u'_5,u'_1}$. Since u_4, u_5, u_1 do not separate x from y, we conclude that $d_{x,u'_1} = d_{x,u'_4} + d_{u'_4,u'_5} + d_{u'_5,u'_1}$ and thus there exists a shortest path from x to u'_1 that traverses u'_4 and u'_5 along its way (in this order) and thus also u'_2 (before arriving at u'_4), contradicting the claim shown above that any shortest path from x to u'_4 does not traverse u'_2. □

The following corollary follows immediately from Lemmas 5.10 and 5.12.

Corollary 5.13 *Let L be a minimal landmark set of a graph G which results from a k-edge-augmented tree by the preprocessing step (where $k \geq 2$). Then $|L| \leq 18k - 18$.*

To summarize, our algorithm for computing $\mathtt{wmd}(G)$ where G is a k-edge-augmented tree (for $k \geq 2$) is to apply the preprocessing step (which can be implemented faster than the next part of the algorithm), and afterwards try all possibilities of sets L such that $|L| \leq 18k - 18$, and for each of them test its feasibility in polynomial time (using a matrix of distances), and among the feasible solutions we pick a cheapest one. Clearly, the algorithm runs in polynomial time (for a constant value of k) and computes a cheapest feasible landmark set. The running time is $O(n^{O(k)})$. Therefore, we have established the following.

Theorem 5.14 *Given a k-edge-augmented tree $G = (V, E)$ for a constant number k, there is a polynomial time algorithm for solving* wmd.

6 Wheels

6.1 Complete Wheels

In this section we consider complete wheels. A (complete) wheel on n vertices $\{1, 2, \dots, n\}$ is defined as follows. There is a cycle C over the vertices $1, 2, \dots, n-1$ (the clockwise order of the vertices along C is $1, 2, \dots, n-1, 1$), and vertex n is adjacent to all other vertices. Vertex n is called the *hub* of the wheel, whereas the other vertices are called *cycle-vertices*. All distances in G are either 1 or 2; clearly the distance between every cycle-vertex and the hub is 1, the distance of every cycle-vertex and its two neighbors along the cycle is 1, and any two non-adjacent cycle-vertices are reachable via a two-edge path through the hub. Consider a feasible landmark set L. A gap between consecutive landmarks is defined as follows: If $\ell, \ell' \in L$ are cycle-vertices such that there is no other landmark along the clockwise path from ℓ to ℓ', then the set of internal vertices of the clockwise path from ℓ to ℓ' is the *gap* (between ℓ and ℓ'), and we say that the gap is adjacent to ℓ and ℓ'. The length of the gap is the number of vertices in the gap. Here, we characterize the lengths of the gaps in a feasible solution L.

Lemma 6.1 *Assume that G is a wheel over at least 8 vertices. Let $L \subseteq V$. L is a feasible landmark set if and only if the following three conditions hold with respect to L:*

1. *There is no gap of length at least 4.*
2. *There is at most one gap of length 3.*

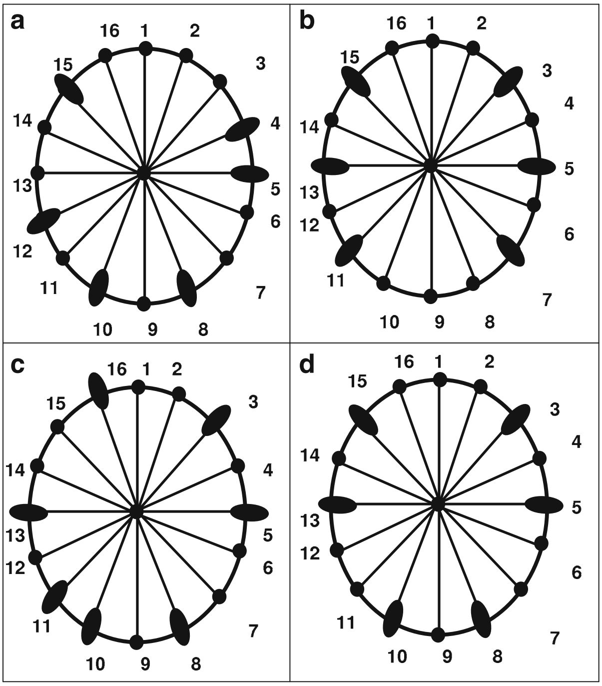

Fig. 4 An example of a complete wheel. Subsets of vertices marked with ellipses are considered. In the subset of **a**, there is a gap of length four 16, 1, 2, 3, contradicting condition 1. Indeed, $d_{1,\ell} = d_{2,\ell} = 2$ for any $\ell \in \{4, 5, 8, 10, 12, 15\}$. In the subset of **b**, there are two gaps of length three, contradicting condition 2. Indeed, $d_{1,\ell} = d_{9,\ell} = 2$ for any $\ell \in \{3, 5, 7, 11, 13, 15\}$. In **c**, there are two consecutive gaps of length two, contradicting condition 3. Indeed, $d_{1,\ell} = d_{15,\ell} = 2$ for any $\ell \in \{3, 5, 8, 10, 11, 13\}$ and $d_{1,16} = d_{15,16} = 1$. Finally, the subset of **d** satisfies the three conditions; the hub is the unique vertex of distance 1 from any landmark, vertex 1 is the unique vertex of distance 2 from any landmark, and the lists of distances of other non-landmark vertices to the landmarks $\{3, 5, 8, 10, 13, 15\}$ are $\{1, 2, 2, 2, 2, 2\}$ (vertex 2), $\{1, 1, 2, 2, 2, 2\}$ (vertex 4), $\{2, 1, 2, 2, 2, 2\}$ (vertex 6), $\{2, 2, 1, 2, 2, 2\}$ (vertex 7), $\{2, 2, 1, 1, 2, 2\}$ (vertex 9), $\{2, 2, 2, 1, 2, 2\}$ (vertex 11), $\{2, 2, 2, 2, 1, 2\}$ (vertex 12), $\{2, 2, 2, 2, 1, 1\}$ (vertex 14), and $\{2, 2, 2, 2, 2, 1\}$ (vertex 16)

3. *Two gaps of length at least two are not adjacent to a common vertex of* $L \cap C$.

Proof See Fig. 4 for an illustration of the conditions. First, assume that L is a feasible landmark set. In this part of the proof we use the fact that the hub vertex can never separate two cycle-vertices, and hence they must be separated by a landmark that is a cycle-vertex. We prove the three conditions. Assume by contradiction that at least one of the conditions does not hold. First, assume that there is a gap of length at least 4 between the landmarks ℓ_1, ℓ_2. Then, there are at least two vertices v, v' from this gap (those that are not adjacent to ℓ_1 and ℓ_2) that have distance 2 from any vertex in $L \cap C$. Thus, there is no vertex in L that separates v from v'. Next, assume that there are (at least) two gaps, each of length 3. Let v, v' be the middle vertices of these gaps. Therefore, for every $\ell \in L \cap C$ we have $d_{\ell,v} = d_{\ell,v'} = 2$. Finally, assume that the two gaps which are adjacent to $\ell \in L \cap C$ have length at least 2. Denote by v, v' the two neighbors of ℓ in C. Then, $d_{v,\ell} = d_{v',\ell} = 1$, for every $\ell' \in L \cap C$ such that $\ell' \neq \ell$ we have $d_{v,\ell'} = d_{v',\ell'} = 2$. In all cases, we conclude that there is no vertex in L that separates v from v'.

For the other direction, assume that L satisfies the three conditions, then $L' = L \setminus \{n\}$ also satisfies these conditions, and it suffices to show that L' is a feasible landmark set. Therefore, without loss of generality assume that L' satisfies the three conditions. Note that since G has at least eight vertices, there are at least seven vertices in the cycle C, and by the three conditions we conclude that $|L'| \geq 3$. Given any set of three cycle-vertices, there is exactly one vertex whose distance from all of these vertices is 1, and this unique vertex is the hub, n, so the hub is separated from every cycle-vertex (even though there is no landmark at the hub in L'). Therefore, it suffices to show that if $v, v' \in C \setminus L'$, then there is $\ell \in L'$ that separates v from v'. By the first and second conditions there is at most one vertex (the middle vertex of a gap of length 3, if it exists) whose distance from every vertex of L' is 2 (any other vertex in C is either a landmark or adjacent to a landmark). Therefore, we can assume without loss of generality that v is adjacent to $\ell \in L'$. If v' is not adjacent to ℓ, then ℓ separates v from v'. Otherwise, v and v' are cycle-vertices adjacent to ℓ (along C). By the third condition, one of their gaps contains a single vertex, so we conclude that either v or v' is adjacent to another landmark $\ell' \in L'$. Since the length of C is at least 7, then ℓ' cannot be adjacent to both v and v', and therefore it separates v from v'. Therefore L' is a feasible landmark set. □

Corollary 6.2 *A minimal landmark set L of a complete wheel G does not contain the hub.*

Based on the characterization in the above lemma, our algorithm is defined by a straightforward dynamic program. Note that without loss of generality, we can assume that G has at least 8 vertices (otherwise G has a constant size, and thus we can compute $\mathtt{wmd}(G)$ in constant time using an exhaustive enumeration). We first guess a position of a single landmark ℓ such that the clockwise path from ℓ to the next landmark along C has at most two edges. By the conditions of Lemma 6.1, in every six consecutive vertices along C there is at least one such landmark, and thus it is sufficient to enumerate six possibilities for ℓ. Without loss of generality and for the sake of presentation we assume that $\ell = n-1$. Using this assumption, our dynamic program positions landmarks along a path, and it selects the location of additional landmarks among the vertices $1, 2, \ldots, n-2$. By our assumption that the next landmark after ℓ must occur at one of the next two vertices, exactly one of the vertices 1,2 must contain a landmark (positioning landmarks at both of them does not result in an optimal solution, or it does not result in a minimal landmark set, as in such a case, the landmark of vertex 1 can be removed without violating the properties). We position the remaining landmarks using a recursive formula where every state of the dynamic program recalls whether we already used a gap of length 3 and therefore we cannot use it again.

