Varianten der Metrischen Dimension spezieller Graphklassen

Inaugural-Dissertation

zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

Yannick Schmitz
aus Düsseldorf

Düsseldorf, April 2024

aus dem Institut für Informatik
der Heinrich-Heine-Universität

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Berichterstatter:

1. Prof. Dr. Egon Wanke

2. Prof. Dr. Jörg Rothe

Tag der mündlichen Prüfung: 18.07.2024

Eidesstattliche Erklärung

Ich versichere an Eides Statt, dass die Dissertation von mir selbständig und ohne unzulässige fremde Hilfe unter Beachtung der „Grundsätze zur Sicherung guter wissenschaftlicher Praxis an der Heinrich-Heine-Universität Düsseldorf“ erstellt worden ist.

Diese Dissertation wurde bisher keiner anderen Fakultät vorgelegt. Es gab bislang keine erfolglosen oder erfolgreichen Promotionsversuche.

Ort, Datum

Unterschrift

Danksagung

Diese Arbeit wäre nicht möglich gewesen, ohne die Hilfe einiger Personen, bei denen ich mich an dieser Stelle bedanken möchte.

Allen voran gilt mein Dank meinem Doktorvater Prof. Dr. Egon Wanke. Dank ihm wurde mein Interesse an der theoretischen Informatik geweckt und er war es auch, der mir das Problem der Metrischen Dimension vorgestellt hat. Ich bedanke mich für die jahrelange Zusammenarbeit, die hilfreichen Diskussionen und die unterhaltsamen Gespräche.

Mein Dank gilt auch Dr. Duygu Vietz, die mein Interesse an diesem Thema ebenfalls unterstützt und meine vorherigen Abschlussarbeiten betreut hat sowie Prof. Dr. Jörg Rothe für das Übernehmen des Zweitgutachtens. Ferner meiner ehemaligen Arbeitskollegin Dr. Dominique Komander für die freundliche Unterstützung und Aufnahme in den Arbeitskreis zu Beginn meiner Zeit am Lehrstuhl und meinem ehemaligen Arbeitskollegen Martin Breuer für die fachliche Zusammenarbeit und die Schachpartien.

Zudem bedanke ich mich bei Nguyen Khoa Tran, Sara Schulte, Philipp Spohr, Laura Kühle, Sven Schrinner, Max Ried, Prof. Dr. Gunnar Klau und Dr. Daniel Schmidt vom Lehrstuhl für Algorithmische Bioinformatik für den fachlichen Austausch und die weniger fachlichen Unterhaltungen in den Mittagspausen. Im Besonderen danke ich Nguyen Khoa Tran für die fachliche Zusammenarbeit sowie Nguyen Khoa Tran und Sara Schulte für die Hilfestellung bei Fragen zwischendurch und die außeruniversitären Unternehmungen.

Ein besonderer Dank gilt meiner Familie, die mich viele Jahre lang unterstützt hat und darüber hinaus meiner Schwester Dr. Alexa Schmitz für die Hilfe bei organisatorischen Fragen.

Abschließend bedanke ich mich bei meinen Freunden Jimmy, Jannik, Khoa, Alex, Christian und Mi für die unzähligen Stunden außerhalb der Arbeit. Danke, dass ihr mich so oft zum Lachen gebracht habt.

Zusammenfassung

Zwei Knoten u, v eines ungerichteten Graphen G werden durch einen Knoten w *getrennt*, wenn sich die Distanz zwischen u und w von der Distanz zwischen v und w unterscheidet. Eine Teilmenge U der Knotenmenge von G ist eine *trennende Menge* für G, wenn jedes Knotenpaar durch mindestens einen Knoten aus U getrennt wird. Die *Metrische Dimension* von G entspricht der Größe einer kleinsten trennenden Menge für G. Das Entscheidungsproblem METRISCHE DIMENSION ist die Frage, ob G eine trennende Menge der Größe $\leq k$ besitzt, für einen gegebenen ungerichteten Graphen G und eine gegebene Zahl k.

Dieses Problem wurde bereits in den 1970ern eingeführt und für zahlreiche Graphklassen untersucht. Zu den bekannteren Ergebnissen zählen zum Beispiel die NP-Vollständigkeit für allgemeine Graphen, planare Graphen, Intervallgraphen, Splitgraphen und Gabriel-Unit-Disc Graphen sowie effiziente Algorithmen für außenplanare Graphen, Co-Graphen, Wheels, Kaktusblockgraphen und Bäume.

Seit der Einführung der Metrischen Dimension wurden zahlreiche Varianten des Problems formuliert und untersucht, beispielsweise die LOKALE METRISCHE DIMENSION, die ADJAZENZDIMENSION, die k-METRISCHE DIMENSION, die EDGE METRIC DIMENSION und die MIXED METRIC DIMENSION, die STARKE METRISCHE DIMENSION und in der gerichteten Version die GERICHTETE METRISCHE DIMENSION und die GERICHTETE STARKE METRISCHE DIMENSION.

Auch in der vorliegenden Arbeit werden einige Varianten der Metrischen Dimension auf speziellen Graphklassen betrachtet. Die Arbeit ist wie folgt aufgebaut: Nach einer Einleitung und der Definition grundlegender Begriffe in Kapitel 1 und Kapitel 2 werden in Kapitel 3 die oben genannten Varianten der Metrischen Dimension vorgestellt. Hierzu wird die Problemstellung der jeweiligen Variante definiert sowie ein Überblick über einige dazugehörige Ergebnisse gegeben. In Kapitel 4 wird zu einer der Varianten, der k-Metrischen Dimension, ein NP-Vollständigkeitsbeweis erbracht. Anschließend wird in Kapitel 5 die gerichtete Variante der Metrischen Dimension betrachtet. Zuerst wird ein NP-Vollständigkeitsbeweis für diese Variante für DAGs erbracht, danach werden ungerichtete und gerichtete Co-Graphen definiert. Für letztere wird ein Algorithmus vorgestellt, der die gerichtete Metrische Dimension in linearer Zeit berechnet. In Kapitel 6 folgen Ergebnisse zur Starken Metrischen Dimension. Zu Beginn des Kapitels wird ein Linearzeitalgorithmus für ungerichtete Co-Graphen gezeigt. Danach werden Graphen definiert, die durch die Verknüpfung mehrerer ungerichteter Graphen entstehen. Für diese Graphen wird der strong resolving Graph betrachtet, auf dessen Grundlage die Starke Metrische Dimension berechnet werden kann. Ein entsprechender Algorithmus, der die Starke Metrische Dimension eines Graphen berechnet, indem er den Graphen in seine zweifachen Zusammenhangskomponenten zerlegt, wird am Beispiel dreier Graphklassen erläutert. Außerdem wird der strong resolving Graph distanzerhaltender Graphen bestimmt. Abschließend wird in Kapitel 7 die gerichtete Starke Metrische Dimension betrachtet. Auch für diese Variante wird ein Linearzeitalgorithmus für gerichtete Co-Graphen vorgestellt, welcher auf den Ergebnissen des vorherigen Kapitels aufbaut. Am Ende der Arbeit finden sich ein Ausblick sowie die Veröffentlichungen und die Manuskripte, auf denen diese Arbeit basiert.

Abstract

Two vertices u, v of an undirected graph G are *resolved* by a vertex w if the distance between u and w differs from the distance between v and w. A subset U of vertices of G is a *resolving set* for G if every pair of vertices is resolved by at least one vertex of U. The *metric dimension* of G is the size of a smallest resolving set for G. The decision problem METRIC DIMENSION is the question whether there exists a resolving set of size $\leq k$ for G, for a given undirected graph G and a given number k.

This problem was already introduced in the 1970s and was studied for numerous classes of graphs. The more known results include for example the NP-completeness for general graphs, planar graphs, interval graphs, split graphs and Gabriel unit disc graphs, as well as efficient algorithms for outerplanar graphs, co-graphs, wheels, cactus-block graphs and trees.

Since the introduction of the metric dimension, several variants were defined and studied, for example the LOCAL METRIC DIMENSION, the ADJACENCY DIMENSION, the k-METRIC DIMENSION, the EDGE METRIC DIMENSION and the MIXED METRIC DIMENSION, the STRONG METRIC DIMENSION and for the directed version the DIRECTED METRIC DIMENSION and the DIRECTED STRONG METRIC DIMENSION.

In this thesis several variants of the metric dimension are studied for certain classes of graphs as well. It is structured as follows: After an introduction and the definition of basic terms in chapter 1 and chapter 2, the variants of the metric dimension mentioned above will be introduced in chapter 3. Their definition as well as a summary of some known results will be given. In chapter 4 it is proven that k-metric dimension is NP-complete. In Chapter 5 the directed metric dimension is considered. At first an NP-completeness proof for DAGs is provided, afterwards undirected and directed co-graphs are defined. An algorithm for computing the directed metric dimension of directed co-graphs in linear time is presented at the end of the chapter. Next, the strong metric dimension is studied in chapter 6. The beginning of the chapter contains a linear time algorithm for undirected co-graphs. Then, graphs which result from merging several undirected graphs are definied. The strong resolving graphs of those graphs are studied, based on which the strong metric dimension can be calculated. An algorithm which computes the strong metric dimension of a graph by decomposing it into its biconnected components explained using grids, cycles and co-graphs as examples. Furthermore, the strong resolving graphs of distance hereditary graphs are determined. Finally, the directed strong metric dimension is studied in chapter 7. A linear time algorithm for directed co-graphs is presented for this variant as well, based on the results of the previous chapters. At the end an outlook as well as a list of the publications and manuscript this thesis is based on is given.

Inhaltsverzeichnis

1 Einleitung

Die vorliegende Arbeit basiert auf den Arbeiten [110, 111, 112, 127]. Die Ergebnisse wurden zu einer Gesamtarbeit zusammengefügt, durch neue Abbildungen verdeutlicht, Beweise wurden angepasst, Notationen vereinheitlicht und noch unveröffentlichte Ergebnisse hinzugefügt.

Ein ungerichteter Graph G besteht aus einer Knotenmenge $V(G)$ und einer Kantenmenge $E(G)$. Dabei ist $V(G)$ eine endliche Menge von Elementen und $E(G)$ eine Menge von Knotenpaaren. Zwei Knoten $u, v \in V(G)$ werden von einem Knoten $w \in V(G)$ *getrennt*, wenn $d_G(u, w) \neq d_G(v, w)$, wobei $d_G(u, v)$ die Länge eines kürzesten Weges zwischen u und v in G ist. Eine Teilmenge der Knoten $U \subseteq V(G)$ heißt *trennende Menge* für G, wenn jedes Knotenpaar in G durch mindestens einen Knoten aus U getrennt wird. Eine minimale trennende Menge wird auch *metrische Basis* für G genannt und die Knoten einer trennenden Menge werden als *Ankerknoten* bezeichnet. Die *Metrische Dimension* von G entspricht der Größe einer metrischen Basis für G. Das Entscheidungsproblem METRISCHE DIMENSION ist die Frage, ob G eine trennende Menge der Größe $\leq k$ besitzt, für einen gegebenen ungerichteten Graphen G und eine gegebene Zahl k.

Die Metrische Dimension wurde unabhängig voneinander von Slater [118] und von Harary und Melter [54] eingeführt. Die Metrische Dimension findet Anwendung in zahlreichen Gebieten, beispielsweise dem Routen in Sensornetzwerken [43, 20, 61] sowie geographischen Routing-Protokollen [92], der Netzwerkerkennung und -verifikation [13], der kombinatorischen Optimierung [113], der Roboternavigation [75] und der Chemie [22, 57].

Der Bereich der Sensornetzwerke sei an dieser Stelle hervorgehoben, da dieser auch Ursprung der Bezeichnung „Ankerknoten“ ist. Da in einem Sensornetzwerk normalerweise nicht alle Sensoren direkt miteinander kommunizieren können, müssen Nachrichten über andere Sensoren weitergeleitet werden. An jedem Sensor muss somit entschieden werden, wohin die Nachricht weitergegeben wird, wobei garantiert werden muss, dass die Nachricht am Ziel ankommt. Eine Möglichkeit hierfür ist die eindeutige Adressierung durch die Abstände zu einer Menge von ausgezeichneten Sensoren, den sogenannten Ankerknoten. Sowohl die Sensoren, deren Adressen sich unterscheiden müssen, als auch die verwendete Distanzfunktion können angepasst werden, wodurch sich verschiedene Varianten der Metrischen Dimension ergeben. Die Starke Metrische Dimension, eine der hier vorgestellten Varianten, ermöglicht sogar optimales Routen.

Da die Metrische Dimension und ihre Varianten für allgemeine Graphen NP-vollständig sind, sind Graphklassen für die diese Probleme effizient gelöst werden können von Interesse. In der vorliegenden Arbeit geht es darum festzustellen, für welche Graphen die verschiedenen Varianten NP-vollständig oder effizient lösbar sind und wenn möglich Polynomialzeit Algorithmen anzugeben. Sie ist wie folgt aufgebaut: Zuerst werden in Kapitel 2 einige grundlegende Begriffe der Graphentheorie definiert, die in dieser Arbeit Verwendung finden. Anschließend werden in Kapitel 3 einige Varianten der Metrischen Dimension vorgestellt. Im Zuge dessen wird auch diskutiert, wie die Metrische Dimension und ihre Varianten auf gerichtete Graphen übertragen werden können. Für die jeweiligen Varianten wird die Problemstellung definiert sowie ein Überblick über einige dazugehörige Ergebnisse gegeben. In Kapitel 4 wird zu einer der Varianten, der k-Metrischen Dimension, ein NP-Vollständigkeitsbeweis erbracht. Dieser erweitert den bisherigen Forschungsstand, da die NP-Vollständigkeit lediglich für ungerade Werte von k gezeigt wurde. Anschließend wird in Kapitel 5 die gerichtete Variante der Metrischen Dimension betrachtet. Zuerst wird ein NP-Vollständigkeitsbeweis für diese Variante für DAGs erbracht, danach werden ungerichtete und gerichtete Co-Graphen definiert. Für letztere wird ein Algorithmus vorgestellt, der die gerichtete Metrische Dimension in linearer Zeit berechnet. In Kapitel 6 folgen Ergebnisse zur Starken Metrischen Dimension. Zu Beginn des Kapitels wird ein Linearzeitalgorithmus für ungerichtete Co-Graphen gezeigt. Danach werden Graphen definiert, die durch die Verknüpfung mehrerer ungerichteter Graphen entstehen. Für diese Graphen wird der strong resolving Graph betrachtet, auf dessen Grundlage die Starke Metrische Dimension berechnet werden kann. Ein entsprechender Algorithmus, der die Starke Metrische Dimension eines Graphen berechnet indem er den Graphen in seine zweifachen Zusammenhangskomponenten zerlegt, wird am Beispiel von Gittern, Kreisen und Co-Graphen erläutert. Abschließend wird in Kapitel 7 die gerichtete Starke Metrische Dimension betrachtet. Auch für diese Variante wird ein Linearzeitalgorithmus für gerichtete Co-Graphen vorgestellt, welcher auf den Ergebnissen des vorherigen Kapitels aufbaut und eines der ersten Ergebnisse zu dieser Variante darstellt. Am Ende der Arbeit finden sich ein Ausblick sowie die Veröffentlichungen und die Manuskripte, auf denen diese Arbeit basiert.

2 Grundlagen

In diesem Kapitel werden einige grundlegende Begriffe und Konzepte der Graphentheorie definiert. Alle in dieser Arbeit betrachteten Graphen sind endlich und einfach.

Definition 1 (ungerichteter Graph)**.** *Ein* ungerichteter Graph *besteht aus einer endlichen Menge von Knoten $V(G)$ und einer Menge von Kanten $E(G) \subseteq \{\{u,v\} \mid u,v \in V(G), u \neq v\}$. Die beiden Knoten u,v sind die* Endknoten *der Kante $\{u,v\}$. Die beiden Endknoten einer Kante sind* adjazent *zueinander und* inzident *zu der Kante.*

Definition 2 (gerichteter Graph)**.** *Ein* gerichteter Graph *besteht aus einer endlichen Menge von Knoten $V(G)$ und einer Menge von Kanten $E(G) \subseteq \{(u,v) \mid u,v \in V(G), u \neq v\}$. Der Knoten u ist der* Startknoten *und der Knoten v der* Endknoten *der Kante (u,v).*

Definition 3 ((induzierter) Teilgraph)**.** *Sei G ein ungerichteter Graph. Ein* Teilgraph *von G ist ein Graph G' mit Knotenmenge $V(G') \subseteq V(G)$ und Kantenmenge $E(G') \subseteq \{\{u,v\} \in E(G) \mid u,v \in V(G')\}$.*
G' ist ein induzierter Teilgraph*, wenn für die Kantenmenge $E(G') = \{\{u,v\} \in E(G) \mid u,v \in V(G')\}$ gilt. Der Graph G' wird durch die Knotenmenge $V(G')$* induziert.