For $i = 0, 1$, denote by $F_1^i(\ell')$ and $F_2^i(\ell')$ the minimum total cost of the landmarks in a solution for the path $P = \{1, 2, \ldots, \ell'\}$ such that ℓ' is a landmark, and such that the last gap before ℓ' has length at most 1 and at least 2, respectively, where the subpath of P from the first landmark (1 or 2) to ℓ' has i gaps of length 3. Then, the following holds (where if the argument of F_1^i or F_2^i is not positive then its corresponding value is ∞).

$$F_1^0(\ell') = \min\{F_1^0(\ell'-1), F_2^0(\ell'-1), F_1^0(\ell'-2), F_2^0(\ell'-2)\} + c(\ell'),$$
$$F_2^0(\ell') = F_1^0(\ell'-3) + c(\ell'),$$
$$F_1^1(\ell') = \min\{F_1^1(\ell'-1), F_2^1(\ell'-1), F_1^1(\ell'-2), F_2^1(\ell'-2)\} + c(\ell'),$$
$$F_2^1(\ell') = \min\{F_1^1(\ell'-3), F_1^0(\ell'-4)\} + c(\ell').$$

The starting conditions are $F_1^0(1) = c(1)$, $F_1^0(2) = c(2)$, and $F_1^1(1) = F_1^1(2) = F_2^0(1) = F_2^0(2) = F_2^1(1) = F_2^1(2) = \infty$. The goal is to compute the value $\min\{F_1^1(\ell), F_1^0(\ell), F_2^1(\ell), F_2^0(\ell)\}$. We thus conclude the following result.

Theorem 6.3 *Given a complete wheel $G = (V, E)$,there exists an algorithm that solves* `wmd` *in linear time $O(n)$.*

In [9], it is (inaccurately) stated that for a complete wheel on n vertices, the metric dimension is always 2. This was corrected in [17] where they showed the following result (which can be derived from the above conditions and a straightforward analysis for the case $n \leq 8$).

Proposition 6.4 *[17] Given a complete wheel G on n vertices,* $\mathtt{md}(G)$ *is as follows: If $n = 4, 7$,* $\mathtt{md}(G) = 3$*; If $n = 5, 6$,* $\mathtt{md}(G) = 2$*; for $n \geq 8$,* $\mathtt{md}(G) = \lfloor \frac{2n}{5} \rfloor$.

Remark 6.5 Since a complete wheel on n vertices is an $(n-1)$-edge-augmented tree, by the last proposition, we conclude that there exist k-edge-augmented trees without any legs that need $\Omega(k)$ landmarks in any feasible solution.

Note that not every k-edge-augmented tree requires $\Omega(k)$ landmarks. Any two-dimensional grid graphs requires only two landmarks in the L_1 norm (i.e., if distances are defined as distances in the corresponding graph) [14,15], while $k = \Omega(n)$ is this case as well.

6.2 The General Case

An incomplete wheel on n vertices $\{1, 2, \ldots, n\}$ is defined as follows. There is a cycle C over the vertices $1, 2, \ldots, n-1$ (the clockwise order of the vertices along C is $1, 2, \ldots, n-1, 1$), and vertex n is adjacent to some of the other vertices, but not to all of them. In what follows we consider either an incomplete or a complete wheel, and refer to it as a wheel. Vertex n is called the *hub* of the wheel, whereas the other vertices are called *cycle-vertices*. The neighbors of n are called *connectors*, and the edges incident at n are called *spokes*. We let *layer* j be the set of vertices of distance j from n, and denote it by V_j (i.e., $V_j = \{u \in V : d_{u,n} = j\}$, and thus $V_0 = \{n\}$, and V_1 is the set of connectors). For $L \subseteq V$, we say that a cycle-vertex u is close to $\ell \in L$ if $d_{\ell,u} < d_{\ell,n} + d_{n,u}$, i.e., the shortest path through the hub between ℓ and u is not a shortest path between them (note that no vertex can be close to the hub). We say that u is close to L if there is $\ell \in L$ such that u is close to ℓ (see Fig. 5). In this

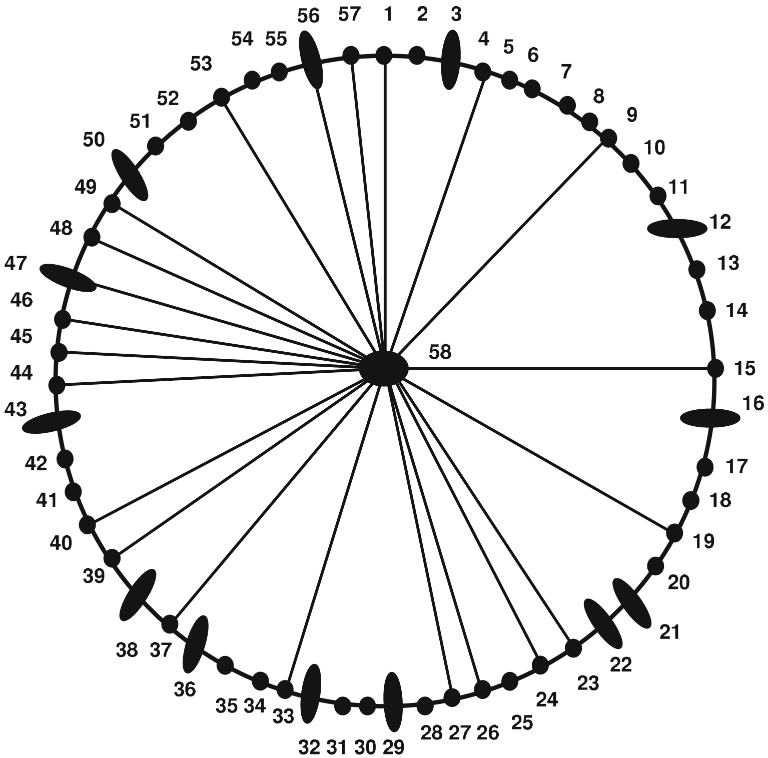

Fig. 5 An example of an incomplete wheel with 22 connectors and 58 vertices. The *ellipse shaped vertices* are landmarks of a landmark set. Vertex 53 is not close to any landmark as its distance from vertex 56 is 2 (the only shortest path traverses the hub), and its distance to vertex 50 is 3 (there are two shortest paths, one of which traverses the hub). The vertex 18 is close to the landmark 16 (with a distance 2) while 19 is not close to 16. The vertex 6 is close to the landmark 12 (with distance 6) but the vertex 18 is not close to 12. Vertex 45 is not close to landmark 47, but it is close to landmark 43

section we consider wheels with at least 22 connectors, and present a polynomial time algorithm for solving wmd for such a graph. Since wheels with at most 21 connectors are k-edge-augmented trees for a value of k such that $k \leq 21$, we conclude that we will obtain a polynomial time algorithm for solving wmd on (arbitrary) wheels. Thus let $G = (V, E)$ be a wheel with at least 22 connectors (where $n = |V| \geq 23$). We first characterize a minimal landmark set L in such graphs (some of the properties that we prove hold for smaller numbers of connectors as well).

Lemma 6.6 *Let L be a feasible landmark set. Then, for every j there is at most one vertex of V_j that is not close to L. Moreover, the vertices in V that are not close to L form a shortest path from some vertex v to the hub.*

Proof Assume by contradiction that there is a value of j such that there are two vertices $u_1, u_2 \in V_j$ that are not close to L. By the definition of closeness, we conclude that for every $\ell \in L$ and for $i = 1, 2$, we have $d_{u_i,\ell} = d_{u_i,n} + d_{n,\ell} = j + d_{n,\ell}$, and thus ℓ does not separate u_1 from u_2, contradicting the feasibility of L.

Let $v \in V$ be the vertex of a maximum layer that is not close to L. The vertex v is well-defined because by definition n is not close to L, and since we showed that each layer has at most one vertex that is not close to L. We next argue that the vertices that are not close to L form a shortest path from v to n. Assume otherwise. For every $\ell \in L$, the shortest path from ℓ to v traverses n, and continues along a shortest path from n to v. The vertices along this path are not close to ℓ, since a subpath of a shortest path is also a shortest path. □

In the example shown in Fig. 5, the path of vertices that are not close to L consists of the two vertices 53 and 58.

Lemma 6.7 *Let $\ell \in V$. The set of vertices that are close to ℓ is a proper subpath P of C containing ℓ. Moreover, consider P as a clockwise path from a vertex u to a vertex v, then the subpath of P from u to ℓ (including u, ℓ) has at most two connectors, and the subpath of P from ℓ to v has at most two connectors. Thus, P contains at most four connectors and all subpaths of P that do not contain ℓ as an inner vertex are unique shortest paths.*

Proof Obviously, ℓ is close to itself. Consider the set of vertices that are close to ℓ such that each one of them has a clockwise shortest path to ℓ on C. Let u be a vertex of maximum distance from ℓ among these vertices. By the property that a subpath of a shortest path is a shortest path, we conclude that this set of vertices is a subpath of C, and this is the clockwise path from u to ℓ. Similarly, consider the set of vertices that are close to ℓ such that each one of them has a counterclockwise shortest path to ℓ, and let v be such a vertex of maximum distance on the cycle from ℓ. We get another subpath of C from ℓ to v. Let P denote the subpath of C that results from concatenating these two subpaths. The subpath of P starting at ℓ and ending at v contains at most two connectors, as otherwise a path from ℓ to v, where the subpath between the first and last connectors (containing at least two edges), is replaced by a two-edge path through the hub is a shortest path, and hence v is not close to ℓ. The proof for the subpath from u to ℓ is similar. Thus, the total number of connectors in P cannot exceed four, and P is a proper subpath of C. Assume by contradiction that a subpath of the subpath of P from ℓ to v is not unique (the proof for the subpath of P from u to ℓ is similar). If there is another shortest path through the hub, then this implies another shortest path from ℓ to v through the hub, and contradicts the property that v is close to ℓ. If there is another shortest path on the cycle, then this shortest path has more than four connectors (as the path from ℓ to v has at most two connectors), in which case it can be replaced with a path through the hub using the first and last connector, and this path is shorter, which is a contradiction. □

Lemma 6.8 *L is a feasible landmark set if and only if the following conditions hold:*

1. *For every layer V_j, there is at most one vertex of V_j that is not close to L.*
2. *For every $\ell \in L$ and every j, if $u_1, u_2 \in V_j$ are close to ℓ and $d_{\ell,u_1} = d_{\ell,u_2}$, then there is $\ell' \in L \setminus \{\ell\}$ that is close to at least one of the vertices u_1 or u_2.*

Proof First assume that L is a feasible landmark set. By Lemma 6.6, the first condition holds. If the second condition does not hold, then let $u_1, u_2 \in V_j$ be two vertices that are close to ℓ, such that $d_{\ell,u_1} = d_{\ell,u_2}$ holds, but neither u_1 nor u_2 is close to another vertex in L. We find that u_1 and u_2 are not separated by ℓ, and are not separated by $\ell' \in L \setminus \{\ell\}$ because for $i = 1, 2$ we have $d_{u_i,\ell'} = j + d_{n,\ell'}$, and this contradicts the feasibility of L.

Next, assume that L satisfies the two conditions. Let $u_1, u_2 \in V$, we need to show that u_1 and u_2 are separated by L. We next prove that there is $\ell' \in L$ such that none of u_1 and u_2 is close to ℓ'. Consider the path P in C connecting u_1 and u_2 that contains at least 10 connectors as internal vertices (such a path exists since G has at least 22 connectors). Assume without loss of generality that this is a clockwise path from u_1 to u_2. Consider the pair of middle connectors v_1, v_2 along P (these are two connectors such that each of the subpaths of P from u_1 to v_1 and from v_2 to u_2 contains at least four connectors as internal vertices). By the first condition, since $v_1, v_2 \in V_1$, at least one of v_1 and v_2 is close to some landmark $\ell' \in L$. By Lemma 6.7, the set of vertices that are close to ℓ' is a proper subpath of C that has at most four connectors. Since this subpath contains either v_1 or v_2, we conclude that none of u_1 and u_2 is close to ℓ'. Therefore, if u_1 and u_2 belong to distinct layers, then there is $\ell' \in L$ (which is not close to any of them) that separates u_1 and u_2, since $d_{\ell',u_i} = d_{\ell',n} + d_{n,u_i}$ for $i = 1, 2$. In what follows we assume that $u_1, u_2 \in V_j$.

Next, if there is a landmark $\ell \in L$ that is close to u_1 but not close to u_2, then $d_{u_1,\ell} < j + d_{n,\ell}$ and $d_{u_2,\ell} = j + d_{n,\ell}$, and therefore ℓ separates u_1 from u_2. Thus, in the remaining case, every $\ell \in L$ that is close to either u_1 or u_2 must be close to both of these vertices. There is at least one landmark $\ell \in L$ close to at least one of the two vertices by the first condition, and it is close to both of them by the assumption. If $d_{u_1,\ell} \neq d_{u_2,\ell}$, then we are done. Thus, assume that $d_{u_1,\ell} = d_{u_2,\ell}$. Since u_1 and u_2 are close to ℓ, we conclude that the (unique) shortest path from u_1 to u_2 in C traverses ℓ which is exactly the middle vertex along this path. Thus, a vertex ℓ which satisfies all these conditions is unique (for a given pair of vertices u_1 and u_2). By the second condition, we conclude that in this case there is another landmark $\ell' \in L$ such that $\ell' \neq \ell$ and it is close to either u_1 or u_2 or both, but according to the assumption, it is close to both, and this is a contradiction to the uniqueness of ℓ as established above. □

Corollary 6.9 *A minimal landmark set L of a wheel with at least 22 connectors does not contain the hub, and $|L| \geq 6$.*

Proof The first part holds because if L satisfies the two conditions of Lemma 6.8, then so does $L \setminus \{n\}$. As for the second part, assume by contradiction that $|L| \leq 5$. G has at least 22 connectors, and by Lemma 6.8, at most one of them is not close to L. Thus, there is a landmark $\ell \in L$ such that at least five connectors are close to ℓ contradicting Lemma 6.7. □

Let L^* be a fixed optimal solution. We say that two landmarks ℓ_1 and ℓ_2 are consecutive if the clockwise path from ℓ_1 to ℓ_2 does not contain an additional landmark. Such a path is called the *natural path between the landmarks*. Given a landmark $\ell \in L^*$, we say that a pair of cycle-vertices x, y is a bad pair with respect to ℓ (or a bad pair of ℓ), if x, y are close to ℓ, belong to a common layer, $d_{x,\ell} = d_{y,\ell}$, and the clockwise path from x to y traverses ℓ. Recall that in this case there must be a landmark $\ell' \neq \ell$ that is close to at least one of x and y, and we say that ℓ' covers the bad pair x, y of ℓ.

Lemma 6.10 *Let x, y be a bad pair with respect to a landmark ℓ. Let $\ell' \neq \ell$ be a landmark that is close to one of x and y (or both). Let ℓ_1 and ℓ_2 be landmarks such that ℓ_1 and ℓ are consecutive landmarks and ℓ and ℓ_2 are consecutive landmarks. Then the landmark ℓ_1 is close to x, or the landmark ℓ_2 is close to y (or both). Thus, every bad pair of ℓ is covered by either ℓ_1 or ℓ_2 (or by both of them).*

Proof First consider the case where the clockwise path (the shortest path) from x to ℓ traverses ℓ_1 (perhaps $x = \ell_1$). Then, the subpath of this path from x to ℓ_1 is the unique shortest path between x and ℓ_1, so x is close to ℓ_1 and we are done. The case where the counterclockwise path from y to ℓ traverses ℓ_2 is symmetric. Without loss of generality, assume that x is close to ℓ' and the clockwise path from ℓ_1 to ℓ_2 traverses x, ℓ, y (in this order) but does not traverse ℓ' (the last assumption holds since there are no landmarks except for ℓ that are internal vertices of this path). We split this case into two subcases. In the first subcase the shortest path P from ℓ' to x is clockwise, and in the second one this path is counterclockwise. In the first subcase, the subpath of P from ℓ_1 to x is a unique shortest path, and in the second subcase the clockwise path from y to ℓ_2 is a subpath of P, and thus it is a unique shortest path. □

A *minimal bad pair of a landmark* ℓ is a bad pair x, y such that $d_{x,\ell}$ is minimal among all bad pairs of ℓ.

Lemma 6.11 *Let $\ell' \neq \ell$ be a landmark that covers a minimal bad pair x, y of ℓ such that either ℓ', ℓ are consecutive or ℓ, ℓ' are consecutive, then ℓ' covers every bad pair of ℓ.*

By Lemma 6.10, it is indeed possible to assume without loss of generality that ℓ' satisfies the condition of the lemma.