Definition 4 (Nachbarschaft)**.** *Sei G ein ungerichteter Graph und sei $u \in V(G)$. Die* offene Nachbarschaft *von u ist $N(u) = \{v \in V(G) \mid \{u,v\} \in E(G)\}$. Die* geschlossene Nachbarschaft *von u ist $N[u] = \{v \in V(G) \mid \{u,v\} \in E(G)\} \cup \{u\}$.*

Sei G ein gerichteter Graph und sei $u \in V(G)$. Die in-Nachbarn *von u sind $N^{-}(u) = \{v \in V(G) \mid (v, u) \in E(G)\}$. Die* out-Nachbarn *von u sind $N^{+}(u) = \{v \in V(G) \mid (u, v) \in E(G)\}$.*

Definition 5 (Twins)**.** *Sei G ein ungerichteter Graph und seien $u, v \in V(G)$. Die beiden Knoten u, v heißen* false twins*, wenn $N(u) = N(v)$. Die beiden Knoten heißen* true twins*, wenn $N[u] = N[v]$.*

Definition 6 ((kürzester) Weg)**.** *Sei G ein ungerichteter Graph. Eine Folge $\{v_1, \ldots, v_k\}$ ist ein* Weg *der Länge k von v_1 nach v_k, wenn $\{v_i, v_{i+1}\} \in E(G)$ für $1 \leq i \leq k-1$.*
Sei G ein gerichteter Graph. Eine Folge $\{v_1, \ldots, v_k\}$ ist ein Weg *der Länge k von v_1 nach v_k, wenn $(v_i, v_{i+1}) \in E(G)$ für $1 \leq i \leq k-1$.*
Ein Weg der Länge k ist ein kürzester Weg *von u nach v, wenn es keinen Weg der Länge k' mit $k' < k$ von u nach v in G gibt.*

Definition 7 (Distanz und Durchmesser)**.** *Sei G ein ungerichteter Graph. Die* Distanz *zwischen zwei Knoten $u, v \in V(G)$ ist die Länge eines kürzesten Weges von u nach v in G.*
Sei G ein gerichteter Graph. Die Distanz von einem Knoten $u \in V(G)$ zu einem Knoten $v \in V(G)$ ist die Länge eines kürzesten Weges von u nach v in G.
Der Durchmesser eines Graphen entspricht der maximalen Distanz zwischen zwei Knoten, beziehungsweise von einem Knoten zu einem anderen Knoten.

Definition 8 ((k-facher) Zusammenhang)**.** *Ein ungerichteter Graph G ist* zusammenhängend*, wenn zwischen jedem Knotenpaar $u, v \in V(G)$ ein Weg in G existiert.*
G ist k-fach zusammenhängend*, wenn es zwischen jedem Knotenpaar k knotendisjunkte Wege gibt.*

Definition 9 ((k-fache) Zusammenhangskomponente)**.** *Sei G ein ungerichteter Graph. Eine* Zusammenhangskomponente *von G ist ein maximaler zusammenhängender Teilgraph von G.*
Eine k-fache Zusammenhangskomponente *von G ist ein maximaler k-fach zusammenhängender Teilgraph von G.*

Definition 10 (Separationsknoten)**.** *Ein Knoten $u \in V(G)$ ist ein* Separationsknoten*, wenn der durch $V(G) \setminus \{u\}$ induzierte Teilgraph mehr Zusammenhangskomponenten besitzt als G.*

Definition 11 (starker Zusammenhang)**.** *Ein gerichteter Graph G ist* stark zusammenhängend*, wenn von jedem Knoten $u \in V(G)$ zu jedem Knoten $v \in V(G)$ ein gerichteter Weg in G existiert.*

Definition 12 (bipartiter Graph)**.** *Ein ungerichteter Graph G ist* bipartit*, wenn sich die Knoten in zwei disjunkte Teilmengen $V_1, V_2 \subseteq V(G)$ aufteilen lassen, sodass $V_1 \cup V_2 = V(G)$ und die Endknoten jeder Kante aus G in unterschiedlichen Teilmengen liegen.*

Definition 13 (Baum, P_n, C_n, K_n, $K_{n,n}$).
Ein Baum *ist ein ungerichteter, zusammenhängender und kreisfreier Graph.*
Der Weg P_n *ist ein ungerichteter Graph mit Knotenmenge* $\{v_1, \ldots, v_n\}$ *und Kantenmenge* $\{\{v_i, v_{i+1}\} \mid 1 \leq i \leq n-1\}$.
Der Kreis C_n *ist ein ungerichteter Graph mit Knotenmenge* $\{v_1, \ldots, v_n\}$ *und Kantenmenge* $\{\{v_i, v_{i+1}\} \mid 1 \leq i \leq n-1\} \cup \{v_n, v_1\}$. *Ein Graph heißt* kreisfrei*, wenn er keinen Kreis als Teilgraphen enthält.*
Der vollständige Graph K_n *ist ein ungerichteter Graph mit Knotenmenge* $\{v_1, \ldots, v_n\}$ *und Kantenmenge* $\{\{u, v\} \mid u, v \in V(K_n), u \neq v\}$.
Der vollständig bipartite Graph $K_{n,m}$ *ist ein ungerichteter Graph mit Knotenmenge* $\{u_1, \ldots, u_n, v_1, \ldots, v_m\}$ *und Kantenmenge* $\{\{u_i, v_j\} \mid 1 \leq i \leq n, 1 \leq j \leq m\}$.

3 Varianten der Metrischen Dimension

3.1 Metrische Dimension

Die Metrische Dimension wurde unabhängig voneinander 1975 von Slater [118] und 1976 von Harary und Melter [54] für ungerichtete Graphen definiert. In einem ungerichteten Graphen G werden zwei Knoten $u, v \in V(G)$ von einem Knoten $w \in V(G)$ *getrennt*, wenn die Distanz zwischen u und w und die Distanz zwischen v und w unterschiedlich sind. Eine Teilmenge der Knoten $U \subseteq V(G)$ ist eine *trennende Menge* für G, wenn jedes Knotenpaar in $V(G)$ durch mindestens einen Knoten aus U getrennt wird. Formal muss für U gelten

$$\forall u, v \in V(G), u \neq v : \exists w \in U : d_G(u, w) \neq d_G(v, w).$$

Eine kleinste trennende Menge für G wird auch *metrische Basis* für G genannt und die Knoten einer trennenden Menge werden als *Ankerknoten* bezeichnet. Die *Metrische Dimension* von G ist die Größe einer metrischen Basis für G und wird mit md(G) bezeichnet. Das Entscheidungsproblem Metrische Dimension ist wie folgt definiert.

Metrische Dimension	
Gegeben:	Ein ungerichteter Graph G und eine Zahl k.
Frage:	Ist md(G) $\leq$ k?

Bereits 1979 erwähnten Gary und Johnson [46], dass dieses Problem NP-vollständig sei, ein Beweis dafür folgte jedoch erst 1996 von Khuller et al. [75].

Weitere NP-Vollständigkeitsbeweise existieren beispielsweise für planare Graphen [30], Intervallgraphen, Permutationsgraphen [44], Splitgraphen, co-bipartite Graphen, Line-Graphen von bipartiten Graphen [32] und Gabriel-Unit-Disc Graphen [61].

Zudem ist die Metrische Dimension parametrisiert nach dem Standardparameter W[2]-schwer[55] und parametrisiert nach der Baumweite W[1]-schwer [17]. Das letzte Ergebnis wurde außerdem um einen NP-Vollständigkeitsbeweis der Metrischen Dimension für Graphen mit Baumweite 24 erweitert [91].

Die Metrische Dimension eines Graphen lässt sich nicht mit einem Faktor von $(1-\epsilon) \cdot log(|V(G)|)$ für ein $\epsilon > 0$ approximieren [56], jedoch existieren FPT-Lösungen für spezielle Graphklassen, wie zum Beispiel Intervallgraphen [44] oder Graphen mit beschränkter Baumlänge [14].

Des Weiteren existieren Polynomialzeit-Lösungen für beispielsweise außenplanare Graphen [30], (gewichtete) Co-Graphen, Bäume mit k zusätzlichen Kanten, Wheels, Kreise, Wege und vollständige Graphen [32], Kaktus-Blockgraphen [60] und Bäume [75, 32]. Ferner existieren Charakterisierungen für Graphen mit Metrischer Dimension 1, $|V(G)|-1$ und $|V(G)|-2$ [22].

Darüber hinaus wurde die Metrische Dimension für Graphen untersucht, die feste Strukturen aufweisen, wie zum Beispiel das Coronaprodukt [131], das lexikographische [109, 66] und das kartesische Produkt von Graphen [19, 67], verknüpfte Graphen [85, 117] und verknüpfte Kreise [64] und auch für die Zerlegung von Graphen in EBCs [125].

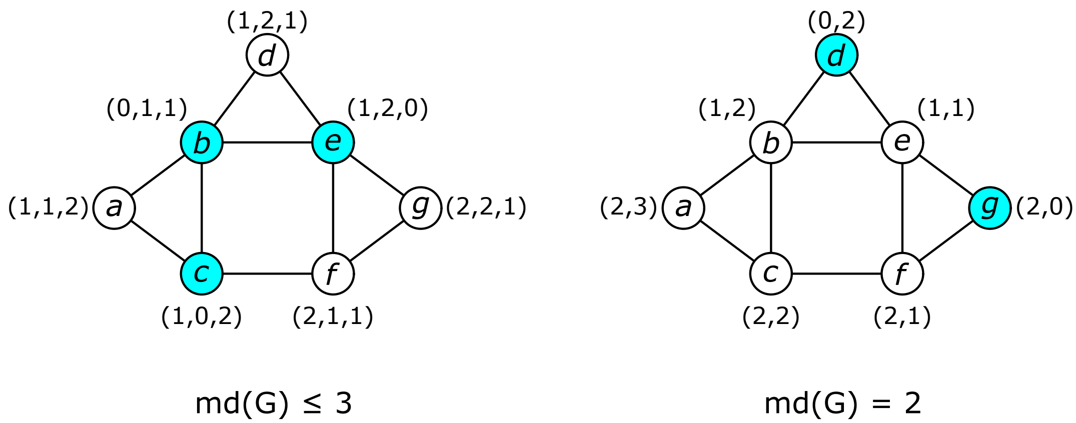


Abbildung 1: Ein ungerichteter Graph G mit einer in blau gefärbten, nicht verkleinerbaren trennenden Menge der Größe 3 (links) und einer metrischen Basis der Größe 2 (rechts). An den Knoten stehen die Distanzen zu den Ankerknoten. Die Distanzvektoren unterscheiden sich paarweise in mindestens einer Position. Da offensichtlich keine trennende Menge der Größe 1 existiert, ist md(G) = 2.

3.2 Lokale Metrische Dimension

Eine mögliche Variante der Metrischen Dimension ist die lokale Metrische Dimension, bei der nur benachbarte Knoten voneinander getrennt werden müssen. Somit ist $U \subseteq V(G)$ eine *lokal trennende Menge* für G, wenn gilt

$$\forall u, v \in V(G), \{u, v\} \in E(G), u \neq v : \exists w \in U : d_G(u, w) \neq d_G(v, w).$$

Analog zur Metrischen Dimension lassen sich eine *lokale metrische Basis* und die *lokale Metrische Dimension*, die mit lmd(G) bezeichnet wird, definieren. Da jede trennende Menge auch eine lokal trennende Menge ist, gilt md(G) $\geq$ lmd(G). Das Entscheidungsproblem LOKALE METRISCHE DIMENSION ist wie folgt definiert.

LOKALE METRISCHE DIMENSION	
Gegeben:	Ein ungerichteter Graph G und eine Zahl k.
Frage:	Ist lmd(G) $\leq$ k?

Die lokale Metrische Dimension wurde von Okamoto et al. eingeführt [100] und ist NP-vollständig für allgemeine [42] und auch planare Graphen [124]. Sie wurde zum Beispiel für das Coronaprodukt [9], das starke Produkt [10] und das hierarchische Produkt von Graphen [77] untersucht.

Ebenfalls existieren Charakterisierungen für Graphen mit lokaler Metrischer Dimension 1 und $|V(G)| - 1$ [100].

Viele Ergebnisse der Metrischen Dimension lassen sich direkt oder nach geringfügiger Anpassung für die lokale Metrische Dimension übernehmen. Beispielsweise kann auch der NP-Vollständigkeitsbeweis von Khuller et al. [75] für die lokale Metrische Dimension erweitert werden. Ein wesentlicher Unterschied ist, dass bei der lokalen Metrischen Dimension keine false twins getrennt werden müssen, welche auch in späteren Teilen dieser Arbeit für die Ergebnisse zu weiteren Varianten der Metrischen Dimension von Bedeutung sind.

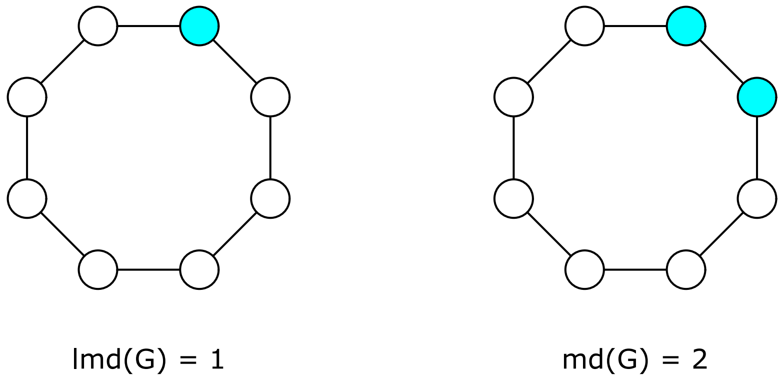


Abbildung 2: Ein Kreis mit 6 Knoten, mit einer lokal trennenden Menge (links) und einer trennenden Menge (rechts). Für die (reguläre) Trennung werden mindestens zwei Knoten benötigt, da ein Knoten seine beiden Nachbarn nicht voneinander trennt. In der lokalen Variante ist dies nicht nötig, sodass gerade Kreise die lokale Metrische Dimension 1 und Metrische Dimension 2 haben.

3.3 Adjazenzdimension

Nicht nur die zu trennenden Knotenpaare können eingeschränkt werden, sondern auch die verwendete Distanzfunktion, was bei der Adjazenzdimension der Fall ist. Eine Menge $U \subseteq V(G)$ ist eine *adjazent trennende Menge* für G, wenn gilt

$$\forall u, v \in V(G), u \neq v : \exists w \in U : d_{adj}(u, w) \neq d_{adj}(v, w),$$

wobei $d_{adj}(u, v) = min\{d_G(u, v), 2\}$. Analog zur Metrischen Dimension lassen sich eine *adjazente metrische Basis* und die *Adjazenzdimension*, die mit amd(G) bezeichnet wird, definieren. Da jede adjazent trennende Menge auch eine trennende Menge ist, gilt amd(G) $\geq$ md(G). Das Entscheidungsproblem Adjazenzdimension ist wie folgt definiert.

Adjazenzdimension	
Gegeben:	Ein ungerichteter Graph G und eine Zahl k.
Frage:	Ist amd(G) $\leq$ k?

Die Adjazenzdimension wurde von Jannesari und Omoomi eingeführt [66] und ist NP-vollständig für allgemeine Graphen, jedoch parametrisiert nach dem Standardparameter FPT [42].

Auch dieses Problem wurde für das lexikographische Produkt von Graphen untersucht [37] und auch für diese Variante existieren Charakterisierungen für Graphen mit Adjazenzdimension 1 und $|V(G)| - 1$ [66].

Die Adjazenzdimension kann außerdem verallgemeinert werden, sodass die Distanzfunktion $d'(u, v) = min\{d_G(u, v), t\}$ betrachtet wird, wie zum Beispiel in [35, 49, 11, 45] geschehen. Diese Variante wird auch als „truncated metric dimension“ bezeichnet.

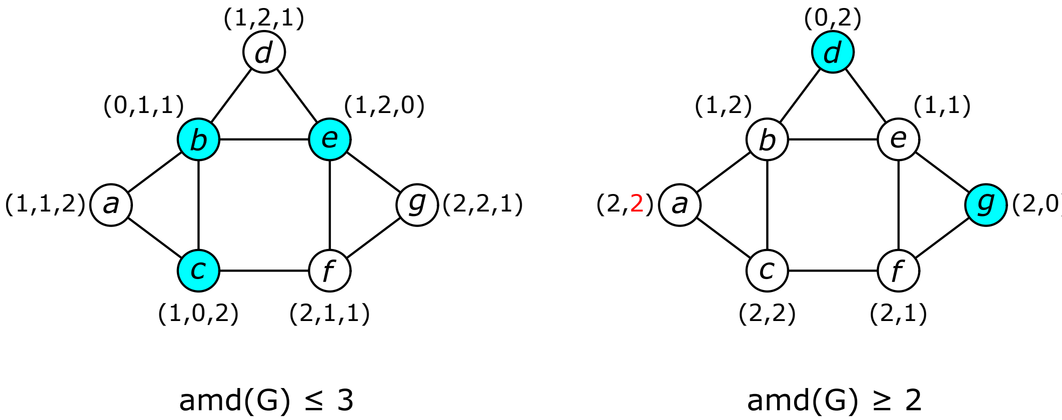


Abbildung 3: Der Graph aus Abbildung 1, mit einer adjazent trennenden Menge der Größe 3 (links). Die trennende Basis aus Abbildung 1 ist keine adjazent trennende Menge für diesen Graphen, da die Knoten a und c nicht getrennt werden. Die durch die veränderte Distanzfunktion angepasste Distanz zwischen a und g ist rot markiert (rechts). Für diesen Graphen gilt amd(G) = 3.

3.4 k-Metrische Dimension

Bei der k-Metrischen Dimension wird die herkömmliche Trennung betrachtet, jedoch wird gefordert, dass jedes Knotenpaar durch k Ankerknoten getrennt wird. Folglich ist $U \subseteq V(G)$ eine *k-fach trennende Menge* für G, wenn gilt

$$\forall u, v \in V(G), u \neq v : \exists w_1, \dots, w_k \in U : d_G(u, w_i) \neq d_G(v, w_i), 1 \leq i \leq k.$$

Analog zur Metrischen Dimension lassen sich eine *k-metrische Basis* und die *k-Metrische Dimension*, die mit kmd(G) bezeichnet wird, definieren. Da jede k-fach trennende Menge auch eine trennende Menge ist, gilt kmd(G) $\geq$ md(G). Außerdem gilt offensichtlich $(\mathrm{k}+1)\mathrm{md}(G) \geq$ kmd(G) Das Entscheidungsproblem k-Metrische Dimension ist wie folgt definiert.

k-Metrische Dimension	
Gegeben:	Ein ungerichteter Graph G und eine Zahl r.
Frage:	Ist kmd(G) $\leq$ r?

Die k-Metrische Dimension wurde von Estrada-Moreno et al. eingeführt [34]. Die NP-Vollständigkeit wurde für ungerade Werte von k in [35] gezeigt und für beliebige Werte in [110]. Der Beweis für beliebige Werte wird auch in Kapitel 4 aufgegriffen.