Proof Let x', y' be a bad pair of ℓ. Thus, the clockwise path from x' to y' traverses x, ℓ, y in this order. Without loss of generality, assume that ℓ', ℓ are consecutive and x is close to ℓ'. If the clockwise path from ℓ' to ℓ traverses x', then this path contains the unique shortest path from ℓ' to x that contains as a subpath the unique shortest path from ℓ' to x', and we are done. Otherwise, the clockwise path from x' to ℓ (which is the unique shortest path between them) traverses ℓ', and therefore the clockwise path from x' to ℓ' is a unique shortest path, and the claim holds. □

We say that a minimal bad pair x, y of ℓ that is covered by ℓ' is *covered from the left by* ℓ' if ℓ' and ℓ are consecutive, and otherwise if ℓ and ℓ' are consecutive, then we say that x, y are covered from the right by ℓ'. By Lemma 6.10, we can assume that a bad pair which is covered is either covered from the right by some ℓ' or covered from the left by some ℓ' (or both). By Lemma 6.11, we find that it is sufficient to verify that minimal bad pairs are covered (rather than all bad pairs).

Corollary 6.12 *Let $L \subseteq V$ be a minimal feasible landmark set. The set L satisfies the following conditions.*

1. *$n \notin L$.*
2. *The set of vertices that are not close to L forms a shortest path from n to some vertex v (possibly $v = n$), in particular, if $v \neq n$, then the path contains exactly one connector and no landmark.*

3. *For every $\ell \in L$, there is $\ell' \in L \setminus \{\ell\}$ that covers the minimal bad pair of ℓ either from the left or from the right (if there exists a bad pair of ℓ).*

Moreover, if a set $L \subseteq V$ satisfies these three conditions, then L is a feasible landmark set.

If there is a cycle-vertex that is not close to L^*, then all such cycle-vertices form a path without a landmark. In order to construct an optimal landmark set, we guess a pair of consecutive landmarks in L^* such that if there exists a cycle-vertex not close to L^*, then all such vertices appear along the natural path between the two guessed landmarks. Without loss of generality and for the sake of presentation, we assume that the two guessed landmarks are 1 and $k \leq n-1$, and the natural path between them is the clockwise path from k to 1. Once again, we will define a dynamic program along the path from 1 to k that positions landmarks along this path. The number of possibilities for the selection of these two landmarks is $O(n^2)$. We first verify that the set of vertices along the natural path from k to 1 that are not close to 1 or to k (if such a cycle-vertex exists) form a shortest path from some vertex to n (and in particular, if it is non-empty, then it contains exactly one connector, since all connectors are in V_1). If this condition does not hold, then this possibility (the choice of two landmarks) is impossible, and we stop considering it. In what follows, we consider a possibility which passed this test. Since the number of connectors along the natural path from k to 1 is at most five, the clockwise path from 1 to k contains at least 17 connectors, and it is not a shortest path.

Let $1 < v < k$. We define $low(v)$ as the minimum index i such that $1 \leq i \leq v$ and the clockwise path from i to v is a unique shortest path. We let $high(v)$ be the maximum index i such that $v \leq i \leq k$ and the clockwise path from v to i is the unique shortest path. $\delta(v)$ is the minimum number $i \geq 1$ such that $v - i \geq low(v)$ and $v + i \leq high(v)$ belong to a common layer. If such i does not exist, then we let $\delta(v) = \infty$. The motivation for this definition of $\delta(v)$ is to identify the minimal bad pair of v (if it exists).

Claim 6.13 *Let $1 < v < k$ be a landmark such that ℓ_1, v are consecutive landmarks and v, ℓ_2 are consecutive landmarks ($1 \leq \ell_1 < v < \ell_2 \leq k$ since $1, k$ are landmarks), assume that v has a bad pair, and let x, y be the minimal bad pair of v. If $\{x, y\}$ is not contained in the clockwise path from 1 to k, then either 1 or k covers x, y. Otherwise, x, y is covered if and only if at least one of the following conditions holds: (i) $x \leq high(\ell_1)$, (ii) $y \geq low(\ell_2)$.*

Proof First consider the case where x is not contained in the clockwise path from 1 to k. As x is close to v, the unique shortest path from x to v traverses 1. Thus, this is the unique shortest path from x to 1, and 1 covers the bad pair x, y. The case where y is not contained in the clockwise path from 1 to k is similar.

Consider the second claim. We have $x = v - \delta(v)$ and $y = v + \delta(v)$. If $\ell_1 \leq x$, then x is close to ℓ_1 if and only if $\ell_1 \leq x \leq high(\ell_1)$ and the claim holds. Otherwise, if $\ell_1 > x$, then the unique shortest path from x to v traverses ℓ_1, and therefore x is close to ℓ_1 as well, and in this case $x \leq \ell_1 \leq high(\ell_1)$. The claim that $y \geq low(\ell_2)$ if and only if y is close to ℓ_2 is proved similarly. □

For the vertex 1, we define the parameters $Low(1)$, $High(1)$, $\Delta(1)$ as follows. $Low(1)$ is the minimum index $1 < i \leq n-1$ such that the clockwise path from i to 1 is the unique shortest path (by definition $n-1$ satisfies this condition, hence $Low(1)$ is well-defined, and we find that the vertices $i, i+1, \ldots, n-1$ are close to 1). $High(1)$ is the maximum index $i > 1$ such that the clockwise path from 1 to i is the unique shortest path (similarly to the existence of $Low(1)$, the value of $High(1)$ is well-defined, vertices $2, 3, \ldots, High(1)$ are close to 1, and $High(1) < k$). $\Delta(1)$ is the minimum number $i \geq 1$ such that $i+1$ and $n-i \geq k+1$ are both close to 1 (and have the same distance from 1) and belong to a common layer. If such a value i does not exist, then we let $\Delta(1) = \infty$. This definition is related to bad pairs and the goal is to define a minimal bad pair. We only consider bad pairs x, y of a specific form, that x appears on the clockwise path between k and 1. It can be the case that there exists a bad pair of 1 even if $\Delta(1) = \infty$, but in this case we show later that k covers all such bad pairs.

For the vertex k, we define similar parameters $Low(k)$, $High(k)$, $\Delta(k)$ as follows. $Low(k)$ is the minimum index $i > 1$ such that the clockwise path from i to k is the unique shortest path (by definition $k-1$ satisfies this condition, hence $Low(k)$ is well-defined). $High(k)$ is the maximum index $i \leq n-1$ such that the clockwise path from k to i is the unique shortest path (note that it is possible that vertex 1 is also close to k, using the clockwise path from k to 1, but we are not interested whether this holds or not). $\Delta(k)$ is the minimum number $i \geq 1$ such that $k+i \leq n-1$ and $k-i \geq 1$ are both close to k (and have the same distance from k) and belong to a common layer. If such i does not exist, we let $\Delta(k) = \infty$. In this case we are only interested in bad pairs x, y where y appears on the clockwise path between k and 1 (and other bad pairs are covered by 1, as we show later).

We also let $low(1) = 1, high(1) = High(1), low(k) = Low(k)$ and $high(k) = k$. We define $\delta(1)$ and $\delta(k)$ in the following way. We let $\delta(1) = \infty$ if $High(k) \geq n - \Delta(1)$, and otherwise, $\delta(1) = \Delta(1)$. The motivation is to define $\delta(1)$ to be infinite if there is no bad pair of 1, or if k covers the minimal bad pair of 1 (this can happen if there is a bad pair of 1 but $\Delta(1) = \infty$, or if $\Delta(1) < \infty$ and $\delta(1) = \infty$). Similarly, we define $\delta(k) = \infty$ if $Low(1) \leq k + \Delta(k)$. Otherwise, we let $\delta(k) = \Delta(k)$. The next lemma establishes the correctness of the differences between Δ and δ and the property that $\Delta(1)$, $\Delta(k)$ can be infinite even if there is a bad pair of 1, and k, respectively.

Lemma 6.14 *$High(k) \geq n - \Delta(1)$ if and only if the minimal bad pair of 1 is covered by k. Similarly, $Low(1) \leq k + \Delta(k)$ if and only if the minimal bad pair of k is covered by 1.*

Proof If there is no bad pair of 1, then $\Delta(1) = \infty$ and the claim holds. Otherwise let x, y be the minimal bad pair of 1. If $\Delta(1) = \infty$, then the inequality holds and the clockwise path from x to 1 traverses k (possibly $x = k$), and k covers this bad pair since the unique shortest path from x to k is contained in the clockwise path from x to 1. In the remaining case, the clockwise path from k to 1 traverses x, and then $x = n - \Delta(1)$. Then, x is close to k if and only if the inequality holds. Thus, the first claim holds. The proof of the second claim is analogous. □

We define a function $F : \{1, 2, \ldots, k\} \times \{left, right\} \rightarrow \mathbb{R}^+$ as follows. $F(v, right)$ ($F(v, left)$) is the minimum cost of a set L such that $1, v \in L$, every

$1 < i < v$ is close to at least one of the vertices in L, and for every $\ell \in L \setminus \{v\}$ ($\ell \in L$, respectively) such that $\delta(\ell)$ is finite, the minimal bad pair of ℓ is covered by a vertex in $L \setminus \{\ell\}$.

We compute the values of F using the following dynamic program. If $\delta(1) = \infty$, then we let $F(1, right) = F(1, left) = c(1)$, and if $\delta(1)$ is finite, then $F(1, left) = \infty$ and $F(1, right) = c(1)$. For $\ell > 1$, we define sets of feasible values of $\ell' < \ell$ (with the goal that ℓ',ℓ will be consecutive landmarks) $\alpha(\ell) = \{\ell' \in \{1, 2, \ldots, \ell-1\} : high(\ell') \geq low(\ell) - 1\}$, $\beta(\ell) = \{\ell' \in \alpha(\ell) : high(\ell') \geq \ell - \delta(\ell)\}$, $\gamma(\ell) = \{\ell' \in \alpha(\ell) : low(\ell) \leq \ell' + \delta(\ell')\}$. The recursive formula is as follows.

$$F(\ell, left) = \min\left\{\min_{\ell' \in \beta(\ell)} F(\ell', left), \min_{\ell' \in \beta(\ell) \cap \gamma(\ell)} F(\ell', right)\right\} + c(\ell),$$

$$F(\ell, right) = \min\left\{\min_{\ell' \in \alpha(\ell)} F(\ell', left), \min_{\ell' \in \gamma(\ell)} F(\ell', right)\right\} + c(\ell).$$

If $\delta(k) = \infty$, then we are looking for $F(k, right)$, and otherwise we are looking for $F(k, left)$. Then, by backtracking we compute the optimal landmark set. We conclude that the algorithm computes an optimal solution in polynomial time. Thus, we established the following.

Theorem 6.15 *Given a wheel $G = (V, E)$,* `wmd` *can be solved in polynomial time. In particular, if G is a wheel with at least 22 connectors, then the running time is $O(n^4)$.*

Proof First, we construct two matrices of sizes $n \times n$, the first one containing all the distances (this can be computed in $O(n^3)$), and the second one indicating for every pair of cycle-vertices whether they are close (each entry is computed in $O(1)$ using the other matrix). The second matrix later allows to check in time $O(1)$ whether a pair of vertices are close. The first matrix will be used to find levels of vertices. In what follows, we call the two chosen vertices (denoted by 1 and k in the algorithm) special vertices.

For every pair of cycle-vertices the following process is done. All the attributes defined above are computed ($\delta(v)$, $high(v)$, $low(v)$, $\alpha(v)$, $\beta(v)$, and $\gamma(v)$) for all vertices on the cycle path between the two special vertices. Note that the values of the attributes depend on the choice of the two special vertices. The additional attributes of the two special vertices are computed as well. This can be done using $O(n)$ time per vertex. Then, the function F is computed using linear time per vertex. We find a total time of $O(n^2)$ for each pair of special vertices, and thus the total running time is $O(n^4)$. □

7 Hard Cases

In this section we extend the classes of graphs for which computing $\mathtt{md}(G)$ is known to be NP-hard. Here the graphs are unweighted, and there is no cost function. We define a variant of 3- Dimensional Matching (3DM) which we call Cover 3-Dimensional Matching (C3DM). We first consider this variant and show that it is NPC. Then, we define a gap creating procedure that we use in our reductions.

Afterwards, we prove that computing $\mathtt{md}(G)$ is NPC for the following classes of graphs: Split graphs, bipartite graphs, co-bipartite graphs, and line graphs of bipartite graphs.

7.1 Preliminaries

Recall that the 3DM problem is defined as follows. Given three disjoint sets each consisting of n elements, $A = \{a_1, a_2, \ldots, a_n\}$, $B = \{b_1, b_2, \ldots, b_n\}$ and $C = \{c_1, c_2, \ldots, c_n\}$, and a set of triples $S \subseteq A \times B \times C$ (where a triple is an element of $A \times B \times C$), is there a subset S' of S (consisting of exactly n triples) such that each element of $A \cup B \cup C$ occurs in exactly one of the triples of S'. It is well-known that 3DM is NPC.

Here we consider the following variant which we call Cover 3- Dimensional Matching (C3DM). Given three disjoint sets each consisting of n elements, $A = \{a_1, a_2, \ldots, a_n\}$, $B = \{b_1, b_2, \ldots, b_n\}$ and $C = \{c_1, c_2, \ldots, c_n\}$, and a set of triples $S \subseteq A \times B \times C$, for every subset S' of S we define its cost to be the sum of two values: the first of which is $|S'|$, and the second one is the number of elements that do not belong to any of the triples in S'. The goal is to find out whether there exists a subset S' of S which has cost of at most n. Clearly a subset S' of S has a cost at most n if and only if it is a feasible 3-dimensional matching. Therefore, C3DM is NP-hard. C3DM is in NP, and therefore it is in fact NPC as well.

In our reductions we need a gap between the cost of optimal solutions for the following two cases: the case that a 3-dimensional matching exists, and the case that such a 3-dimensional matching does not exist. For this, we define a subroutine CREATE-GAP that receives an input of C3DM with the parameter n and creates (in polynomial time) another instance of C3DM with the parameter $n' = 2^{12}(n)^2$, that satisfies the following properties. The first property is $n' \geq 2^{12}$ (the reason for this large value will become clear in the next section). Additionally, if the original instance has a solution of cost at most n, then for the new instance there is a solution of cost at most n'. However, if for the original instance the cost of an optimal solution is at least $n + 1$, then the cost of any solution for the new instance is at least $n' + \sqrt{n'}$. Thus, for the new instance, the question whether there is a solution of cost at most n' is equivalent to the question whether there is a solution of cost at most $n' + \sqrt{n'} - 1$.

Lemma 7.1 *There exists a polynomial time subroutine CREATE-GAP as described.*

Proof The subroutine CREATE-GAP is defined as follows. Consider an instance to C3DM with the parameter n and $3n$ elements, and construct "an $N = 2^{12}n$ copies instance of C3DM" by repeating each element N times. Thus, instead of the sets A, B, C we have N sets A_i, B_i, C_i (for $i = 1, 2, \ldots, N$), where $A_i = \{a^i_1, a^i_2, \ldots, a^i_n\}$, $B_i = \{b^i_1, b^i_2, \ldots, b^i_n\}$ and $C_i = \{c^i_1, c^i_2, \ldots, c^i_n\}$,and each triple has N copies, i.e., if $(a, b, c) \in S$, then for $i = 1, 2, \ldots, N$ we have (a^i, b^i, c^i) in the collection of triples (that is, letting $n' = Nn$, the number of elements in the new instance is $3n' = 3Nn$). We claim that the question whether the new instance has a sub-collection $S' \subseteq S$ of cost at most $n' + \sqrt{n'} - 1$ is equivalent to the question whether it has a solution of cost at most n'. This last claim holds because every such sub-collection S' induces N sub-collections S'_i (for $i = 1, 2, \ldots, N$) where S'_i is the set of triples whose elements are all in $A_i \cup B_i \cup C_i$. The cost of S' is similarly partitioned into the N copies of the original instance. Therefore, if there is a sub-collection S' of cost at most $n' + \sqrt{n'} - 1$ then at least one of the S'_i's has cost at most $\lfloor \frac{n' + \sqrt{n'} - 1}{N} \rfloor = n$. To prove the other direction we assume that it is possible to get a sub-collection of cost at most n to the original instance. In this case we can take N copies of this solution (one for every i), and obtain a solution of cost $nN = n'$ to the new instance. □

7.2 Split Graphs, Bipartite Graphs, and Co-bipartite Graphs

A split graph $G = (V, E)$ is a graph such that the vertex set V can be partitioned into a clique set Cl and an independent set Is such that every pair of vertices in Cl is adjacent and every pair of vertices in Is is not adjacent.

Theorem 7.2 *1. Given a split graph $G = (V, E)$ and a value K, deciding whether $\mathtt{md}(G) \leq K$ is NPC.*

2. Given a bipartite graph $G = (V, E)$ and a value K, deciding whether $\mathtt{md}(G) \leq K$ is NPC.

3. Given a co-bipartite graph $G = (V, E)$ and a value K, deciding whether $\mathtt{md}(G) \leq K$ is NPC.

Proof The problems are in NP, as the problem for general graphs is in NP. To see that the problems are NPC we will use three reductions from C3DM. In each case, with a slight abuse of notation, we assume that the subroutine CREATE-GAP was already applied to the considered instance of C3DM. See Fig. 6 for an illustration of the construction of bipartite graphs.