Das Problem wurde für Bäume [130] und (gewichtete) vollständige Graphen, vollständig bipartite Graphen und Wheels untersucht [3]. Darüber hinaus wurden das lexikographische Produkt [36] und das hierarchische Produkt von Graphen sowie die Verbindung und die Verknüpfung von Graphen[76] bezüglich der k-Metrischen Dimension betrachtet.

Für $k = 1$ entspricht die k-Metrische Dimension der klassischen Variante der Metrischen Dimension. Der Fall $k = 2$ ist auch als fehlertolerante Metrische Dimension bekannt und wurde von Hernando et al. eingeführt [59]. Entsprechend der vorherigen Definition ist $U \subseteq V(G)$ eine *fehlertolerante trennende Menge* für G, wenn gilt

$$\forall u, v \in V(G), u \neq v : \exists w_1, w_2 \in U : d_G(u, w_1) \neq d_G(v, w_1) \wedge d_G(u, w_2) \neq d_G(v, w_2).$$

Namensgebend ist die äquivalente Definition, dass $U \subseteq V(G)$ eine fehlertolerante trennende Menge für G ist, wenn gilt

$$\forall w \in U : U \setminus \{w\} \text{ ist eine trennende Menge für } G.$$

Analog zur Metrischen Dimension lassen sich eine *fehlertolerante metrische Basis* und die *fehlertolerante Metrische Dimension*, die mit fmd(G) bezeichnet wird, definieren. Da jede fehlertolerante trennende Menge auch eine trennende Menge ist, gilt fmd(G) $\geq$ md(G). Das Entscheidungsproblem FEHLERTOLERANTE METRISCHE DIMENSION ist wie folgt definiert.

FEHLERTOLERANTE METRISCHE DIMENSION	
Gegeben:	Ein ungerichteter Graph G und eine Zahl r.
Frage:	Ist fmd(G) $\leq$ r?

Im Gegensatz zu dem Fall $k \geq 3$ ist die fehlertolerante Metrische Dimension jedes Graphen definiert. Insbesondere Graphen mit twins besitzen keine k-trennende Mengen für $k \geq 3$, weswegen die fehlertolerante Metrische Dimension häufig separat betrachtet wird. Beispielsweise wurde die fehlertolerante Metrische Dimension von Bäumen [130], Cographen [126] und Sonnengraphen [124] untersucht. Außerdem wurden die Graphen charakterisiert, deren fehlertolerante Metrische Dimension um genau eins größer ist als ihre Metrische Dimension [108].

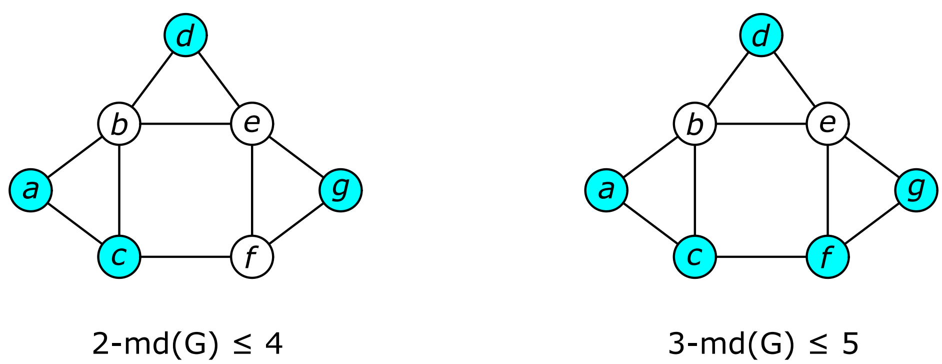


Abbildung 4: Der Graph aus Abbildung 1 mit einer fehlertoleranten trennenden Menge (links) und einer 3-fach trennenden Menge (rechts). Die gesamte Knotenmenge ist eine 4-fach trennende Menge. Für diesen Graphen existiert keine 5-fach trennende Menge, da die Knoten b und d nur durch sich selbst und durch a und c getrennt werden. Analog werden d und e nur durch d, e, f und g getrennt.

3.5 Edge Metric Dimension und Mixed Metric Dimension

Eine Variante der Metrischen Dimension, die erst seit kurzem untersucht wird, ist die Kanten Metrische Dimension (eng.: Edge Metric Dimension), bei der nicht die Knoten, sondern die Kanten eines Graphen getrennt werden müssen. Somit ist $U \subseteq V(G)$ eine *Kanten trennende Menge* für G, wenn gilt

$$\forall e_1, e_2 \in E(G), e_1 \neq e_2 : \exists w \in U : d_G(e_1, w) \neq d_G(e_2, w).$$

Hierbei ist für eine Kante $\{u, v\} \in E(G)$ die Distanz zu einem Knoten w definiert als $d_G(e, w) = min\{d_G(u, w), d_G(v, w)\}$. Analog zur Metrischen Dimension lassen sich eine *Kanten metrische Basis* und die *Kanten Metrische Dimension*, die mit emd(G) bezeichnet wird, definieren. Das Entscheidungsproblem Kanten Metrische Dimension ist wie folgt definiert.

Kanten Metrische Dimension	
Gegeben:	Ein ungerichteter Graph G und eine Zahl k.
Frage:	Ist emd(G) $\leq$ k?

Die Kanten Metrische Dimension wurde von Kelenc et al. [73] eingeführt, die ebenfalls zeigten, dass sie NP-vollständig für allgemeine Graphen ist. Die Metrische Dimension und die Kanten Metrische Dimension sind nicht durch die jeweils andere beschränkt [79], es wurden jedoch Graphen untersucht, für die die Kanten Metrische Dimension durch die Metrische Dimension beschränkt werden kann [79, 80, 48]. Außerdem wurden beispielsweise einige planare Graphen [128],Gitter [48, 47], Kaktusgraphen [115] und unicyclic Graphen [116] betrachtet.

Des Weiteren wurde diese Variante für verschiedene Produkte von Graphen untersucht, zum Beispiel das kartesische Produkt [73, 135], das hierarchische Produkt [78] sowie das lexikographische Produkt, das Coronaprodukt und Joingraphen [101].

Es existieren Charakterisierungen für Graphen mit Kanten Metrischer Dimension $|V(G)| - 1$ und $|V(G) - 2|$ [135, 134, 47], die Auswirkung des Entfernens von Knoten und Kanten wurde in [129] betrachtet und die Approximierbarkeit wurde in [63] untersucht.

Neben der Kanten Metrischen Dimension wurde die Gemischte Metrische Dimension (eng.: Mixed Metric Dimension) definiert, bei der nicht nur die Kanten, sondern Knoten und Kanten getrennt werden müssen. Somit ist $U \subseteq V(G)$ eine *gemischt trennende Menge* für G, wenn gilt

$$\forall u, v \in V(G) \cup E(G), u \neq v : \exists w \in U : d_G(u, w) \neq d_G(v, w).$$

Analog zur Metrischen Dimension lassen sich eine *gemischte metrische Basis* und die *Gemischte Metrische Dimension*, die mit mmd(G) bezeichnet wird, definieren. Das Entscheidungsproblem Gemischte Metrische Dimension ist wie folgt definiert.

Gemischte Metrische Dimension	
Gegeben:	Ein ungerichteter Graph G und eine Zahl k.
Frage:	Ist mmd(G) $\leq$ k?

Die Gemischte Metrische Dimension wurde von Kelenc et al. [72] eingeführt und ist ebenfalls NP-vollständig für allgemeine Graphen.

Diese Variante wurde beispielsweise für Bäume, vollständig bipartite Graphen [72], unicycliv Graphen [114], einige planare Graphen [102, 70] und das kartesische Produkt [97] betrachtet.

Außerdem wurden die Abhängigkeiten zwischen der Kanten Metrischen Dimension, der Gemischten Metrischen Dimension und der Starken Metrischen Dimension untersucht [28].

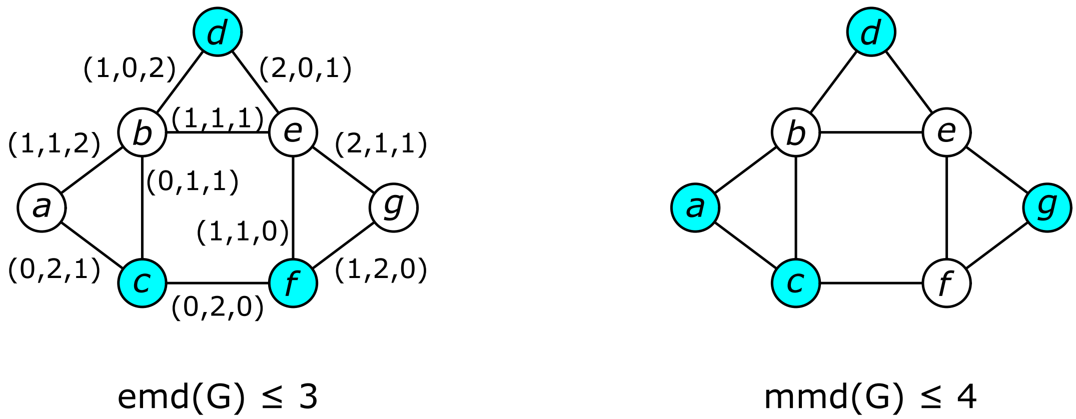


Abbildung 5: Der Graph aus Abbildung 1 mit einer Kanten trennenden Menge (links) und einer gemischt trennenden Menge (rechts). Im linken Bild stehen an den Kanten die Distanzen zu den Ankerknoten. Für eine gemischt trennende Menge muss der Knoten a gewählt werden, um den Knoten b von der Kante $\{a, b\}$ zu trennen. Analog muss der Knoten g ein Ankerknoten sein. Der Knoten b und die Kante $\{b, d\}$ werden nur durch den Knoten d getrennt (analog e und $\{d, e\}$). Anschließend müssen noch der Knoten b und die Kante $\{b, c\}$ getrennt werden. Hierfür kann der Knoten c oder f verwendet werden (genauso wie zur Trennung von e und $\{e, f\}$).

3.6 Gerichtete Metrische Dimension

3.6.1 Übertragbarkeit der Metrischen Dimension auf gerichtete Graphen

Das Problem Metrische Dimension sowie die vorgestellten Varianten können auch auf gerichteten Graphen betrachtet werden. Viele Aspekte lassen sich analog definieren, allerdings müssen einige zusätzliche Überlegungen angestellt werden. Denn in einem gerichteten Graphen G können sich, anders als in einem ungerichteten Graphen, die Distanzen $d_G(u, v)$ und $d_G(v, u)$ unterscheiden. Demnach muss als erstes unterschieden werden, ob die Distanzen von oder zu den Ankerknoten betrachtet werden. Jedoch entspricht die eine Distanz der jeweils anderen, wenn alle Kanten in G umgedreht werden, siehe Abbildung 6.

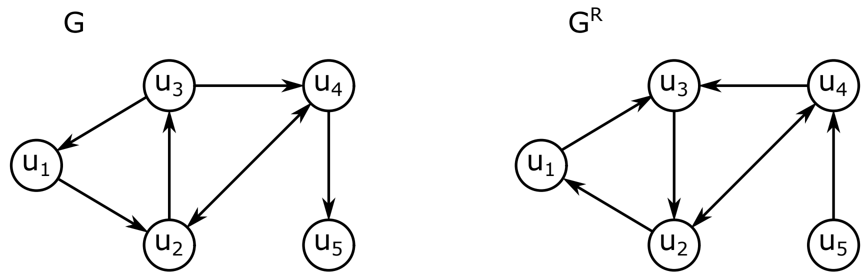


Abbildung 6: Ein Graph G (links) und sein reverser Graph G^R (rechts). Für jede Kante $(u,v) \in E(G)$ wird eine Kante $(v,u) \in E(G^R)$ erzeugt. Da die Distanz $d_G(u_1,u_3) = 2$ ist, ist ebenfalls $d_{G^R}(u_3,u_1) = 2$. Zwischen den Knoten u_2 und u_4 existieren in G die Kanten in beide Richtungen und folglich sind auch beide Kanten in G^R. Von u_5 gibt es in G keinen Weg zu u_2, sodass es in G^R auch keinen Weg von u_2 nach u_5 gibt.

Der Graph, der entsteht wenn jede Kante (u,v) in G durch eine Kante (v,u) ersetzt wird, ist auch als reverser Graph von G bekannt und wird mit G^R bezeichnet. Es gilt also $d_G(u,v) = d_{G^R}(v,u)$, wodurch die betrachtete Richtung für die Metrische Dimension in vielen Fällen lediglich eine Frage der Notation ist. Für denselben Graphen kann es jedoch einen Unterschied machen, welche Richtung betrachtet wird, siehe dazu Abbildung 7. Insbesondere werden im weiteren Verlauf dieser Arbeit gerichtete Co-Graphen betrachtet, welche unter Kanteninvertierung abgeschlossen sind. In dieser Arbeit werden für gerichtete Graphen nur die Distanzen von den Ankerknoten zu den anderen Knoten betrachtet.

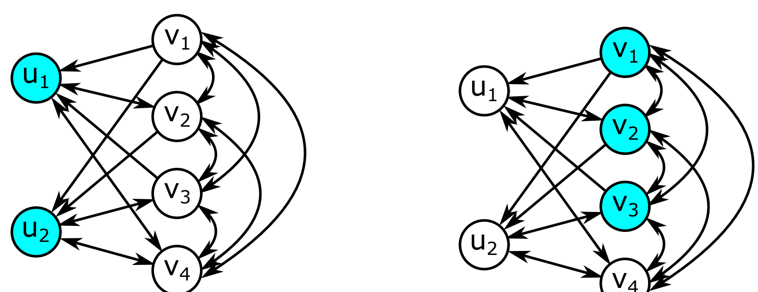


Abbildung 7: Der dargestellte Graph ist stark zusammenhängend und hat eine unterschiedliche Metrische Dimension, je nachdem, ob die Distanzen von (links) oder zu (rechts) den Ankerknoten betrachtet werden. Im ersten Fall reichen die beiden Knoten u_1, u_2 aus um alle Knotenpaare zu trennen. Da der Graph stark zusammenhängend ist, sind die Ankerknoten von allen anderen Knoten getrennt und die Knoten v_1 bis v_4 haben die Distanzvektoren $(2,2),(1,2),(2,1)$ und $(1,1)$. Werden die Distanzen zu den Ankerknoten betrachtet, müssen mindestens drei der vier Knoten v_1, v_2, v_3, v_4 gewählt werden, da jeder dieser Knoten Distanz 1 zu allen anderen Knoten hat.

Des Weiteren gilt es zu beachten, dass in schwach zusammenhängenden Graphen ein Knoten nicht immer von allen anderen Knoten aus erreichbar ist. Somit kann es vorkommen, dass es in einem gerichteten Graphen G keinen Knoten w in G gibt, sodass $d_G(w,u) \neq d_G(w,v)$ für ein Knotenpaar u,v aus G gilt oder dass es sogar nicht einmal einen

Knoten w gibt, von dem aus Wege nach u und v existieren. Dieser Fall kann entweder separat betrachtet, durch die Eigenschaften der Graphen beziehungsweise der Graphklassen ausgeschlossen, oder durch eine dafür eingeführte Distanz behoben werden. Für die letzte Möglichkeit ist das Symbol ∞ geläufig, sodass $d_G(w,u) = \infty$ definiert wird, wenn es keinen Weg von w nach u in G gibt. Diese Distanz wird häufig als unterschiedlich zu den anderen Distanzen definiert, sodass $d_G(w,u) \neq d_G(w,v)$, wenn $d_G(w,u) = \infty$ und $d_G(w,v) \neq \infty$. Folglich würde der Knoten w die beiden Knoten u und v trennen. Diese Möglichkeit kann zusammengefasst dadurch realisiert werden, dass die Distanzfunktion für einen gerichteten Graphen G als $d_G : V(G) \times V(G) \to \mathbb{N} \cup \{\infty\}$ definiert wird.

Der vorangegangene Ansatz behebt zudem ein weiteres Problem, das bei schwach zusammenhängenden Graphen berücksichtigt werden muss. In einem zusammenhängenden ungerichteten Graphen hat ein Ankerknoten zu sich selbst die Distanz 0 und zu allen anderen Knoten eine größere Distanz, sodass diese unterschiedlich sind und der Ankerknoten sich selbst von jedem anderen Knoten trennt. Wie oben angesprochen müssen in einem schwach zusammenhängenden Graphen jedoch nicht zwischen allen Knotenpaaren Wege existieren, was auch für Paare gilt, die Ankerknoten enthalten. Demnach wird ein Ankerknoten w eines Graphen G nicht zwangsweise von einem Knoten u in G getrennt, wenn kein Weg von w nach u in G existiert. Diese Eigenschaft gilt sogar für Knotenpaare, die aus zwei Ankerknoten bestehen. Somit gibt es je nach Definition gerichtete Graphen, für die keine trennende Menge existiert und deren gerichtete Metrische Dimension nicht definiert ist. Dieses Problem kann gelöst werden, indem die Distanzfunktion wie oben beschrieben durch ∞ erweitert wird, oder indem die Metrische Dimension so definiert wird, dass nur Knotenpaare getrennt werden müssen, die keine Ankerknoten enthalten. Diese beiden Möglichkeiten sind nicht äquivalent. Die angesprochenen Probleme und Lösungsmöglichkeiten sind in Abbildung 8 zusammengefasst.