Given an instance of C3DM with $3n$ elements partitioned into the three ground sets $A = \{a_1, a_2, \ldots, a_n\}$, $B = \{b_1, b_2, \ldots, b_n\}$ and $C = \{c_1, c_2, \ldots, c_n\}$ and a collection S of τ triples, each containing one element of A, one element of B and one element of C (and so $\tau \leq n^3$). Let $\nu = \lceil \log_2 \tau \rceil$ and $\gamma = \nu + 8$. Note that $\gamma < \sqrt{n} - 1$ since $n \geq 2^{12}$ (the function $\sqrt{n} - 3\log_2 n - 10 \leq \sqrt{n} - 1 - (\lceil \log_2 \tau \rceil + 8)$ is monotonically increasing, and it is equal to $64 - 36 - 10 > 0$ for $n = 2^{12}$). Let $K = n + \gamma - 3$.

We describe theconstruction of a graph G whose vertices are partitioned into two sets I and J. In the set J we will have a vertex for each element of $A \cup B \cup C$ (we will use the element and its corresponding vertex interchangeably), and an additional νvertices $d_0, d_1, \ldots, d_{\nu-1}$. In the set I we have one vertex for each triple $s \in S$. We further assume that these vertices have indices $s_0, s_1, \ldots, s_{\tau-1}$ (we will use the triple and its corresponding vertex interchangeably). Moreover, I has four additional vertices s_A, s_B, s_C, s_D. We next describe the edge set of the graph connecting a vertex of I and a vertex of J. If $u \in J$ and $v \in I$, then $\{u, v\} \in E$ in the seven following cases:

1. $u \in A$ and $v = s_A$.
2. $u \in B$ and $v = s_B$.
3. $u \in C$ and $v = s_C$.
4. $u \in A \cup B \cup C$ and $v = s_D$.

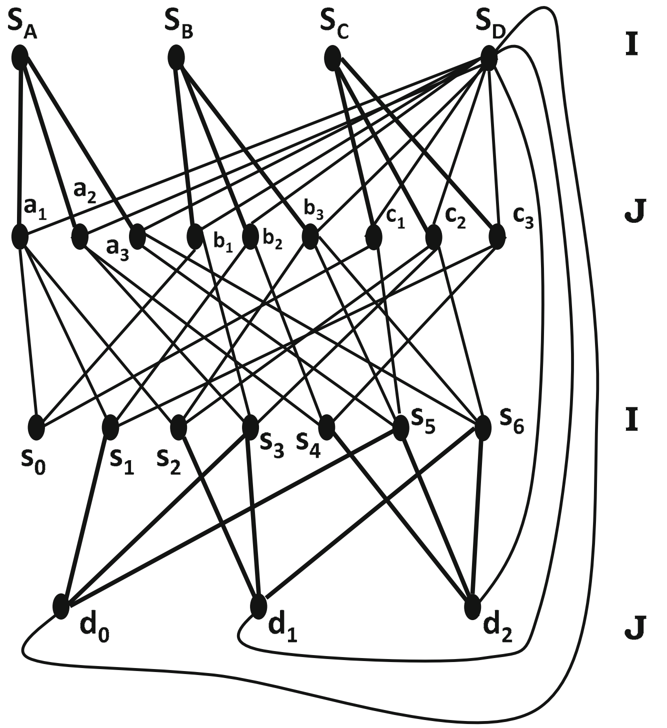

Fig. 6 An example for constructing the input for $\mathtt{md}(G)$ for the case of bipartite graphs. The instance of C3DM is defined as follows (the instances actually used for the reduction are much larger and the resulting graphs are much larger as well). Let $n = 3$, $\tau = 7$, and $\nu = 3$. The triples are $\{(a_1, b_1, c_1), (a_1, b_2, c_3), (a_1, b_3, c_2), (a_2, b_1, c_2), (a_2, b_2, c_3), (a_3, b_3, c_1), (a_3, b_3, c_2)\}$

5. $u \in A \cup B \cup C$ and $v \in \{s_0, s_1, \ldots, s_{\tau-1}\}$ and the triple corresponding to v contains the element u.
6. $u = d_i$ (for some value of i) and $v = s_j$ for some $j = 0, 1, \ldots, \tau - 1$, such that the i-th (least significant) bit of the binary representation of the index j is 1.
7. $u = d_i$ (for some value of i) and $v = s_D$.

The set of additional edges of G (between pairs of vertices of I and between pairs of vertices of J) is defined according to the following cases. For the case of bipartite graphs there are no additional edges. For the case of split graphs, J is a clique and I is an independent set, and for the case of co-bipartite graphs both I and J are cliques. Clearly, the construction of the graph G in all cases can be done in polynomial time.

If there exists a solution to C3DM of cost at most n, then there exists a set of n indices $i_1, \ldots, i_n$ such that the collection of triples $\{s_{i_1}, \ldots, s_{i_n}\}$ is a 3-dimensional matching. Let $L = \{s_{i_1}, \ldots, s_{i_n}\} \cup \{s_A, s_B, s_C, s_D\} \cup \{d_0, d_1, \ldots, d_{\nu-1}\}$. We have $|L| < K$ and thus it suffices to show that L is a feasible landmark set. Let $x, y \in V$. It suffices to prove that if $x, y \notin L$ then there is a landmark in L that separates x from y. Therefore, assume that $x, y \in A \cup B \cup C \cup S$. Consider the case $x, y \in A \cup B \cup C$. The vertex corresponding to a triple that contains x separates x from y, unless they both belong to this triple, but in this case they belong to different ground sets. Since x and y belong to different ground sets, x is adjacent to exactly one of the vertices s_A, s_B, s_C and y is adjacent only to another vertex in $\{s_A, s_B, s_C\}$. Next consider the case in which $x, y \in S$. Let j be an index of a bit in which the index of x differs from the index of y. Then d_j is adjacent to exactly one of them, and thus separates x from y. In the last case, without loss of generality, $x \in A \cup B \cup C$ whereas $y \in S$. In this case, x is adjacent to exactly one of the vertices s_A, s_B, s_C, and y is either adjacent to all of them (in the case of co-bipartite graphs) or none of them (in split graphs and bipartite graphs), and therefore there exists a landmark that separates x from y. We conclude that in this case $\mathtt{md}(G) \le K$.

To prove the other direction, assume that G has a landmark set L of cardinality at most K. We define $V' \subseteq V$ in the following way: $V' = A \cup B \cup C \cup S$, and $L' = V' \cap L$. We say that an element $x \in A \cup B \cup C$ is covered if either $x \in L'$ or x is contained in a triple $s_i \in L'$, and otherwise x is uncovered. We claim that each of$A$, B and C has at most one uncovered element. To see this fact, assume by contradiction that it does not hold. Without loss of generality, A has two uncovered elements x_1, x_2.

For the case of split graphs and co-bipartite graphs, the distance from any vertex of $L \cap J$ to either x_1 or x_2 is 1, and therefore such a vertex does not separate x_1 from x_2 (if it is a landmark). For every $\ell \in L' \cap I$ and $i = 1, 2$, we have $d_{x_i,\ell} = 2$ because ℓ is not adjacent to x_i (as it does not contain x_i) and there is a path of length 2 between these two vertices using an intermediate vertex which is the vertex corresponding to the element of ℓ which belongs to A. Therefore, such a vertex does not separate x_1 from x_2. Finally $(L \cap I) \setminus L' \subseteq \{s_A, s_B, s_C, s_D\}$ and none of s_A, s_B, s_C, s_D separates x_1 from x_2, as their distances to s_A and s_D are 1, and their distances to s_B and s_C are 2, through some vertex of B or C, respectively.

For the case of bipartite graphs, the distance from any vertex $w \in L \cap J$ to either x_1 or x_2 is 2 since there is a path via s_D (and no direct edge between w and x_1 or x_2). For every $\ell \in L' \cap I$ and $i = 1, 2$, we have $d_{x_i,\ell} = 3$ because ℓ or its neighbors are not adjacent to x_i (as ℓ does not contain x_i) and there is a path of length 3 between these two vertices which traverses an element of ℓ and s_D (a path of length 2 cannot exist as the graph is bipartite with the partitions I and J). Therefore, such a vertex does not separate x_1 from x_2. Finally $(L \cap I) \setminus L' \subseteq \{s_A, s_B, s_C, s_D\}$ and none of s_A, s_B, s_C, s_D separates x_1 from x_2, as their distances from s_A and s_D are 1, and their distances from s_B and s_C are 3 (once again, the distance cannot be 2), and there must exist paths of length 3 since every vertex of A has a path of length 2 to all vertices of $B \cup C$, as every element of $A \cup B \cup C$ has s_D as a neighbor.