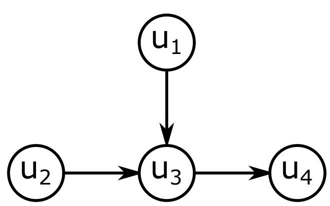


Abbildung 8: Ein Graph mit vier Knoten u_1, u_2, u_3, u_4. Wird die Distanzfunktion mit ∞ erweitert, so reicht der Knoten u_1 aus um alle Knotenpaare zu trennen, da $d(u_1, u_2) = \infty$, $d(u_1, u_1) = 0$, $d(u_1, u_3) = 1$ und $d(u_1, u_4) = 2$. Wenn ∞ nicht verwendet wird und alle Knoten außer den Ankerknoten getrennt werden müssen, so muss zusätzlich zu u_1 auch u_2 als Ankerknoten gewählt werden, um u_2 von u_3 und u_4 zu trennen, da ∞ und 1 bzw. ∞ und 2 nicht vergleichbar sind. Wenn alle Knotenpaare getrennt werden müssen, existiert für den abgebildeten Graphen keine trennende Menge, da u_1 und u_2 nicht getrennt werden können. In diesem Fall ist die Metrische Dimension des Graphen nicht definiert.

3.6.2 Definition der Gerichteten Metrischen Dimension

Für diese Arbeit ist $U \subseteq V(G)$ eine *trennende Menge* für einen gerichteten Graphen G, wenn gilt

$$\forall u, v \in V(G), u \neq v : \exists w \in U : d_G(w, u) \neq d_G(w, v).$$

Analog zur Metrischen Dimension im ungerichteten Fall lassen sich eine *metrische Basis* und die *gerichtete Metrische Dimension*, die mit gmd(G) bezeichnet wird, definieren. Das Entscheidungsproblem GERICHTETE METRISCHE DIMENSION ist wie folgt definiert.

GERICHTETE METRISCHE DIMENSION	
Gegeben:	Ein gerichteter Graph G und eine Zahl k.
Frage:	Ist gmd(G) $\leq$ k?

Da sich alle Ergebnisse zur ungerichteten Metrischen Dimension auch auf den gerichteten Fall übertragen lassen, wenn in den Graphen jede ungerichtete Kante durch zwei entgegengesetzte gerichtete ersetzt wird, folgt bereits die NP-Vollständigkeit der gerichteten Metrischen Dimension. Sie wurde zuerst in [23] betrachtet und die NP-Vollständigkeit der gerichteten Metrischen Dimension von orientierten Graphen wurde zuerst in [103] gezeigt. Das Problem ist sogar auf DAGs (gerichtete, kreisfreie Graphen) NP-vollständig, was für die Distanzfunktion mit ∞ in [27] gezeigt wurde. Der Fall, dass nicht die Distanz ∞ verwendet wird, aber die Ankerknoten nicht getrennt werden müssen, wird in Kapitel 5.1 betrachtet. Die gerichtete Metrische Dimension ist zudem FPT, wenn nach gerichteter modularer Weite parametrisiert wird [27].

Diese Variante wurde für eine Vielzahl von orientierten Graphen [4, 16, 23] betrachtet, insbesondere von orientierten Bäumen. Des Weiteren wurde die gerichtete Metrische Dimension von Linegraphen [41], von gerichteten unicyclic Graphen [27] und von gerichteten Cayley-Graphen [38, 98, 1] untersucht. Die zwei-Wege Distanz wurde in [40] betrachtete.

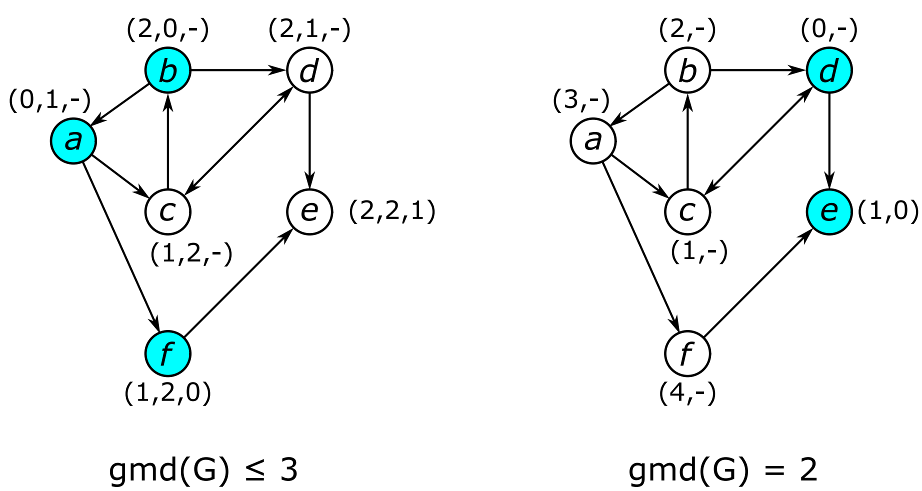


Abbildung 9: Ein gerichteter Graph G mit einer nicht verkleinerbaren trennenden Menge der Größe 3 (links) und einer metrischen Basis der Größe 2 (rechts).

3.7 Starke Metrische Dimension

Eine weitere bekannte Variante der Metrischen Dimension auf ungerichteten Graphen ist die Starke Metrische Dimension. Diese wird häufig getrennt von den anderen Varianten betrachtet, da sich die Art der Trennung der Knotenpaare stark unterscheidet.

Definition 14. *[99] Sei G ein ungerichteter Graph.*

- *Zwei Knoten $u, v \in V(G)$ werden von $w \in V(G)$* stark getrennt*, wenn es einen kürzesten Weg zwischen u und w gibt, der v enthält, oder wenn es einen kürzesten Weg zwischen v und w gibt, der u enthält.*
- *Eine Knotenmenge $U \subseteq V(G)$ heißt* stark trennende Menge*, wenn jedes Knotenpaar $u, v \in V(G)$ durch mindestens einen Knoten $w \in U$ stark getrennt wird. Eine kleinste stark trennende Menge wird auch* starke metrische Basis *genannt.*
- *Die* Starke Metrische Dimension *von G entspricht der Größe einer starken metrischen Basis für G und wird mit $\mathrm{smd}(G)$ bezeichnet.*

Da ein Knoten w zwei Knoten u, v auch (regulär) trennt, wenn er sie stark trennt, jedoch nicht umgekehrt, gilt $\mathrm{smd}(G) \geq \mathrm{md(G)}$. Ein Vorteil von stark trennenden Mengen liegt darin, dass mit ihrer Hilfe der Abstand zweier Knoten in einem Graphen genau bestimmt werden kann. Für eine gegebene trennende Menge $U = \{w_1, \ldots w_k\}$ für einen Graphen G entspricht der Abstand zwischen zwei Knoten $u, v \in V(G)$ der maximalen Differenz ihrer Distanzen zu einem Ankerknoten, also

$$d_G(u, v) \;=\; \max_{i=1}^{k} |d_G(u, w_i) - d_G(v, w_i)|.$$

Somit definieren die Distanzvektoren der Knoten zu den Knoten einer stark trennenden Menge die Kantenmenge des Graphen eindeutig, was optimales routen ermöglicht [113].

Das Entscheidungsproblem Starke Metrische Dimension ist wie folgt definiert.

Starke Metrische Dimension	
Gegeben:	Ein ungerichteter Graph G und eine Zahl k.
Frage:	Ist $\mathrm{smd}(G) \leq k$?

Die Starke Metrische Dimension wurde von Sebö und Tannier [113] eingeführt und ist für allgemeine Graphen NP-vollständig [99]. Das dazugehörige Optimierungsproblem kann aufgrund der Äquivalenz zu Vertex Cover (die in Kapitel 6 aufgegriffen wird) allerdings mit Faktor 2 approximiert werden, was auch in [29] gezeigt wurde. Die Starke Metrische Dimension lässt sich jedoch nicht bezüglich der Metrischen Dimension abschätzen [96].

Auch die Starke Metrische Dimension lässt sich auf einigen Graphklassen effizient lösen, beispielsweise auf Bäumen [113], Splitgraphen [96] und distanzerhaltenden Graphen [93]. Hauptsächlich wurde diese Variante jedoch für verschiedene Produkte von Graphen untersucht, beispielsweise das starke Produkt [83], das direkte Produkt, das kartesische Produkt [106], die kartesische Summe [87], das lexikographische Produkt [84] und das Coronaprodukt von Graphen, Joingraphen [89] und gewurzelte Produktgraphen [88].

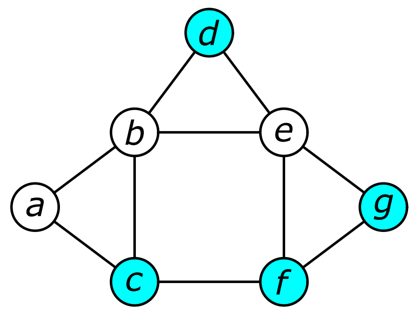


Abbildung 10: Der Graph aus Abbildung 1, mit einer stark trennenden Menge. Aus jedem der drei Knotenpaare $\{a, d\}$, $\{d, g\}$ und $\{a, g\}$ muss mindestens ein Knoten als Ankerknoten gewählt werden, da sich die kürzesten Wege zwischen diesen Knotenpaaren nicht erweitern lassen. Somit kann es keinen anderen Knoten geben, der diese Paare stark trennt. Auch alle drei Knoten a, d, g zusammen bilden keine stark trennende Menge für G, da zum Beispiel b und f nicht stark getrennt werden würden. Für diesen Graph gilt $\mathrm{smd}(G) = 4$.

3.8 Gerichtete Starke Metrische Dimension

Auch die Starke Metrische Dimension kann auf gerichtete Graphen übertragen werden, allerdings müssen hierbei beide Wege zwischen zwei Knoten berücksichtigt werden.

Definition 15. *Sei G ein stark zusammenhängender gerichteter Graph.*

- *Zwei Knoten $u, v \in V(G)$ werden durch $w \in V(G)$* stark von u nach v getrennt, *wenn es einen kürzesten Weg von w nach v gibt, der u enthält ($w \to u \to v$), oder wenn es einen kürzesten Weg von u nach w gibt, der v enthält ($u \to v \to w$).*

- *Zwei Knoten $u, v \in V(G)$ werden* stark getrennt, *wenn sie stark von u nach v und stark von v nach u getrennt werden.*

- *Eine Knotenmenge $U \subseteq V(G)$ heißt* stark trennende Menge, *wenn jedes Knotenpaar $u, v \in V(G)$ stark durch Knoten aus U getrennt wird. Eine kleinste stark trennende Menge wird auch* starke metrische Basis *genannt.*

- *Die Starke Gerichtete Metrische Dimension von G entspricht der Größe einer starken metrischen Basis für G und wird mit* gsmd(G) *bezeichnet.*

Das Entscheidungsproblem gerichtete Starke Metrische Dimension ist wie folgt definiert.

Gerichtete Starke Metrische Dimension	
Gegeben:	Ein gerichteter Graph G und eine Zahl k.
Frage:	Ist $\mathrm{gsmd(G)} \leq \mathrm{k}$?

Die Starke Gerichtete Metrische Dimension wurde ebenfalls von Sebö und Tannier [113] eingeführt und ist für allgemeine Graphen NP-vollständig [99]. Für diese Variante existiert ebenfalls eine Äquivalenz zu Vertex Cover [99] (siehe Kapitel 7) und auch im gerichteten Fall kann die Kantenmenge eines Graphen eindeutig mithilfe einer stark trennenden Menge bestimmt werden [99]. Darüber hinaus ist jedoch wenig über diese Variante bekannt.

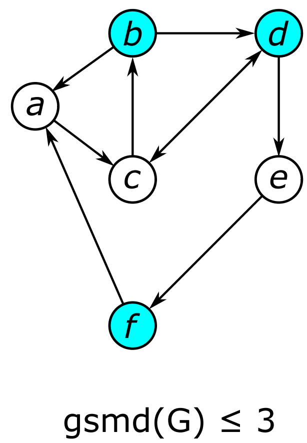


Abbildung 11: Ein stark zusammenhängender gerichteter Graph G mit einer stark trennenden Menge. Beispielsweise kann f nach b durch die Knoten d oder e getrennt werden, um jedoch b nach f zu trennen, muss einer der beiden Knoten selbst als Ankerknoten gewählt werden. Der Knoten d trennt außerdem e nach f, um jedoch f nach e zu trennen, muss auch hier einer der Knoten selbst gewählt werden, da es keinen längeren kürzesten Weg gibt, der den kürzesten Weg von f nach e enthält. Für diesen Graphen gilt gsmd(G) = 3.

3.9 Weitere Varianten

Neben den vorgestellten Varianten der Metrischen Dimension existieren zahlreiche weitere Varianten. Viele davon stellen zusätzliche Anforderungen an die trennende Menge, zum Beispiel dass sie auch eine unabhängige Menge [24, 119] oder eine dominierende Menge [18, 121] sein muss. Weitere Varianten finden sich beispielsweise in [19, 104, 105, 5, 6, 39, 33, 133, 51, 52, 81, 94, 95]. Zur gewichteten Metrischen Dimension [2, 3, 31] und ihrer Varianten wurden bereits einige Ergebnisse in den vorherigen Kapiteln vorgestellt.

Darüber hinaus lassen sich die verschiedenen Varianten fast beliebig kombinieren, wie zum Beispiel in [68, 69, 122, 120, 90, 132, 8, 15] geschehen.

Eine Übersicht über einige Varianten und Ergebnisse der Metrischen Dimension bieten beispielsweise [123, 58, 107, 86, 82].

4 Komplexität der k-Metrischen Dimension

In diesem Kapitel wird die NP-Vollständigkeit der k-Metrischen Dimension gezeigt. Dies erweitert den aktuellen Forschungsstand, da bisher lediglich ein Beweis für ungerade Werte von k bekannt ist [35].

Neben dem NP-Vollständigkeitsbeweis der Metrischen Dimension von Khuller et al. [74], der auf einer Reduktion von 3-SAT basiert, zählt der NP-Vollständigkeitsbeweis von Epstein et al. [32] zu den bekanntesten Beweisen. Dieser zeigt durch eine Reduktion von 3-dimensionalem Matching die NP-Vollständigkeit der Metrischen Dimension auch für einige bekannte Graphklassen (siehe Kapitel 3.1). Dieses Ausgangsproblem ist wie folgt definiert und bekanntermaßen NP-vollständig [46].

	3-dimensionales Matching (3DM)
Gegeben:	Drei disjunkte Grundmengen A, B, C mit jeweils n Elementen und eine Menge von Tripeln $S \subseteq A \times B \times C$.
Frage:	Existiert eine Auswahl der Tripel $M \subseteq S$ der Größe n, sodass jedes Element aus A, B und C in genau einem Tripel aus M enthalten ist?

Eine solche Lösung M, die jedes Element genau einmal überdeckt, wird auch Matching genannt. Basierend auf dem Beweis von Epstein et al. wird im Folgenden die NP-Vollständigkeit der k-Metrischen Dimension gezeigt. Dafür wird zunächst das oben genannte 3DM Problem erweitert, sodass jedes Element aus A, B und C in genau k Tripeln enthalten sein muss; es wird also ein k-Matching gesucht. Somit ergibt sich das folgende Problem.

	3-dimensionales k-Matching (3DkM)
Gegeben:	Drei disjunkte Grundmengen A, B, C mit jeweils n Elementen und eine Menge von Tripeln $S \subseteq A \times B \times C$.
Frage:	Existiert eine Auswahl der Tripel $M \subseteq S$ der Größe $k \cdot n$, sodass jedes Element aus A, B und C in genau k Tripeln aus M enthalten ist?

Offensichtlich entspricht dieses Problem für $k = 1$ dem bekannten 3DM Problem, sodass im Folgenden die NP-Vollständigkeit für $k \geq 2$ gezeigt wird. Zunächst sei jedoch erwähnt, dass für diesen Beweis nicht einfach eine 3DM-Instanz vervielfacht werden kann, wie das folgende Beispiel zeigt.

Betrachte die 3DM Instanz mit $A = \{a_1, a_2, a_3\}$, $B = \{b_1, b_2, b_3\}$, $C = \{c_1, c_2, c_3\}$ und $S = \{(a_1, b_1, c_1), (a_1, b_2, c_3), (a_2, b_1, c_2), (a_2, b_3, c_2), (a_3, b_2, c_1), (a_3, b_3, c_3)\}$. Offensichtlich existiert für diese Instanz keine Lösung. Da jedes Element in genau zwei Tripeln aus S enthalten ist, existiert jedoch für die 3D2M Instanz mit denselben Mengen (auch ohne verdoppeln der Tripel) eine Lösung.

An dieser Stelle folgen zwei Reduktionen von 3DM auf 3DkM, wobei die erste leichter verständlich ist, jedoch mit mehreren gleichen Tripeln arbeitet, wodurch die Menge S in der 3DkM Problemdefinition als Multimenge definiert werden müsste. Im zweiten Beweis geht es hauptsächlich um eine Konstruktion, bei der keine gleichen Tripel auftreten; die Grundidee der Reduktion bleibt dieselbe wie im ersten Beweis.

Theorem 1. 3-DIMENSIONALES k-MATCHING *ist NP-vollständig für jedes* $k \geq 2$.

Beweis. Das Problem 3DkM ist offensichtlich in NP, da nichtdeterministisch eine Auswahl der Tripel getroffen werden kann und für diese in polynomieller Zeit überprüft werden kann, ob jedes Element aus A, B und C in genau k Tripeln enthalten ist. Für die NP-Schwere wird eine Reduktion von 3DM verwendet. Sei

$$\begin{aligned} A &= \{a_1, \ldots, a_n\}, & B &= \{b_1, \ldots, b_n\}, \\ C &= \{c_1, \ldots, c_n\}, \text{ und } & S &= \{s_1, \ldots, s_m\} \end{aligned}$$

eine 3DM-Instanz. Definiere eine 3DkM-Instanz mit

$$\begin{aligned} A' &= A \cup \{a_{n+1}, \ldots, a_{3n}\}, & B' &= B \cup \{b_{n+1}, \ldots, b_{3n}\}, \\ C' &= C \cup \{c_{n+1}, \ldots, c_{3n}\}, \text{ und } & S' &= S \cup R \cup T \end{aligned}$$

mit $R = \{(a_i, b_i, c_i) \mid n+1 \leq i \leq 3n\}$. Die Menge T enthält für jedes $i \in \{1, \ldots, n\}$ genau $(k-1)$ Kopien der Tripel $(a_i, b_{i+n}, c_{i+n}), (a_{i+n}, b_i, c_{i+2n}), (a_{i+2n}, b_{i+2n}, c_i)$. Somit enthalten die Tripel in T jedes Element aus A', B' und C' genau $(k-1)$-mal und die Tripel in R jedes Element aus $A' \setminus A, B' \setminus B$ und $C' \setminus C$ genau einmal. Da keine weiteren Tripel hinzukommen, muss jedes k-Matching für die 3DkM-Instanz alle Tripel aus R und T enthalten, um die Elemente, die neu hinzukommen, also nicht Teil der 3DM-Instanz sind, zu überdecken. Diese Tripel enthalten jedes neue Element genau k-mal und alle Elemente der ursprünglichen 3DM-Instanz genau $(k-1)$-mal. Folglich entsprechen die Tripel eines k-Matchings, die nicht Teil von R und T sind einem Matching für die 3DM-Instanz, da diese Tripel auch alle Elemente aus S sind. Umgekehrt entsteht aus einem Matching für die 3DM-Instanz durch Hinzunahme der Tripel aus R und T ein k-Matching für die 3DkM-Instanz. Die Reduktion kann offensichtlich in polynomieller Zeit durchgeführt werden. □

Theorem 2. 3-DIMENSIONALES k-MATCHING *ist NP-vollständig für jedes* $k \geq 2$.

Beweis. Wie oben angesprochen liegt der Fokus bei diesem Beweis auf der Konstruktion der Tripelmenge. Sei erneut

$$\begin{aligned} A &= \{a_1, \ldots, a_n\}, & B &= \{b_1, \ldots, b_n\}, \\ C &= \{c_1, \ldots, c_n\}, \text{ und } & S &= \{s_1, \ldots, s_m\} \end{aligned}$$

eine 3DM-Instanz. Definiere erneut eine 3DkM-Instanz mit

$$\begin{aligned} A' &= A \cup \{a_{n+1}, \ldots, a_{3n}\}, & B' &= B \cup \{b_{n+1}, \ldots, b_{3n}\}, \\ C' &= C \cup \{c_{n+1}, \ldots, c_{3n}\}, \text{ und } & S' &= S \cup R \cup T \end{aligned}$$

mit $R = \{(a_i, b_i, c_i) \,|\, n+1 \leq i \leq 3n\}$. Lediglich die Menge T wird in diesem Beweis angepasst, überdeckt jedoch weiterhin alle Elemente genau $(k-1)$-mal. Somit gilt dieselbe Argumentation wie im vorherigen Beweis. Damit jedes Element genau $(k-1)$-mal überdeckt wird, muss T genau $3n(k-1)$ Tripel enthalten, die mithilfe der Menge

$$T_{p,q} \subseteq (A \times B) \cup (A \times C) \cup (B \times C)$$

definiert werden. Hierbei sei

$$\begin{aligned} T_{p,q} = \;& \{(a_i, b_j) \,|\, i \in \{p, \ldots, p+q-1\},\, j \in \{p+q, \ldots, p+2q-1\}\} \\ \cup\;& \{(b_i, c_j) \,|\, i \in \{p, \ldots, p+q-1\},\, j \in \{p+q, \ldots, p+2q-1\}\} \\ \cup\;& \{(c_i, a_j) \,|\, i \in \{p, \ldots, p+q-1\},\, j \in \{p+q, \ldots, p+2q-1\}\}. \end{aligned}$$

Diese $3q^2$ Tupel überdecken jedes Element aus

$$\{a_p, \ldots, a_{p+2q-1},\, b_p, \ldots, b_{p+2q-1},\, c_p, \ldots, c_{p+2q-1}\}$$

genau q-mal. Um alle neuen Elemente $(k-1)$-mal zu überdecken sei nun

$$T' = \bigcup_{i=0}^{r-1} T_{n+1+i2(k-1),\, k-1}, \quad \text{mit } r = \frac{n}{k-1}.$$

Die Menge T' enthält $r3(k-1)^2 = \frac{n}{k-1} \cdot 3(k-1)^2 = 3n(k-1)$ Tupel, die nur die neuen Elemente enthalten. Im letzten Schritt werden die $3n(k-1)$ Tupel von T' zu $3n(k-1)$ Tripeln für T erweitert, indem jedes Element aus $A \cup B \cup C$ zu genau $k-1$ Tupeln von T' hinzugefügt wird, sodass Tripel der Form $A' \times B' \times C'$ entstehen. Wie aus einer Lösung für die 3DM-Instanz eine Lösung für die 3DkM-Instanz konstruiert werden kann und vice versa ist analog zum vorherigen Beweis. Die Reduktion kann offensichtlich in polynomieller Zeit durchgeführt werden. $\square$

Theorem 3. k-Metrische Dimension *ist NP-vollständig für jedes $k \geq 2$.*

Beweis. Das Problem k-MD ist offensichtlich in NP, da nichtdeterministisch eine Auswahl der Knoten getroffen werden kann und für diese in polynomieller Zeit überprüft werden kann, ob sie eine k-trennende Menge ist. Für die NP-Schwere wird eine Reduktion von 3D$(k-1)$M verwendet. Sei $A = \{a_1, \ldots, a_n\}, B = \{b_1, \ldots, b_n\}, C = \{c_1, \ldots, c_n\}$ und $S = \{s_1, \ldots, s_m\}$ eine 3D$(k-1)$M-Instanz I mit $k \geq 2$. Ohne Beschränkung der Allgemeinheit sei $n > k$. Zudem wird davon ausgegangen, dass jedes Element in mindestens einem Tripel enthalten ist. Für diese Instanz wird ein Graph G und eine Zahl x definiert, sodass G genau dann eine k-trennende Menge der Größe x besitzt, wenn ein $(k-1)$-Matching für I existiert.

Der Graph G wird wie folgt definiert, siehe auch Abbildung 12. G enthält die Knoten a_i, b_i und c_i für $1 \leq i \leq n$ und die Knoten s_i für $1 \leq i \leq m$. Diese Knoten entsprechen jeweils einem Element in A, B, C und S. Die Knoten und die Elemente werden synonym verwendet. Zusätzlich enthält G die Knoten $a_0, b_0, c_0, v_0, v_A, v_B, v_C$ und $d_1, \ldots, d_{m'}$

mit $m' = \lceil log(m) \rceil$. Die d-Knoten werden für eine binär-Darstellung der Indizes der Tripel aus S verwendet. Da die Indizes alle unterschiedlich sind, unterscheiden sich ihre binär-Darstellungen in mindestens einer Position, was die Trennung der entsprechenden Knoten ermöglicht. Die bisherigen Knoten werden wie folgt durch Kanten verbunden.

1. Jeder Knoten a_i, $0 \leq i \leq n$ wird verbunden mit
   (a) Knoten v_A,
   (b) Knoten v_0, und
   (c) Knoten s_j, $1 \leq j \leq m$ genau dann, wenn das Tripel s_j das Element a_i enthält.
2. Jeder Knoten b_i, $0 \leq i \leq n$ wird verbunden mit
   (a) Knoten v_B,
   (b) Knoten v_0, und
   (c) Knoten s_j, $1 \leq j \leq m$ genau dann, wenn das Tripel s_j das Element b_i enthält.
3. Jeder Knoten c_i, $0 \leq i \leq n$ wird verbunden mit
   (a) Knoten v_C,
   (b) Knoten v_0, und
   (c) Knoten s_j, $1 \leq j \leq m$ genau dann, wenn das Tripel s_j das Element c_i enthält.
4. Jeder Knoten d_i, $1 \leq i \leq m'$ wird verbunden mit
   (a) Knoten v_0 und
   (b) Knoten s_j, $1 \leq j \leq m$ genau dann, wenn das i-te Bit in der binär-Darstellung von j eine 1 ist.

Der Graph G enthält außerdem sogenannte *leg-Knoten*, die Wege (*legs*) mit $\lceil k/2 \rceil$ und $\lfloor k/2 \rfloor$ Knoten bilden. Jeweils ein Weg mit $\lceil k/2 \rceil$ und ein Weg mit $\lfloor k/2 \rfloor$ wird an die Knoten $L_{\text{root}} = \{v_A, v_B, v_C, v_0, d_1, \ldots, d_{m'}\}$ angefügt, siehe Abbildung 12. Die Knoten aus L_{root} sind die Wurzeln der Wege. Seien L_v die Knoten der beiden Wege mit Wurzelknoten v und sei L die Menge aller Knoten in den Wegen. Jede Menge L_v enthält genau k Knoten für jedes $v \in L_{\text{root}}$ und L enthält genau $(4 + m')k$ Knoten.

Der Graph G kann für die Instanz I in polynomieller Zeit konstruiert werden und besitzt die folgenden Eigenschaften.

E1: Die Distanz zwischen
   (a) zwei Knoten aus $\{v_B, v_B, v_C\}$ ist 4,
   (b) zwei Knoten aus $\{d_1, \ldots, d_{m'}\}$ ist 2,
   (c) einem Knoten aus $\{v_B, v_B, v_C\}$ und einem Knoten aus $\{d_1, \ldots, d_{m'}\}$ ist 3,

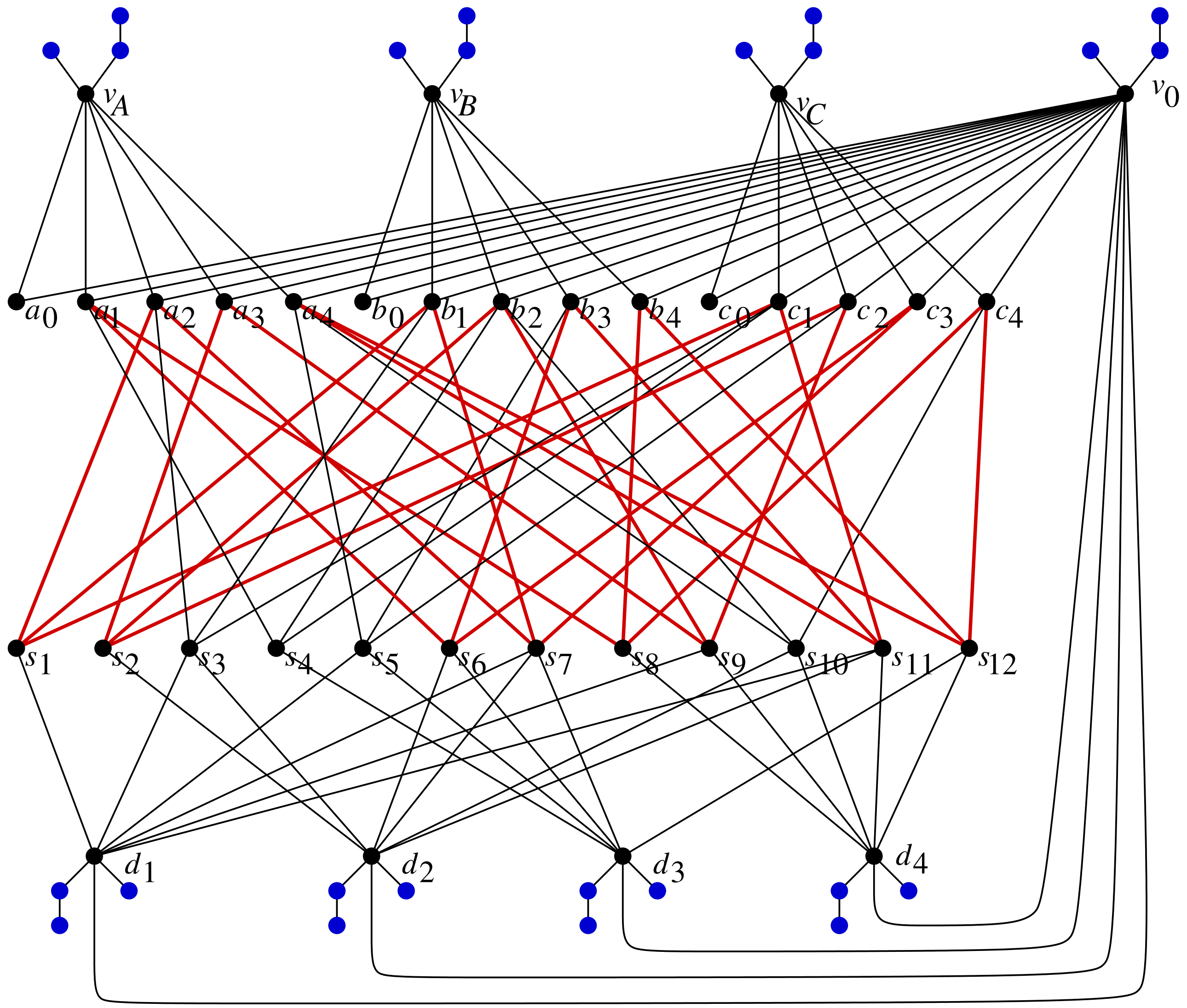


Abbildung 12: Der Graph der bei der Reduktion von 3D2M auf 3-MD entsteht aus der Instanz $A = \{a_1, \dots, a_4\}$, $B = \{b_1, \dots, b_4\}$, $C = \{c_1, \dots, c_4\}$, $S = \{s_1, \dots, s_{12}\}$ mit $s_1 = (a_2, b_1, c_1)$, $s_2 = (a_3, b_2, c_2)$, $s_3 = (a_2, b_1, c_1)$, $s_4 = (a_1, b_2, c_1)$, $s_5 = (a_4, b_3, c_2)$, $s_6 = (a_1, b_3, c_3)$, $s_7 = (a_2, b_1, c_3)$, $s_8 = (a_1, b_4, c_4)$, $s_9 = (a_3, b_2, c_2)$, $s_{10} = (a_4, b_2, c_4)$, $s_{11} = (a_4, b_3, c_1)$, $s_{12} = (a_4, b_4, c_4)$. Die Tripel $M = \{s_1, s_2, s_6, s_7, s_8, s_9, s_{11}, s_{12}\}$ bilden ein 2-Matching für die 3D2M Instanz und die Knoten $L \cup \{a_0, b_0, c_0\} \cup M$ bilden eine 3-fach trennende Menge für die 3-MD Instanz. Die Metrische Dimension des Graphen ist $(4 + 4)3 + 3 + (3 - 1)n = 35$. Die Tripel des Matchings sind in rot dargestellt und die Knoten aus L sind blau gefärbt.

(d) dem Knoten v_0 und einem Knoten aus $\{v_B, v_B, v_C\}$ ist 2, und

(e) dem Knoten v_0 und einem Knoten aus $\{d_1, \dots, d_{m'}\}$ ist 1.

E2: Jede k-trennende Menge für G enthält alle Knoten aus L. Dies liegt daran, dass für jeden Knoten $v \in L_{\text{root}}$ die beiden Knoten aus L_v, die adjazent zu v sind, nur durch die Knoten aus L_v getrennt werden können.

Für I existiert genau dann ein $(k-1)$-Matching in S, wenn G eine k-trennende Menge der Größe

$$x = (4 + m')k + 3 + (k-1)n$$

besitzt.

„$\Rightarrow$:“ Sei $M \subseteq S$ ein $(k-1)$-Matching für I. Dann ist

$$U \;=\; L \,\cup\, \{a_0, b_0, c_0\} \,\cup\, M$$

eine k-trennende Menge der geforderten Größe für G.

Zwei Knoten $u_1, u_2 \in V(G)$ werden wie folgt durch jeweils mindestens k Knoten aus U getrennt.

1. Seien $u_1, u_2 \in L_v$, $v \in L_{\text{root}}$.
   (a) Wenn $d(u_1, v) = d(u_2, v)$, dann trennt jeder der k Knoten aus L_v die Knoten u_1 und u_2.
   (b) Wenn $d(u_1, v) \neq d(u_2, v)$, dann trennt jeder der k Knoten aus $L_{v'}$ für ein $v' \in L_{\text{root}} \setminus \{v\}$ die Knoten u_1 und u_2.
2. Wenn $u_1 \in L_{v_1}$, $u_2 \in L_{v_2}$, $v_1, v_2 \in L_{\text{root}}$, $v_1 \neq v_2$, und o.B.d.A. $d(u_1, v_1) \leq d(u_2, v_2)$, dann trennt jeder der k Knoten aus L_{v_1} die Knoten u_1 und u_2. Die beiden Knoten u_1, u_2 können nicht denselben Abstand zu v_1 und gleichzeitig denselben Abstand zu v_2 haben, da $d_G(v_1, v_2) > 0$.
3. Wenn $u_1 \in L_{v_A} \cup L_{v_B} \cup L_{v_C}$ und $u_2 \notin L$, dann trennt jeder der k Knoten aus L_{v_0} die Knoten u_1 und u_2, da $d_G(u_1, v_0) \geq 3$ und $d_G(u_2, v_0) \leq 2$.
4. Sei $u_1 \in L_{d_1} \cup \cdots \cup L_{d_{m'}}$ und $u_2 \notin L$.
   (a) Wenn $u_2 \notin \{v_B, v_C\}$, dann trennt jeder der k Knoten aus L_{v_A} die Knoten u_1 und u_2, da $d_G(u_1, v_A) \geq 4$ und $d_G(u_2, v_A) \leq 3$.
   (b) Wenn $u_2 \notin \{v_A, v_C\}$, dann trennt jeder der k Knoten aus L_{v_B} die Knoten u_1 und u_2, da $d_G(u_1, v_B) \geq 4$ und $d_G(u_2, v_B) \leq 3$.
   (c) Wenn $u_2 \notin \{v_A, v_B\}$, dann trennt jeder der k Knoten aus L_{v_C} die Knoten u_1 und u_2, da $d_G(u_1, v_C) \geq 4$ und $d_G(u_2, v_C) \leq 3$.
5. Sei $u_1 \in L_{v_0}$ und $u_2 \notin L$.
   (a) Wenn $u_2 \in \{v_A, a_0, \ldots, a_n\}$, dann trennt jeder der k Knoten aus L_{v_A} die Knoten u_1 und u_2.
   (b) Wenn $u_2 \in \{v_B, b_0, \ldots, b_n\}$, dann trennt jeder der k Knoten aus L_{v_B} die Knoten u_1 und u_2.