Therefore, A has at most one uncovered element, and similarly each one of the sets B and C has at most one uncovered element. Therefore, the collection of triples consisting of $L' \setminus (A \cup B \cup C)$ costs at most $|L'| + 3 \le K + 3 = n + \gamma < n + \sqrt{n} - 1$ when it is considered as a solution to C3DM, and thus (since the input for C3DM is the one resulting from applying the procedure CREATE-GAP), there is a solution of cost at most n for the C3DM instance. $\square$

7.3 Line Graphs of Bipartite Graphs

Here, we consider the problem of computing $\mathtt{wmd}(G')$ where G' is a line graph of a bipartite graph G. Instead of describing our reduction in terms of G', we will discuss it with respect to G. To do so, we introduce the problem of computing the EDGE METRIC DIMENSION OF A GRAPH G as follows. The input consists of a graph $G = (V, E)$. An edge $\ell \in E$ is called a *separating edge landmark* for two edges $e, e' \in E$ with $e \neq e'$, if the length of the shortest path whose first edge is e and its last edge is ℓ (this is the shortest path from e to ℓ) differs from the length of the shortest path from e' to ℓ. A distance between a pair of edges is the distance between the two corresponding vertices in the line graph G', that is, the number of edges in the shortest path that starts at the first edge and ends at the second edge minus 1. A subset $L \subseteq E$ is an *edge landmark set* for the graph G, if for any two edges $e, e' \in E$ with $e \neq e'$ there exists a separating edge landmark $\ell \in L$ that separates e from e'. The goal of our problem is to select a feasible edge landmark set of a minimum cardinality. Note that the goal of this problem is equivalent to computing $\mathtt{md}(G')$ where G' is the line graph of G. Thus, in order to show that computing the metric dimension of line graphs of bipartite graphs is NP-hard, we will prove that computing the edge metric dimension of a bipartite graph G is NP-hard.

Lemma 7.3 *Given a bipartite graph G and a value K, deciding whether there is a feasible edge landmark set of cardinality at most K is NPC.*

Proof The problem is in NP, as given an edgelandmark set, it is easy to verify in polynomial time that it is indeed feasible and that its cardinality is at most K. To see that the problem is NPC we will use a reduction from C3DM. Once again, we assume that the subroutine CREATE-GAP was already applied to the considered instance of C3DM. See Fig. 7 for an illustration of the construction of the reduction.

Given an instance of C3DM with $3n$ elements $A = \{a_1, a_2, \ldots, a_n\}$, $B = \{b_1, b_2, \ldots, b_n\}$ and $C = \{c_1, c_2, \ldots, c_n\}$ and a collection S of τ triples, each containing one element of A, one element of B and one element of C (note that $\tau \leq n^3$), we let $\nu = \lceil \log_2 \tau \rceil$ and $\gamma = \nu + 8$, and construct a bipartite graph as a layer graph with 5 layers V_1, V_2, V_3, V_4, V_5 as follows.

V_1 and V_2 have a vertex for each triple,V_3 has one vertex for each element of $A \cup B \cup C$ (we will use the element to denote this vertex and vice versa), and an additional $\nu = \lceil \log_2 \tau \rceil$vertices $d_0, d_1, \ldots, d_{\nu-1}$. V_4 has $\nu + 4$ vertices, where $V_4 = \{v_A, v_B, v_C, v_D, d'_0, d'_1, \ldots, d'_{\nu-1}\}$. Finally, V_5 has four vertices and we let$V_5 = \{u_A, u_B, u_C, u_D\}$. We next describe the edge set of G. Each triple $(a, b, c) \in S$ has unique index in $0, 1, \ldots, \tau - 1$. For $(a, b, c) \in S$ of index j, the two vertices in V_1 and V_2 corresponding to this triple are denoted by s_j and s'_j, respectively, and we have the following edges $\{s_j, s'_j\}$, $\{s'_j, a\}$, $\{s'_j, b\}$, $\{s'_j, c\}$. Moreover, if the index j of the triple (a, b, c) has 1 in its i-th bit then we add an edge $\{s'_j, d_i\}$. In addition, for each vertex a in $V_3 \cap A$ (corresponding to an element $a \in A$), we have an edge $\{a, v_A\}$. Similarly, for each $b \in B \cap V_3$, there is an edge $\{b, v_B\}$, and for each $c \in C \cap V_3$, there is an edge $\{c, v_C\}$. Every vertex of V_3 is adjacent to v_D. For each $i = 0, 1, \ldots, \nu - 1$ we have an edge $\{d_i, d'_i\}$, and the last four edges of G are

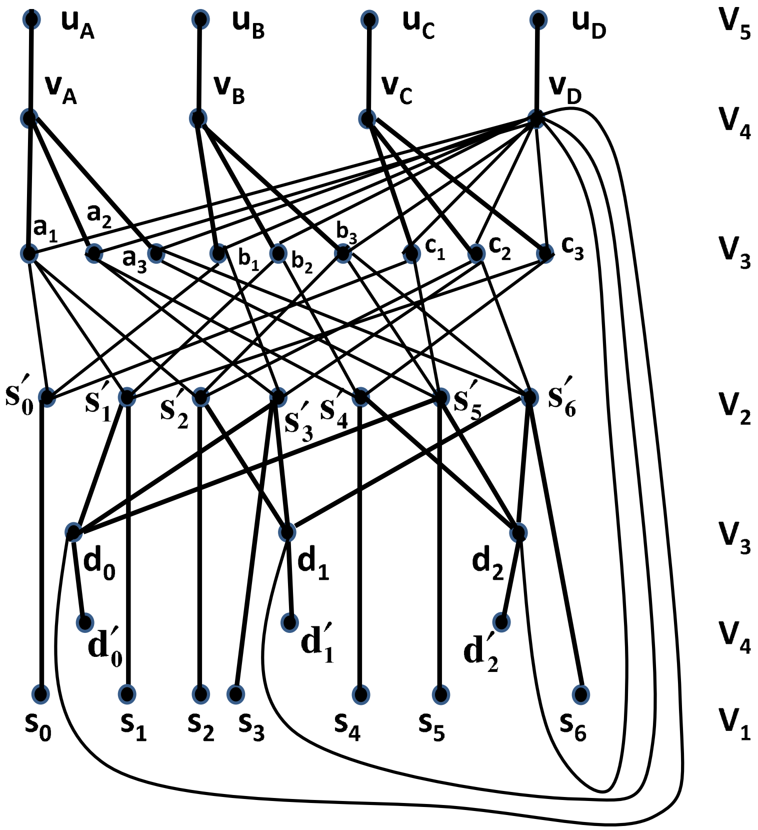

Fig. 7 An example for constructing the input in Lemma 7.3. The instance of C3DM is defined as in Fig. 6

$\{v_A, u_A\}, \{v_B, u_B\}, \{v_C, u_C\}, \{v_D, u_D\}$. Note that the construction of G can be done in polynomial time. We argue that G has an edge landmark set of cardinality at most $K = n + \nu + 4$ if and only if the optimal solution to C3DM has cost at most $n + \gamma$. This is sufficient since $\gamma < \sqrt{n} - 1$, so a solution of cost at most $n + \gamma$ is equivalent to a solution of cost at most n for C3DM (since the input for C3DM is the one resulting from applying the procedure CREATE-GAP).

First, assume that there is a solution to C3DMof cost at most $n + \gamma$. Then, there is a solution to C3DM of cost at most n, that is, there is a sub-collection S' of n disjoint triples. Our edge landmark set L will consist of the n edges from $V_1 \times V_2$ connecting the pair of vertices s_j to s'_j for the triples of S'. We also have an edge landmark at $\{d_i, d'_i\}$ for $i = 0, 1, \ldots, \nu - 1$ and an edge landmark in the four edges $\{v_A, u_A\}, \{v_B, u_B\}, \{v_C, u_C\}, \{v_D, u_D\}$. $|L| = K$, and it remains to show that this is a feasible edge landmark set. Let e, e' be a pair of edges. We will show that there is $\ell \in L$ that separates e from e'. Note that we need to consider only pairs of edges such that $e, e' \notin L$. Let $\hat{e}, \bar{e} \notin L$. Assume that $\hat{e}$ connects vertices from V_4 and V_5, and $\bar{e}$ connects a vertex of V_{4-i} and a vertex of V_{5-i}. The distance between the two edges is at least i. Note that there is a path consisting of exactly two edges between any vertex in

V_3 to u_D (through v_D), and therefore a path consisting of exactly three edges between any vertex in V_2 to u_D. Moreover, every non-landmark edge connecting a vertex in V_3 and a vertex in V_4 has an edge among $\{v_A, u_A\}, \{v_B, u_B\}, \{v_C, u_C\}, \{v_D, u_D\}$ that is of distance 1 from it. Therefore, by letting j be the minimum distance of a non-landmark edge $\bar{e}$ to one of the landmark edges $\{v_A, u_A\}, \{v_B, u_B\}, \{v_C, u_C\}, \{v_D, u_D\}$, j determines the pair of consecutive layers that $\bar{e}$ connects. Therefore, it suffices to consider the cases in which $e, e' \notin L$ connect pairs of vertices of the same pair of consecutive layers.

Assume that each of e, e' connects a vertex of V_1 with a vertex of V_2. The two triples corresponding to the end-vertices of e and e' have different index. Therefore, there exists a value of i such that the indices of these triples differ in the i-th bit. Then, the edge $\{d_i, d'_i\}$ separates e from e' since it has a path of three edges only to one of e and e'.

Next assume that each of e, e' connects a vertex of V_2 with a vertex of V_3. Letw, w' denote the vertices of e and e' that belong to V_3, respectively. Let $s'_j, s'_{j'}$ denote the other end-vertices of e and e', respectively. First assume that $s'_j \neq s'_{j'}$, and therefore there exists a value of i such that the indices of the corresponding triples differ in the i-th bit. Then, the landmark edge $\{d_i, d'_i\}$ has distance at most 2 from one of e and e' and a larger distance from the other one. Thus, we can assume that $s'_j = s'_{j'}$, and since $e \neq e'$, we have $w \neq w'$. If $w = d_j$, then the landmark edge $\{d_j, d'_j\}$ has distance 1 from e and a distance 2 from e'. Therefore, without loss of generality $w, w' \in A \cup B \cup C$. Therefore, w and w' belong to different sets(among A, B and C), and two of the landmark edges $\{v_A, u_A\}, \{v_B, u_B\}, \{v_C, u_C\}$ separate e from e'.

Finally assume that each of e, e' connects a vertex of V_3 with a vertex of V_4. $e, e' \notin L$ and therefore each of these edges is incident to one of the vertices v_A, v_B, v_C, v_D. If they are incident to different vertices among these four vertices, then two of the following landmark edges $\{v_A, u_A\}, \{v_B, u_B\}, \{v_C, u_C\}, \{v_D, u_D\}$ separate e from e'. Moreover, if either e or e' is incident to one of the vertices d_j (for some j), then the landmark edge $\{d_j, d'_j\}$ separates e from e'. Let w, w' denote the vertices of e and e' which belong to V_3, respectively. Then, $w, w' \in A \cup B \cup C$ and $w \neq w'$. Denote by (a, b, c) and (a', b', c') the triples in S' which contain w and w', respectively. First, assume that $(a, b, c) = (a', b', c')$. Since $w \neq w'$, w and w' belong to different sets among A, B and C. Without loss of generality assume that $w \in A$ and $w' \notin A$. In this case the edge $\{v_A, u_A\}$ separates e from e' (it has distance 1 from e and a larger distancefrom e'). Otherwise, $(a, b, c) \neq (a', b', c')$, and let j denote the index of (a, b, c). Then the distance between the landmark edge $\{s_j, s'_j\}$ to e is 1 and a larger distance from e' (since $\{s'_j, w\} \in E$ and $\{s'_j, w'\} \notin E$). Therefore, L is a feasible edge landmark set and $|L| = K$ as we required.

To prove the other direction, assumethat G has an edge landmark set L of cardinality at most K. We say that L has chosen a triple $(a, b, c) \in S$ whose index is j if (at least) one of the edges $\{s_j, s'_j\}, \{s'_j, a\}, \{s'_j, b\}, \{s'_j, c\}$ belongs to L. We say that an element $x \in A \cup B \cup C$ is covered if either it belongs to a chosen triple, or (at least) one of the edges connecting x to a vertex of V_4 belongs to L, and otherwise x is uncovered. We claim that each of$A$, B and C has at most one uncovered element. To see this fact assume by contradiction that it does not hold. Without loss of generality, A has two uncovered elements x_1, x_2. We claim that there is no $\ell \in L$ that separates the edge $e_1 = \{x_1, v_A\}$ from the edge $e_2 = \{x_2, v_A\}$. We consider the possible landmark edges as follows.

If $\ell = \{s_j, s'_j\}$, then the triple corresponding to ℓ does not contain x_i (for $i = 1, 2$), and we conclude that the distance from ℓ to e_i is 3 (it is not 2 because there is no edge between s'_j and x_i, thus it is 3 using the path through s'_j, a and v_A). Next assume that ℓ connects a vertex of V_2 with a vertex of V_3. Then, for $i = 1, 2$ the distance from ℓ to e_i cannot be 1 because this would imply that ℓ connects a vertex of the form s_j to x_i, and cannot be the case as x_i is not covered. This distance can be 2 for both x_1 and x_2 if ℓ has an end-vertex a in A (using the path through a and v_A) or 3 for both x_1 and x_2 otherwise (as there is no edge between an end-vertex of ℓ and an end-vertex of e_i, there cannot be an edge between two vertices of V_3, and there is a path of two edges between any two vertices of V_3 through v_D). Next, assume that ℓ connects a vertex of V_3 with a vertex of V_4. Note that ℓ is not incident with x_1 or x_2, since they are not covered. Then, the distance from ℓ to each of e_1 and e_2 is 1 if ℓ is incident with v_A, 2, (to both e_1 and e_2) if ℓ is incident with v_D, and 3 otherwise (it is not at most 2 because there is no edge connecting the end-vertices of e_i and the end-vertices of ℓ, and there is a path of length 3 through v_D). Finally, if ℓ connects a vertex of V_4 with a vertex of V_5, then for $i = 1, 2$, the distance from ℓ to e_i is 1 if $\ell = \{v_A, u_A\}$, 2 if $\ell = \{v_D, u_D\}$, and 4 otherwise. In all these cases ℓ does not separate e_1 from e_2.

Therefore A has at most one uncovered element, and similarly B and C have at most one uncovered element (each). Therefore, the collection of chosen triples costs at most $|L'| + 3 \le K + 3$ when it is considered as a solution to C3DM.

Corollary 7.4 *Given a line graph of a bipartite graph G and a value K, deciding whether* $\mathtt{md}(G) \le K$ *is NPC.*

8 Conclusion

We studied the weighted version of the metric dimension optimization problem. We found a number of graph classes where finding a landmark set of minimum cost can be done in polynomial time, and a number of graph classes where the problem is NPC. For each case studied here, the weighted problem and the unweighted problem belong to the same complexity class. Two interesting graph classes for which the complexity of the problem is open are interval graph and series-parallel graphs.

References

1. Babai, L.: On the order of uniprimitive permutation groups. Ann. Math. **113**(3), 553–568 (1981)
2. Beerliova, Z., Eberhard, F., Erlebach, T., Hall, A., Hoffmann, M., Mihalák, M., Ram, L.S.: Network discovery and verification. IEEE J. Sel. Areas Commun. **24**(12), 2168–2181 (2006)
3. Cáceres, J., Hernando, M.C., Mora, M., Pelayo, I.M., Puertas, M.L., Seara, C., Wood, D.R.: On the metric dimension of cartesian products of graphs. SIAM J. Discr. Math. **21**(2), 423–441 (2007)
4. Chartrand, G., Eroh, L., Johnson, M.A., Oellermann, O.R.: Resolvability in graphs and the metric dimension of a graph. Discr. Appl. Math. **105**(1–3), 99–113 (2000)
5. Chartrand, G., Zhang, P.: The theory and applications of resolvability in graphs: A survey. Congressus Numerantium **160**, 47–68 (2003)

6. Chvátal, V.: Mastermind. Combinatorica **3**(3), 325–329 (1983)
7. Corneil, D., Perl, Y., Stewart, L.: A linear recognition algorithm for cographs. SIAM J. Comput. **14**(4), 926–934 (1985)
8. Díaz, J., Pottonen, O., Serna, M.J., van Leeuwen, E.J.: On the complexity of metric dimension. In: Epstein, L., Ferragina, P. (eds.) ESA, Lecture Notes in Computer Science, vol. 7501, pp. 419–430. Springer, Berlin (2012)
9. Harary, F., Melter, R.: The metric dimension of a graph. Ars Combinatoria **2**, 191–195 (1976)
10. Hartung, S., Nichterlein, A.: On the parameterized and approximation hardness of metric dimension. In: Umans, C. (ed.) Proceedings of IEEE Conference on Computational Complexity (CCC), pp. 266–276 (2013)
11. Hauptmann, M., Schmied, R., Viehmann, C.: Approximation complexity of metric dimension problem. J. Discr. Algorithms **14**, 214–222 (2012)
12. Hoffmann, S., Wanke, E.: Metric dimension for Gabriel unit disk graphs is NP-complete. In: Bar-Noy, A., Halldórsson, Magnús, M. (eds.) ALGOSENSORS: Lecture Notes in Computer Science, vol. 7718, pp. 90–92. Springer, Berlin (2013). Also in CoRR, abs/1306.2187 (2013)
13. Johnson, D.: The NP-completeness column: an ongoing guide. J. Algorithms **6**(3), 434–451 (1985)
14. Khuller, S., Raghavachari, B., Rosenfeld, A.: Landmarks in graphs. Discr. Appl. Math. **70**(3), 217–229 (1996)
15. Melter, R.A., Tomescu, I.: Metric bases in digital geometry. Comput. Vis. Graph. Image Process. **25**, 113–121 (1984)
16. Sebö, A., Tannier, E.: On metric generators of graphs. Math. Oper. Res. **29**(2), 383–393 (2004)
17. Shanmukha, B., Sooryanarayana, B., Harinath, K.S.: Metric dimension of wheels. Far East J. Appl. Math. **8**(3), 217–229 (2002)
18. Slater, P.J.: Leaves of trees. Congressus Numerantium **14**, 549–559 (1975)
19. Wojciechowski, J.: Minimal equitability of hairy cycles. J. Combin. Math. Combin. Comput. **57**, 129–150 (2006)