(c) Wenn $u_2 \in \{v_C, c_0, \dots, c_n\}$, dann trennt jeder der k Knoten aus L_{v_C} die Knoten u_1 und u_2.

(d) Wenn $u_2 \in \{d_i\} \cup \{s_j \,|\, \text{das } i\text{-te Bit in der binär-Darstellung von } j \text{ ist eine } 1\}$, dann trennt jeder der k Knoten aus L_{d_i} die Knoten u_1 und u_2. Da die Indizes mit 1 beginnen, enthält jede binär-Darstellung mindestens eine 1, wodurch alle s-Knoten betrachtet werden.

6. Wenn $u_1 \in L_{\text{root}}$ und $u_2 \notin L$, dann trennt jeder der k Knoten aus L_{u_1} die Knoten u_1 und u_2.

7. Sei $u_1 = s_{i_1} \in \{s_1, \dots, s_{m'}\}$ und $u_2 \notin L \cup L_{\text{root}}$.

   (a) Wenn $u_2 = s_{i_2} \in \{s_1, \dots, s_{m'}\}$ mit $i_1 \neq i_2$, dann trennt jeder der k Knoten aus L_{d_j} die Knoten u_1 und u_2, wenn sich die binär-Darstellungen von i_1 und i_2 in Position j unterscheiden.

   (b) Wenn $u_2 \in \{a_0, \dots, a_n\}$, $u_2 \in \{b_0, \dots, b_n\}$ oder $u_2 \in \{c_0, \dots, c_n\}$, dann trennt jeder der k Knoten aus L_{v_A}, L_{v_B} beziehungsweise L_{v_C} die Knoten u_1 und u_2.

8. Sei $u_1 \in \{a_1, \dots, a_n\}$ und $u_2 \notin L \cup L_{\text{root}} \cup \{s_1, \dots, s_{m'}\}$.

   (a) Wenn $u_2 \in \{b_0, \dots, b_n\} \cup \{c_0, \dots, c_n\}$, dann trennt jeder der k Knoten aus L_{v_A} die Knoten u_1 und u_2.

   (b) Wenn $u_2 \in \{a_1, \dots, a_n\}$ und $u_1 \neq u_2$, dann trennt jeder Knoten s_i aus M für den das Tripel s_i entweder das Element u_1 oder das Element u_2 enthält die Knoten u_1 und u_2. Da jedes Element in genau $k-1$ Tripeln aus M enthalten ist, existieren insgesamt $2(k-1) \geq k$ solcher Knoten für $k \geq 2$.

   (c) Wenn $u_2 = a_0$, dann trennt jeder Knoten s_i aus M für den das Tripel s_i das Element u_1 enthält die Knoten u_1 und u_2. Zusätzlich trennt a_0 die Knoten u_1 und u_2. Da jedes Element in genau $k-1$ Tripeln aus M enthalten ist, sind dies genau $(k-1)+1 = k$ Knoten.

9. Wenn $u_1 \in \{b_1, \dots, b_n\}$ und $u_2 \notin L \cup L_{\text{root}} \cup \{s_1, \dots, s_{m'}\}$, dann werden die Knoten u_1 und u_2 analog zu Fall 8 von mindestens k Knoten getrennt.

10. Wenn $u_1 \in \{c_1, \dots, c_n\}$ und $u_2 \notin L \cup L_{\text{root}} \cup \{s_1, \dots, s_{m'}\}$, dann werden die Knoten u_1 und u_2 analog zu Fall 8 von mindestens k Knoten getrennt.

11. Wenn $u_1, u_2 \in \{a_0, b_0, c_0\}$, dann trennt jeder der k Knoten aus L_{v_A}, L_{v_B} oder L_{v_C} die Knoten u_1 und u_2.

„$\Leftarrow$:“ Sei $U \subseteq V(G)$ eine k trennende Menge der Größe $(4+m')k + 3 + (k-1)n$ für G. Aufgrund der oben beschriebenen Eigenschaft E2 enthält U alle $(4+m')k$ Knoten aus L. Aus den Fallunterscheidungen der vorherigen Beweisrichtung geht hervor, dass durch diese Knoten bereits die meisten Knotenpaare in G k-fach getrennt werden. Die einzigen

Knotenpaare die noch zu betrachten sind, sind die, bei denen beide Knoten aus $\{a_0, \dots, a_n\}$, $\{b_0, \dots, b_n\}$ oder $\{c_0, \dots, c_n\}$ stammen. Betrachte die Knotenpaare a_0, a_i, b_0, b_i und c_0, c_i für $1 \leq i \leq n$. Diese Paare können nicht durch die Knoten aus $L \cup \{v_A, v_B, v_C, v_0, d_1, \dots, d_{m'}\}$ getrennt werden. Die einzige Möglichkeit diese $3n$ Knotenpaare mit maximal $3 + (k-1)n$ Knoten jeweils mindestens k-fach zu trennen besteht darin, $k-1$ der s-Knoten zu wählen, die ein $(k-1)$-Matching bilden sowie die drei Knoten a_0, b_0, c_0. An dieser Stelle ist es relevant, dass n größer als k ist. Die Knoten a_i, b_i und c_i für $1 \leq i \leq n$ kommen für die Trennung nicht in Frage, da sie nur den jeweiligen Knoten selbst von ihrem entsprechenden 0-Knoten trennen und somit insgesamt mehr als $3+(k-1)n$ Knoten benötigt würden, um alle Paare k-fach zu trennen. □

5 Komplexität der gerichteten Metrischen Dimension

5.1 Komplexität der gerichteten Metrischen Dimension für DAGs

Wie in Kapitel 3.6.2 angesprochen, wird in diesem Kapitel die NP-Vollständigkeit der gerichteten Metrischen Dimension für DAGs gezeigt. Es werden die Distanzen von den Ankerknoten zu den anderen Knoten verwendet und die Distanzen ohne ∞. Damit die gerichtete Metrische Dimension für jeden DAG definiert ist, müssen Ankerknoten nicht von anderen Knoten getrennt werden.

Theorem 4. Gerichtete Metrische Dimension *ist NP-vollständig für DAGs.*

Beweis. Das Problem gerichtete Metrische Dimension ist offensichtlich in NP, da nichtdeterministisch eine Auswahl von Ankerknoten getroffen werden kann und für diese in polynomieller Zeit überprüft werden kann, ob sie eine trennende Menge ist. Für die NP-Schwere wird eine Reduktion von Hitting Set verwendet. Dieses Problem ist wie folgt definiert und bekanntermaßen NP-vollständig [71].

Hitting Set	
Gegeben:	Eine Menge von Teilmengen S einer Grundmenge X und eine Zahl k.
Frage:	Gibt es eine Auswahl von k Elementen aus U, sodass jede Teilmenge in S mindestens eines dieser k Elemente enthält?

Sei also $X = \{X_1, \ldots, X_n\}$ und $S = \{S_1, \ldots, S_m\}$ eine Instanz I für Hitting Set. Für diese Instanz wird ein DAG G konstruiert, der genau dann eine trennende Menge der Größe $3+k$ besitzt, wenn eine Menge $X' \subseteq X$ der Größe k existiert, sodass $X' \cap S_j \neq \emptyset$ für $1 \leq j \leq m$. O.B.d.A. wird angenommen, dass $m > n$ und dass für jedes $S_j \in S$ mindestens ein Element $X_i \in X$ existiert, sodass $X_i \notin S_j$.

Der Graph G für die Instanz I enthält

1. drei Knoten v_a, v_b, v_c,
2. n Knoten x_i für $1 \leq i \leq n$, einen für jedes Element aus X und
3. $2m$ Knoten s_j, s'_j, zwei für jede Teilmenge in S.

Außerdem enthält G

1. n Kanten (v_a, x_i) für $1 \leq i \leq n$,
2. eine Kante (v_b, x_1),
3. $n-1$ Kanten (x_i, x_{i+1}) für $1 \leq i \leq n-1$,
4. zwei Kanten $(v_c, s_1), (v_c, s'_1)$,

5. m Kanten (s_j, s'_j) für $1 \leq j \leq m$,

6. $4(m-1)$ Kanten $(s_j, s_{j+1}), (s_j, s'_{j+1}), (s'_j, s_{j+1}), (s'_j, s'_{j+1})$ für $1 \leq j \leq m-1$,

7. nm Kanten (x_i, s_j) für $1 \leq i \leq n$, $1 \leq j \leq m$ und

8. eine Kante (x_i, s'_j) für $1 \leq i \leq n$, $1 \leq j \leq m$ genau dann wenn $X_i \notin S_j$.

Für ein Beispiel siehe Abbildung 13.

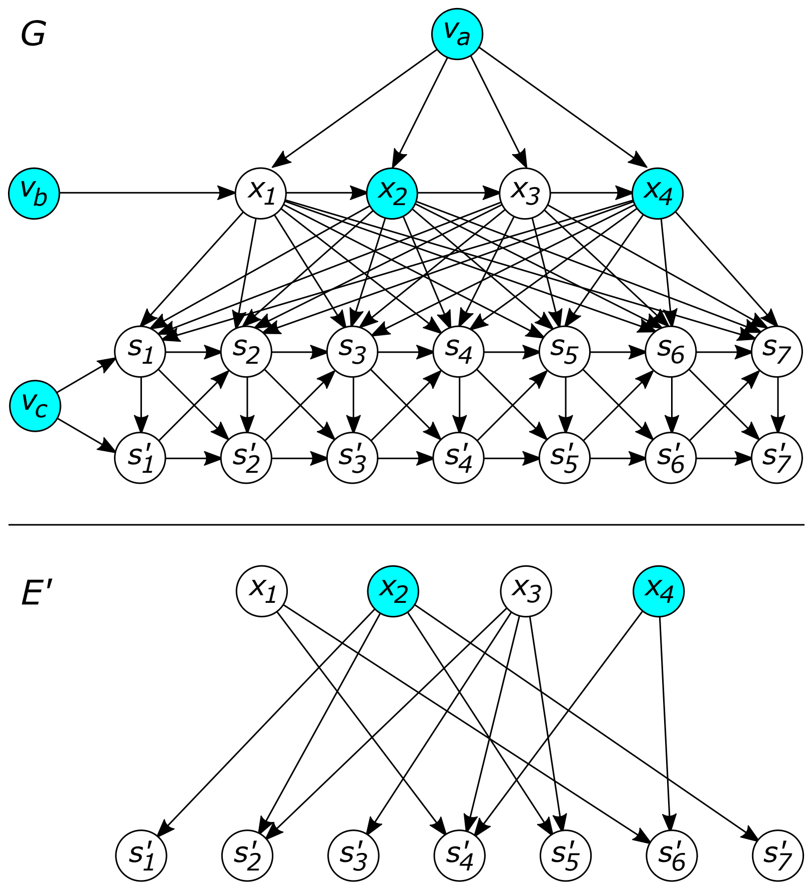


Abbildung 13: Der Graph G, der aus der Hitting Set Instanz $X = \{x_1, x_2, x_3, x_4\}$ und $S = \{S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$ mit $S_1 = \{x_1, x_3, x_4\}$, $S_2 = \{x_1, x_4\}$, $S_3 = \{x_1, x_2, x_4\}$, $S_4 = \{x_2\}$, $S_5 = \{x_1, x_4\}$, $S_6 = \{x_2, x_3\}$ und $S_7 = \{x_1, x_3, x_4\}$ entsteht. Die Kanten zwischen den x- und den s'-Knoten sind für eine bessere Übersicht separat gezeichnet. Die Menge $\{x_2, x_4\}$ ist eine minimale Lösung für die Hitting Set Instanz und die Menge $\{v_a, v_b, v_c, x_2, x_4\}$ ist eine minimale trennende Menge für G.

Jede minimale trennende Menge für G enthält die drei Knoten v_a, v_b, v_c, da diese keine einlaufenden Kanten besitzen und somit alle anderen Knoten Ankerknoten sein müssten, wenn es einer dieser drei Knoten nicht ist. Der Knoten v_b trennt alle Knotenpaare (x_i, x_j), $1 \leq i < j \leq n$, da $d_G(v_b, x_i) = i$. Der Knoten v_c trennt alle Knoten, die für unterschiedliche

Teilmengen stehen, also die Paare $(s_i, s_j), (s_i, s'_j), (s'_i, s_j), (s'_i, s'_j)$, $1 \leq i < j \leq m$, da $d_G(v_c, s_i) = d_G(v_c, s'_i) = i$. Der Knoten v_a trennt alle Paare $(x_i, s_j), (x_i, s'_j)$, da $d_G(v_a, x_i) = 1$, $1 \leq i \leq n$ und $d_G(v_a, s_j) = d_G(v_a, s'_j) = 2$, $1 \leq j \leq m$. Die letzte Distanz folgt daraus, dass für jedes S_j ein X_i existiert, sodass $X_i \notin S_j$.

Die einzigen Knotenpaare, die noch zu trennen sind, sind die Paare (s_j, s'_j) für $1 \leq j \leq m$. Diese Knotenpaare können nur durch die beiden Knoten selbst oder durch einen Knoten x_i mit $X_i \in S_j$ getrennt werden, da dann $d_G(x_i, s_j) = 1$ und $d_G(x_i, s'_j) = 2$.

Sei $X' \subseteq X$ ein hitting set der Größe k für die Instanz I, das heißt $X' \cap S_j \neq \emptyset$ für $1 \leq j \leq m$. Dann ist

$$U = \{v_a, v_b, v_c\} \cup \bigcup_{X_i \in X'} x_i$$

eine trennende Menge der Größe $3 + k$ für G.

Sei U eine trennende Menge der Größe $3 + k$ für G. Wenn U einen Knoten s_j oder s'_j enthält, ersetze ihn durch einen Knoten x_i mit $X_i \in S_j$. Für die resultierende trennende Menge U' ist

$$X' = \{x_i \mid x_i \in U'\}$$

ein hitting set der Größe k für I.

Die Reduktion kann offensichtlich in polynomieller Zeit durchgeführt werden.

□

5.2 Definition ungerichteter Co-Graphen

In dieser Arbeit wird die Metrische Dimension von gerichteten und ungerichteten Co-Graphen betrachtet. Co-Graphen sind sowohl für Probleme auf ungerichteten als auch auf gerichteten Graphen eine häufig untersuchte Graphklasse, da sie durch leicht verständliche Konstruktionsregeln definiert sind und viele hilfreiche Eigenschaften besitzen. Ungerichtete Co-Graphen und ihre gerichtete Version werden im weiteren Verlauf dieser Arbeit sowohl in Hinblick auf die gerichtete (Kapitel 5.3), die Starke (Kapitel 6), als auch die gerichtete Starke Metrische Dimension (Kapitel 7) untersucht. Insbesondere für die gerichtete Metrische Dimension sind viele Erkenntnisse aus dem ungerichteten Fall übertragbar, sodass ungerichtete Co-Graphen an dieser Stelle ausführlicher behandelt werden.

Ungerichtete Co-Graphen sind wie folgt definiert.

Definition 16.

- *Der Graph mit nur einem Knoten ist ein ungerichteter Co-Graph.*
- *Seien G_1 und G_2 zwei ungerichtete Co-Graphen. Die* Vereinigung $G = G_1 \cup G_2$ *dieser Graphen ist ein ungerichteter Co-Graph mit Knotenmenge $V(G_1) \cup V(G_2)$ und Kantenmenge $E(G_1) \cup E(G_2)$.*
- *Seien G_1 und G_2 zwei ungerichtete Co-Graphen. Der* Join $G = G_1 \times G_2$ *dieser Graphen ist ein ungerichteter Co-Graph mit Knotenmenge $V(G_1) \cup V(G_2)$ und Kantenmenge $E(G_1) \cup E(G_2) \cup \{\{u, v\} | u \in V(G_1), v \in V(G_2)\}$.*

Für ungerichtete Co-Graphen existiert eine äquivalente Definition, in der der Join zweier Co-Graphen durch das Komplementieren eines Co-Graphen ersetzt wird. Diese Definition ist auch namensgebend für Co-Graphen, deren Name eine Abkürzung des englischen Originals „complement-reducible graphs“ ist. Die Äquivalenz der beiden Definitionen ist durch die beiden Gleichungen $G_1 \times G_2 = \overline{\overline{G_1} \cup \overline{G_2}}$ und $\overline{G_1 \cup G_2} = \overline{G_1} \times \overline{G_2}$ ersichtlich. Für ein Beispiel siehe Abbildung 14.

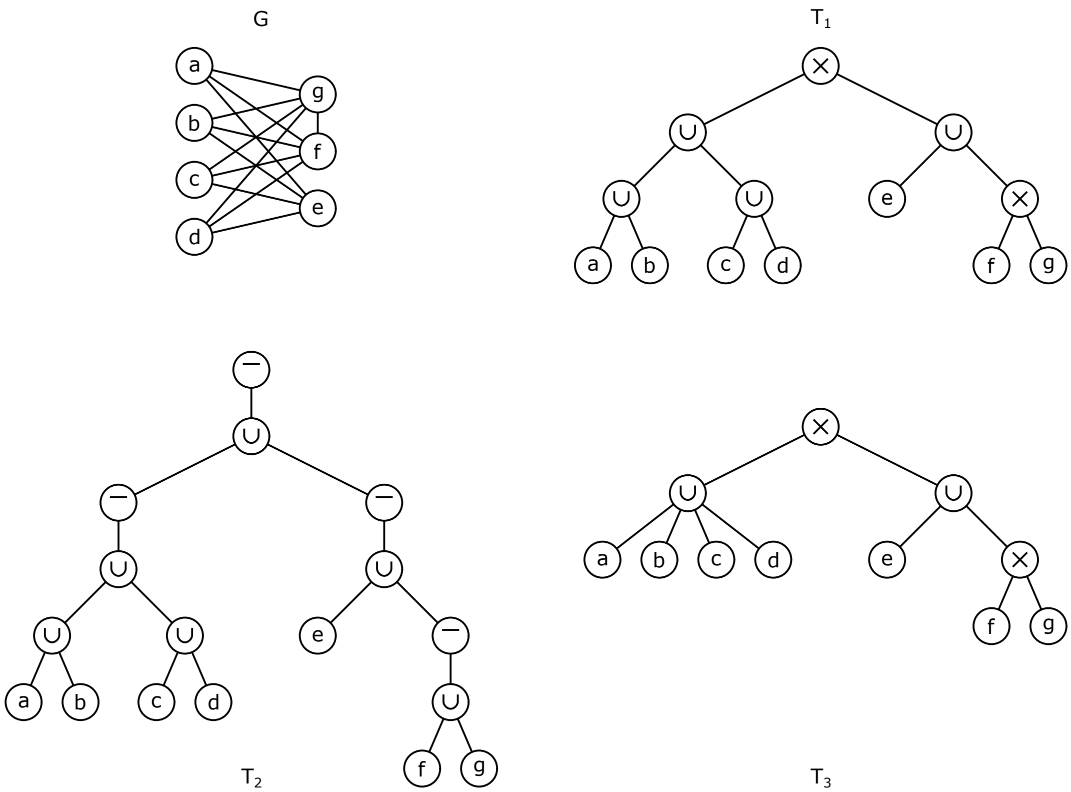


Abbildung 14: Ein Co-Graph G (oben links), ein Co-Baum T_1 von G (oben rechts) mit den Operationen Vereinigung und Join, ein Co-Baum T_2 von G (unten links) mit den Operationen Vereinigung und Komplement und der kanonische Co-Baum T_3 von G. Die Knoten a, b, c, d können in beliebiger Reihenfolge vereinigt werden, wodurch die beiden nicht kanonischen Co-Bäume nicht eindeutig sind.

Für ungerichtete Co-Graphen wurden bereits verschiedene Varianten der Metrischen Dimension untersucht (vergleiche Kapitel 3).

Besteht ein ungerichteter Co-Graph G aus der Vereinigung oder dem Join zweier ungerichteter Co-Graphen G_1 und G_2, so besitzen alle Knoten aus G_1 dieselbe Distanz zu allen Knoten aus G_2. Demnach kann kein Knoten aus G_1 zwei Knoten aus G_2 trennen und vice versa. Somit können für die Berechnung einer trennenden Menge für G trennende Mengen für G_1 und G_2 vereinigt werden, falls diese bereits bestimmt wurden und durch weitere Knoten ergänzt werden, um mit letzteren gegebenenfalls Knoten aus G_1 von Knoten aus G_2 zu trennen. Dieser Ansatz des Zusammensetzens und Ergänzens bereits vorhandener

Teillösungen zu einer Gesamtlösung ist der Grundgedanke der dynamischen Programmierung und kommt bei allen Varianten der Metrischen Dimension für Co-Graphen zum Einsatz. Insbesondere lässt sich zu jedem ungerichteten Co-Graph ein sogenannter Co-Baum berechnen, der den rekursiven Aufbau des Graphen widerspiegelt und dadurch die ideale Grundlage für einen solchen dynamischen Ansatz bietet. Der Co-Baum eines ungerichteten Co-Graph ist wie folgt definiert.

Definition 17. *[25]*

- *Sei G ein ungerichteter Co-Graph mit nur einem Knoten v. Der Co-Baum T von G besteht aus einem einzelnen Blatt, das mit v assoziiert ist. Dieses Blatt ist die Wurzel von T.*

- *Sei $G = G_1 \cup G_2$ ein ungerichteter Co-Graph. Der Co-Baum T von G besteht aus der Vereinigung der beiden Co-Bäume T_1 von G_1 und T_2 von G_2 sowie einem zusätzlichen Knoten. T enthält zusätzlich zu den Kanten in T_1 und T_2 eine Kante vom neuen Knoten zur Wurzel von T_1 und eine Kante vom neuen Knoten zur Wurzel von T_2. Der zusätzliche Knoten ist die Wurzel von T und mit $\cup$ markiert.*

- *Sei $G = G_1 \times G_2$ ein ungerichteter Co-Graph. Der Co-Baum T von G besteht aus der Vereinigung der beiden Co-Bäume T_1 von G_1 und T_2 von G_2 sowie einem zusätzlichen Knoten. T enthält zusätzlich zu den Kanten in T_1 und T_2 eine Kante vom neuen Knoten zur Wurzel von T_1 und eine Kante vom neuen Knoten zur Wurzel von T_2. Der zusätzliche Knoten ist die Wurzel von T und mit $\times$ markiert.*

Für einen Co-Graphen G mit Co-Baum T werden in dieser Arbeit ein Knoten $u \in V(G)$ und das mit ihm assoziierte Blatt in T äquivalent verwendet. Zudem wird für einen Knoten $w \in T$ mit T_w der Teilbaum von T mit Wurzel w notiert. Der von den mit den Blättern von T_w assoziierten Knoten induzierte Teilgraph von G wird mit G_w bezeichnet.

Der Co-Baum für einen ungerichteten Co-Graph ist nicht eindeutig und kann in linearer Zeit berechnet werden [65]. Des Weiteren kann ein sogenannter kanonischer Co-Baum bestimmt werden, der aufeinanderfolgende Vereinigungen oder Joins zusammenfasst und somit nicht binär ist. Dieser kanonische Co-Baum ist für einen ungerichteten Co-Graph eindeutig und kann ebenfalls in linearer Zeit berechnet werden [50]. Für ein Beispiel siehe Abbildung 14.

Der berechnete Co-Baum wird bei dynamischen Ansätzen häufig von den Blättern bis zur Wurzel durchlaufen. Wird der Co-Baum nur konstant oft durchlaufen und werden an jedem Knoten des Baums nur konstant viele Informationen berechnet, können Probleme in linearer Zeit gelöst werden, da die Größe des Co-Baums nur linear in der Größe des entsprechenden Graphen ist. Mit diesem Ansatz kann für Co-Graphen beispielsweise ein minimales Vertex Cover in linearer Zeit bestimmt werden, was in 6.2.4 benötigt wird.

5.3 Linearzeitalgorithmus für gerichtete Co-Graphen

Gerichtete Co-Graphen sind rekursiv über drei Operationen definiert. Die ersten beiden sind die gerichteten Äquivalente zu den ungerichteten Operationen. Demnach sind ungerichtete Co-Graphen, bei denen jede ungerichtete Kante durch zwei entgegengesetzte gerichtete ersetzt wird, ebenfalls gerichtete Co-Graphen. Gerichtete Co-Graphen sind wie folgt definiert. Für ein Beispiel siehe Abbildung 15.

Definition 18. *[12]*

- *Der Graph mit nur einem Knoten ist ein gerichteter Co-Graph.*
- *Seien G_1 und G_2 zwei gerichtete Co-Graphen. Die* Vereinigung *$G = G_1 \cup G_2$ dieser Graphen ist ein gerichteter Co-Graph mit Knotenmenge $V(G_1) \cup V(G_2)$ und Kantenmenge $E(G_1) \cup E(G_2)$.*
- *Seien G_1 und G_2 zwei gerichtete Co-Graphen. Der* Join *$G = G_1 \times G_2$ dieser Graphen ist ein gerichteter Co-Graph mit Knotenmenge $V(G_1) \cup V(G_2)$ und Kantenmenge $E(G_1) \cup E(G_2) \cup \{(u,v),(v,u) | u \in V(G_1), v \in V(G_2)\}$.*
- *Seien G_1 und G_2 zwei gerichtete Co-Graphen. Der* gerichtete Join *$G = G_1 \gg G_2$ dieser Graphen ist ein gerichteter Co-Graph mit Knotenmenge $V(G_1) \cup V(G_2)$ und Kantenmenge $E(G_1) \cup E(G_2) \cup \{(u,v) | u \in V(G_1), v \in V(G_2)\}$.*

Für gerichtete Co-Graphen kann analog ein Co-Baum mit drei Markierungen für die inneren Knoten definiert werden, der ebenfalls in linearer Zeit berechnet werden kann [26]. In dieser Arbeit sind für den gerichteten Fall lediglich binäre Co-Bäume relevant.

Im restlichen Teil dieses Kapitels wird die gerichtete Metrische Dimension gerichteter Co-Graphen betrachtet. Hierfür werden die Distanzen von den Ankerknoten zu den anderen Knoten verwendet und es werden ausschließlich stark zusammenhängende Graphen betrachtet. Die Berechnung der gerichteten Metrischen Dimension basiert auf der Idee, die im Kontext der ungerichteten Co-Graphen angesprochen wurde und auch für gerichtete Co-Graphen gilt. Für keine der drei Operationen kann ein Knoten aus G_1 zwei Knoten aus G_2 trennen und vice versa. Daher wird der Co-Baum mit einem dynamischen Ansatz von den Blättern bis zur Wurzel durchlaufen und an jedem inneren Knoten eine trennende Menge für den Teilgraphen berechnet, der durch die Blätter im jeweiligen Teilbaum induziert wird. Diese wird aus den beiden vorher bestimmten Lösungen für den linken und rechten Teilbaum zusammengesetzt und gegebenenfalls durch maximal einen Knoten erweitert. Wichtig hierbei ist, dass die induzierten Teilgraphen mit den Distanzwerten des Gesamtgraphen betrachtet werden.

Lemma 1. *Sei G ein stark zusammenhängender gerichteter Co-Graph mit Co-Baum T und sei $w \in T$. Zwei Knoten $u, v \in V(G_w)$ können nur durch Knoten aus $V(G_w)$ getrennt werden.*

Beweis. Durch jede der drei Co-Graph Operationen erhalten alle Knoten aus $V(G_w)$ dieselbe Nachbarschaft außerhalb von G_w. Somit besitzt jeder Knoten, der nicht in $V(G_w)$ ist, dieselbe Distanz zu u und v und kann diese nicht trennen. □

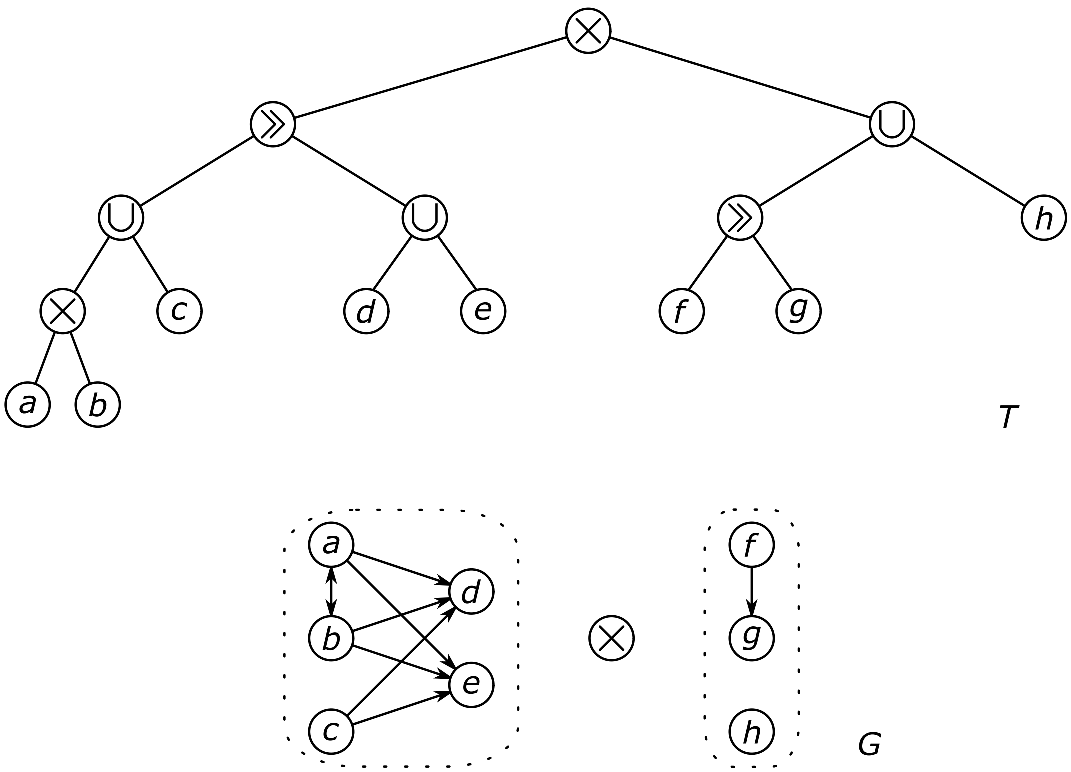


Abbildung 15: Ein gerichteter Co-Graph G (unten) mit einem Co-Baum T (oben). Zwischen den Knoten aus $\{a, b, c, d, e\}$ und den Knoten aus $\{f, g, h\}$ existieren alle gerichteten Kanten in G, die der Übersicht halber nicht eingezeichnet sind. Da G stark zusammenhängend ist, ist die Wurzel von T mit $\times$ markiert.

Aus dem vorherigen Lemma folgt, dass wenn U eine trennende Menge für einen stark zusammenhängenden gerichteten Co-Graphen G ist, $U \cap V(G_w)$ eine trennende Menge für G_w in G ist. Es muss allerdings keine trennende Menge für G_w sein, insbesondere da G_w nicht stark zusammenhängend sein muss.

Betrachte zum Beispiel den Graphen $G = ((G_1 \gg G_2) \cup G_3) \times G_4$, bei dem die vier Graphen G_1, G_2, G_3 und G_4 jeweils aus nur einem Knoten u_1, u_2, u_3 und u_4 bestehen. Der Knoten u_1 trennt alle Knoten der Menge $\{u_1, u_2, u_3\}$ in G, jedoch trennt er die Knoten u_2 und u_3 nicht in $(G_1 \gg G_2) \cup G_3$, da hier kein Weg von u_1 nach u_3 existiert und die Distanz somit undefiniert ist.

Das Ziel dieses Kapitels ist es, die gerichtete Metrische Dimension von gerichteten Co-Graphen mittels eines bottom-up Algorithmus zu berechnen. Nach Lemma 1 muss jedes Knotenpaar durch einen Knoten in dem Teilbaum getrennt werden, in dem das Knotenpaar zusammenkommt. Daher muss für jeden inneren Knoten des Co-Baums eine trennende Menge für die Knoten im entsprechenden Teilbaum berechnet werden. Somit sind besonders die Knoten relevant, zu denen alle Ankerknoten im Teilbaum dieselbe Distanz haben, da diese Knoten eventuell nicht von den Knoten der anderen Teilbäume getrennt werden. Da stark zusammenhängende gerichtete Co-Graphen einen Durchmesser von ≤ 2 haben, gibt es maximal zwei verschiedene Distanzen von einem Ankerknoten zu den anderen Knoten im Graphen. Demnach sind nur die folgenden zwei Arten von Knoten entscheidend.

Definition 19. *Sei G ein gerichteter Graph, sei $U \subseteq V(G)$ eine nicht leere Teilmenge der Knoten und sei $u \in V(G) \setminus U$. Der Knoten u heißt 1-Knoten bezüglich U, wenn $\forall w \in U :$ $(w, u) \in E(G)$. Der Knoten u heißt 2-Knoten bezüglich U, wenn $\forall w \in U : (w, u) \notin E(G)$.*

Wenn U eine trennende Menge für G ist, dann existiert offensichtlich maximal ein 1-Knoten (und maximal ein 2-Knoten) bezüglich U, da zwei dieser Knoten nicht durch die Knoten aus U voneinander getrennt werden würden.

Da im Algorithmus für jeden inneren Knoten des Co-Baums eine Lösung, sprich eine trennende Menge, aus den Teillösungen für die beiden Teilbäume bestimmt werden soll, werden zunächst die Fälle betrachtet, in denen die Vereinigung der Teillösungen ausreichend ist. Wenn ein innerer Knoten des Co-Baums mit $\cup$ oder $\times$ markiert ist und als Nachfolger ein Blatt und einen inneren Knoten besitzt, wird o.B.d.A. davon ausgegangen, dass das Blatt der rechte Nachfolger ist.

Lemma 2. *Sei G ein stark zusammenhängender gerichteter Co-Graph mit Co-Baum T. Sei $w \in T$ ein innerer Knoten von T und T_l und T_r der linke und rechte Teilbaum von w mit Wurzeln l und r. Sei U_l eine trennende Menge für $V(G_l)$ in G und sei U_r eine trennende Menge für $V(G_r)$ in G. Wenn l oder r ein Blatt ist, ist $U_l = \emptyset$ beziehungsweise $U_r = \emptyset$.*

1. *w ist mit $\cup$ markiert.*

   (a) *l und r sind innere Knoten.*
   $U_l \cup U_r$ ist eine minimale trennende Menge für $V(G_w)$ in G genau dann, wenn U_l eine minimale trennende Menge für $V(G_l)$ in G ist, U_r eine minimale trennende Menge für $V(G_r)$ in G ist, und G_l keinen 2-Knoten bezüglich U_l oder G_r keinen 2-Knoten bezüglich U_r besitzt.

   (b) *l ist ein innerer Knoten und r ein Blatt.*
   U_l ist eine minimale trennende Menge für $V(G_w)$ in G genau dann, wenn U_l eine minimale trennende Menge für $V(G_l)$ in G ist und G_l keinen 2-Knoten bezüglich U_l besitzt.

2. *w ist mit $\times$ markiert.*

   (a) *l und r sind innere Knoten.*
   $U_l \cup U_r$ ist eine minimale trennende Menge für $V(G_w)$ in G genau dann, wenn U_l eine minimale trennende Menge für $V(G_l)$ in G ist, U_r eine minimale trennende Menge für $V(G_r)$ in G ist, und G_l keinen 1-Knoten bezüglich U_l oder G_r keinen 1-Knoten bezüglich U_r besitzt.

   (b) *l ist ein innerer Knoten und r ein Blatt.*

U_l ist eine minimale trennende Menge für $V(G_w)$ in G genau dann, wenn U_l eine minimale trennende Menge für $V(G_l)$ in G ist und G_l keinen 1-Knoten bezüglich U_l besitzt.

3. *w ist mit $\gg$ markiert.*

   (a) *l und r sind innere Knoten.*

   $U_l \cup U_r$ ist eine minimale trennende Menge für $V(G_w)$ in G genau dann, wenn U_l eine minimale trennende Menge für $V(G_l)$ in G ist, U_r eine minimale trennende Menge für $V(G_r)$ in G ist, und G_l keinen 1-Knoten bezüglich U_l oder G_r keinen 2-Knoten bezüglich U_r besitzt.

   (b) *l ist ein innerer Knoten und r ein Blatt.*

   U_l ist eine minimale trennende Menge für $V(G_w)$ in G genau dann, wenn U_l eine minimale trennende Menge für $V(G_l)$ in G ist und G_l keinen 1-Knoten bezüglich U_l besitzt.

   (c) *l ist ein Blatt und r ein innerer Knoten.*

   U_r ist eine minimale trennende Menge für $V(G_w)$ in G genau dann, wenn U_r eine minimale trennende Menge für $V(G_r)$ in G ist und G_r keinen 2-Knoten bezüglich U_l besitzt.

Beweis. Da zwei Knoten aus $V(G_l)$ beziehungsweise zwei Knoten aus $V(G_r)$ in G durch mindestens einen Knoten der zugehörigen trennenden Menge getrennt werden, verbleibt nur der Fall, dass ein Knoten $v_l \in V(G_l)$ und ein Knoten $v_r \in V(G_r)$ in G getrennt werden müssen.

1. w ist mit $\cup$ markiert.

   (a) l und r sind innere Knoten.

   Wenn v_l kein 2-Knoten bezüglich U_l ist, dann existiert ein Knoten $u_l \in U_l$, sodass $d_G(u_l, v_l) = 1$. Wenn v_r kein 2-Knoten bezüglich U_r ist, dann existiert ein Knoten $u_r \in U_r$, sodass $d_G(u_r, v_r) = 1$. Da $d_G(u_l, v_r) = d_G(u_r, v_l) = 2$, werden v_l und v_r durch u_l oder u_r in G getrennt. Somit ist $U_l \cup U_r$ eine trennende Menge für $V(G_w)$ in G. Nach Lemma 1 ist $U_l \cup U_r$ eine minimale trennende Menge für $V(G_w)$ in G, wenn U_l eine minimale trennende Menge für $V(G_l)$ in G und U_r eine minimale trennende Menge für $V(G_r)$ in G ist.

   Wenn v_l ein 2-Knoten bezüglich U_l und v_r ein 2-Knoten bezüglich U_r ist, dann gilt für jeden Knoten $u \in U_l \cup U_r$, dass $d_G(u, v_l) = d_G(u, v_r) = 2$. Somit ist $U_l \cup U_r$ keine trennende Menge für $V(G_w)$ in G.

   (b) l ist ein innerer Knoten und r ein Blatt.

   Wenn v_l kein 2-Knoten bezüglich U_l ist, dann existiert ein Knoten $u_l \in U_l$, sodass $d_G(u_l, v_l) = 1$. Da $d_G(u_l, v_r) = 2$, werden v_l und v_r durch u_l in G getrennt. Somit ist U_l eine trennende Menge für $V(G_w)$ in G. Nach Lemma 1 ist U_l eine minimale trennende Menge für $V(G_w)$ in G, wenn U_l eine minimale trennende Menge für $V(G_l)$ in G ist.

Wenn v_l ein 2-Knoten bezüglich U_l ist, dann gilt für jeden Knoten $u \in U_l$, dass $d_G(u, v_l) = d_G(u, v_r) = 2$. Somit ist U_l keine trennende Menge für $V(G_w)$ in G.

2. w ist mit $\times$ markiert.

   Der Beweis verläuft analog zum vorherigen Fall.

3. w ist mit $\gg$ markiert.

   (a) l und r sind innere Knoten.

   Wenn v_l kein 1-Knoten bezüglich U_l ist, dann existiert ein Knoten $u_l \in U_l$, sodass $d_G(u_l, v_l) = 2$. Wenn v_r kein 2-Knoten bezüglich U_r ist, dann existiert ein Knoten $u_r \in U_r$, sodass $d_G(u_r, v_r) = 1$. Da $d_G(u_l, v_r) = 1$ und $d_G(u_r, v_l) = 2$, werden v_l und v_r durch u_l oder u_r in G getrennt. Somit ist $U_l \cup U_r$ eine trennende Menge für $V(G_w)$ in G. Nach Lemma 1 ist $U_l \cup U_r$ eine minimale trennende Menge für $V(G_w)$ in G, wenn U_l eine minimale trennende Menge für $V(G_l)$ in G und U_r eine minimale trennende Menge für $V(G_r)$ in G ist.

   Wenn v_l ein 1-Knoten bezüglich U_l und v_r ein 2-Knoten bezüglich U_r ist, dann gilt für jeden Knoten $u \in U_l$, dass $d_G(u, v_l) = d_G(u, v_r) = 1$ und für jeden Knoten $v \in U_r$, dass $d_G(u, v_l) = d_G(u, v_r) = 2$. Somit ist $U_l \cup U_r$ keine trennende Menge für $V(G_w)$ in G.

   (b) l ist ein innerer Knoten und r ein Blatt.

   Wenn v_l kein 1-Knoten bezüglich U_l ist, dann existiert ein Knoten $u_l \in U_l$, sodass $d_G(u_l, v_l) = 2$. Da $d_G(u_l, v_r) = 1$, werden v_l und v_r durch u_l in G getrennt. Somit ist U_l eine trennende Menge für $V(G_w)$ in G. Nach Lemma 1 ist U_l eine minimale trennende Menge für $V(G_w)$ in G, wenn U_l eine minimale trennende Menge für $V(G_l)$ in G ist.

   Wenn v_l ein 1-Knoten bezüglich U_l ist, dann gilt für jeden Knoten $u \in U_l$, dass $d_G(u, v_l) = d_G(u, v_r) = 1$. Somit ist U_l keine trennende Menge für $V(G_w)$ in G.

   (c) l ist ein Blatt und r ein innerer Knoten.

   Wenn v_r kein 2-Knoten bezüglich U_r ist, dann existiert ein Knoten $u_r \in U_r$, sodass $d_G(u_r, v_r) = 1$. Da $d_G(u_r, v_l) = 2$, werden v_l und v_r durch u_r in G getrennt. Somit ist U_r eine trennende Menge für $V(G_w)$ in G. Nach Lemma 1 ist U_r eine minimale trennende Menge für $V(G_w)$ in G, wenn U_r eine minimale trennende Menge für $V(G_r)$ in G ist.

   Wenn v_r ein 2-Knoten bezüglich U_r ist, dann gilt für jeden Knoten $u \in U_r$, dass $d_G(u, v_l) = d_G(u, v_r) = 2$. Somit ist U_r keine trennende Menge für $V(G_w)$ in G.

□

Aus dem vorherigen Beweis geht hervor, dass wenn $U_l \cup U_r$ keine trennende Menge für $V(G_w)$ in G ist, maximal ein weiterer Ankerknoten gewählt werden muss. Wenn w mit

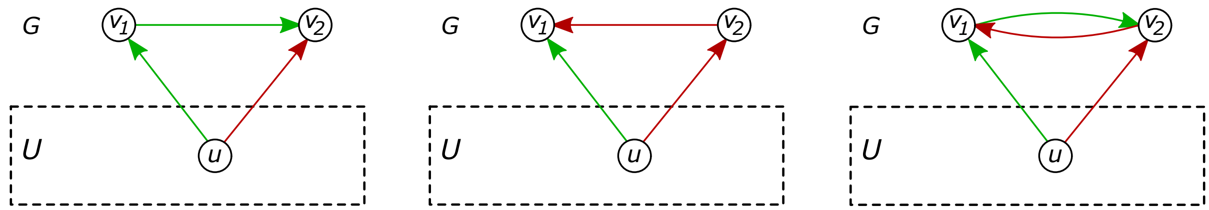
$\cup$ markiert ist, kann einer der beiden 2-Knoten gewählt werden. In beiden Fällen bleibt der jeweils andere Knoten ein 2-Knoten und eventuell vorhandene 1-Knoten verlieren ihre Eigenschaft. Auch für den Basisfall, dass l und r Blätter sind, kann beliebig einer der beiden Knoten als Ankerknoten gewählt werden. Lediglich in dem Fall, dass r ein Blatt ist, gibt es einen Unterschied. Wenn der 2-Knoten v_l aus G_l gewählt wird, ist das Blatt ein 2-Knoten bezüglich $U_l \cup \{v_l\}$. Wenn in $V(G_l)$ ein 1-Knoten bezüglich U_l existiert, behält oder verliert dieser seine Eigenschaft, je nachdem, ob v_l eine Kante zu diesem Knoten besitzt. Wird hingegen das Blatt als Ankerknoten gewählt, bleibt der 2-Knoten aus G_l ein 2-Knoten; wenn jedoch ein 1-Knoten existiert, verliert er immer seine Eigenschaft. Da nur die 1- und 2-Knoten für den Algorithmus relevant sind, gilt die Möglichkeit das Blatt als Ankerknoten zu wählen als mindestens genauso gut wie jede andere Lösung, sodass sie bevorzugt wird. Wenn w mit $\times$ markiert ist, kann analog verfahren werden.

Lediglich der Fall, dass w mit $\gg$ markiert ist, benötigt weitere Überlegungen. Zum einen existieren für den Basisfall zwei Möglichkeiten, bei denen entweder ein 1-Knoten oder ein 2-Knoten entsteht, je nachdem, welches Blatt als Ankerknoten gewählt wird. Zum anderen sind die Knoten, die eventuell nicht getrennt werden, nicht die Knoten, die ihre Eigenschaft beibehalten. So muss der 1-Knoten aus $V(G_l)$ von dem 2-Knoten aus $V(G_r)$ getrennt werden. Hierzu kann wieder einer der beiden Knoten selbst verwendet werden und jeder der beiden Knoten verliert seine Eigenschaft, wenn im jeweils anderen Teilbaum mindestens ein Ankerknoten existiert. Jedoch sind es der 2-Knoten aus $V(G_l)$ und der 1-Knoten aus $V(G_r)$ die, basierend auf den vorhandenen Kanten, ihre Eigenschaft beibehalten können, wenn ein Ankerknoten gewählt wird. Offensichtlich kann maximal einer der beiden Knoten seine Eigenschaft verlieren, wenn nur ein weiterer Ankerknoten gewählt wird. Wünschenswert wäre demnach ein Knoten, der sowohl den 1-Knoten aus G_l von dem 2-Knoten aus G_r trennt, als auch entweder den 2-Knoten aus G_l oder den 1-Knoten aus G_r seine Eigenschaft verlieren lässt. Diese gesuchten Knoten werden in der folgenden Definition behandelt.

Definition 20. *Sei G ein stark zusammenhängender gerichteter Graph und $U \subseteq V(G)$ eine trennende Menge für G. Sei v_1 ein 1-Knoten und v_2 ein 2-Knoten bezüglich U. Ein Knoten u heißt* double remover*, wenn G keinen 1- und keinen 2-Knoten bezüglich $U \cup \{u\}$ besitzt.*

Abbildung 16 zeigt, welche Kanten vorhanden sein müssen und welche nicht vorhanden sein dürfen, damit ein 1-Knoten, ein 2-Knoten oder beide gleichzeitig ein double remover sind. Für den dritten Fall existiert kein gerichteter Co-Graph mit der geforderten Kantenmenge. Das nächste Lemma zeigt zudem, dass nur 1- oder 2-Knoten double remover sein können, siehe auch Abbildung 17.

Lemma 3. *Sei G ein stark zusammenhängender gerichteter Co-Graph mit Co-Baum T. Sei $w \in T$ ein innerer Knoten und sei U eine trennende Menge für $V(G_w)$ in G, sodass ein 1-Knoten v_1 und ein 2-Knoten v_2 bezüglich U in G_w existiert. Dann existiert kein double remover für v_1 und v_2 in $V(G_w) \setminus \{v_1, v_2\}$.*



(a) (b) (c)

Abbildung 16: Die drei Fälle in denen (a) ein 1-Knoten, (b) ein 2-Knoten oder (c) beide ein double remover sind. Die grünen Kanten müssen vorhanden sein, die roten Kanten dürfen nicht existieren. Der dritte Fall ist kein gerichteter Co-Graph.

Beweis. Angenommen es gäbe einen solchen double remover $u \in V(G_w) \setminus \{v_1, v_2\}$. Da U eine trennende Menge für $V(G_w)$ in G ist, werden insbesondere die drei Knotenpaare v_1, v_2, v_1, u und v_2, u durch Knoten aus U in G getrennt. Da v_1 ein 1-Knoten und v_2 ein 2-Knoten bezüglich U sind, trennt jeder Knoten aus U dieses Paar. Außerdem existieren mindestens zwei Knoten $w_1, w_2 \in U$, wobei w_1 das Paar v_1, u und w_2 das Paar v_2, u trennt. Somit gilt $(w_1, v_1), (w_2, v_1), (w_2, u) \in E(G)$ und $(w_1, v_2), (w_2, v_2), (w_1, u) \notin E(G)$. Da u ein double remover ist gilt $(u, v_1) \notin E(G)$ und $(u, v_2) \in E(G)$. Für diese Knoten- und Kantenmenge existiert keine Aufteilung in zwei nicht leere Graphen G_1, G_2, sodass $G = G_1 \cup G_2$, $G = G_1 \times G_2$ oder $G = G_1 \gg G_2$, siehe auch Abbildung 17. □

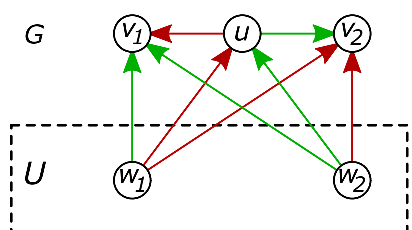


Abbildung 17: Der Fall in dem ein Knoten u ein double remover für den 1-Knoten v_1 und den 2-Knoten v_2 ist. Die grünen Kanten müssen vorhanden sein, die roten Kanten dürfen nicht existieren. Es müssen mindestens zwei Ankerknoten in U existieren, um die Knoten u, v_1, v_2 untereinander zu trennen. Für diesen Fall ist G kein gerichteter Co-Graph.

Die gerichtete Metrische Dimension kann nun wie folgt für einen stark zusammenhängenden gerichteten Co-Graphen G mit Co-Baum T berechnet werden. Für jeden inneren Knoten $w \in T$ werden maximal vier trennende Mengen für $V(G_w)$ in G berechnet. Diese werden mit U_{w,t_w} bezeichnet, wobei $t_w \in \{0, 1, 2, 12\}$. Hierbei ist

1. $U_{w,0}$ eine minimale trennende Menge für $V(G_w)$ in G, sodass G_w keinen 1-Knoten oder 2-Knoten bezüglich $U_{w,0}$ enthält.

2. $U_{w,1}$ eine minimale trennende Menge für $V(G_w)$ in G, sodass G_w einen 1-Knoten, aber keinen 2-Knoten bezüglich $U_{w,1}$ enthält.

3. $U_{w,2}$ eine minimale trennende Menge für $V(G_w)$ in G, sodass G_w keinen 1-Knoten, aber einen 2-Knoten bezüglich $U_{w,2}$ enthält.

4. $U_{w,12}$ eine minimale trennende Menge für $V(G_w)$ in G, sodass G_w einen 1-Knoten und einen 2-Knoten bezüglich $U_{w,12}$ enthält.

Zusätzlich wird für jede der Mengen, wenn vorhanden, gespeichert, welcher Knoten ein 1- oder 2-Knoten ist. Für $U_{w,12}$ wird außerdem gespeichert, ob einer der beiden Knoten ein double remover ist und wenn ja welcher.

Für jeden inneren Knoten werden die Lösungen der beiden Teilbäume kombiniert und maximal ein weiterer Knoten als Ankerknoten gewählt. Dieser zusätzliche Knoten ist immer ein 1- oder 2-Knoten. Initial sind für jeden inneren Knoten alle vier trennenden Mengen undefiniert und nicht für jeden Knoten existiert eine trennende Menge für jeden der vier Typen. Die Kombinationsmöglichkeiten sind in den Tabellen 1 bis 3 zusammengefasst. Diese zeigen, wie die trennenden Mengen U_{l,t_l} und U_{r,t_r} des linken und rechten Teilbaums zu einer trennenden Menge U_{w,t_w} zusammengesetzt werden können, wofür in einigen Fällen der Knoten u als Ankerknoten aufgenommen werden muss. G_w enthält dann den 1-Knoten v_1 und den 2-Knoten v_2 bezüglich U_{w,t_w}. Alle double remover, die entstehen oder als Ankerknoten verwendet werden, sind mit einem Stern gekennzeichnet. Wenn ein Eintrag mit „$-$“ markiert ist, muss kein Ankerknoten hinzugenommen werden oder es existiert kein 1- beziehungsweise 2-Knoten bezüglich der resultierenden trennenden Menge. Die blau hinterlegten Zellen entsprechen zwar zulässigen Kombinationsmöglichkeiten, können jedoch vernachlässigt werden, da, wie weiter oben besprochen, Lösungen existieren, die mindestens genauso gut sind. Für ein Beispiel siehe Abbildung 18.

Da der Co-Baum nur einmal durchlaufen werden muss und die Berechnungen an jedem Knoten des Baums nur konstante Zeit benötigen, folgt das nachstehende Theorem.

Theorem 5. *Die gerichtete Metrische Dimension eines stark zusammenhängenden gerichteten Co-Graphen kann in linearer Zeit berechnet werden.*

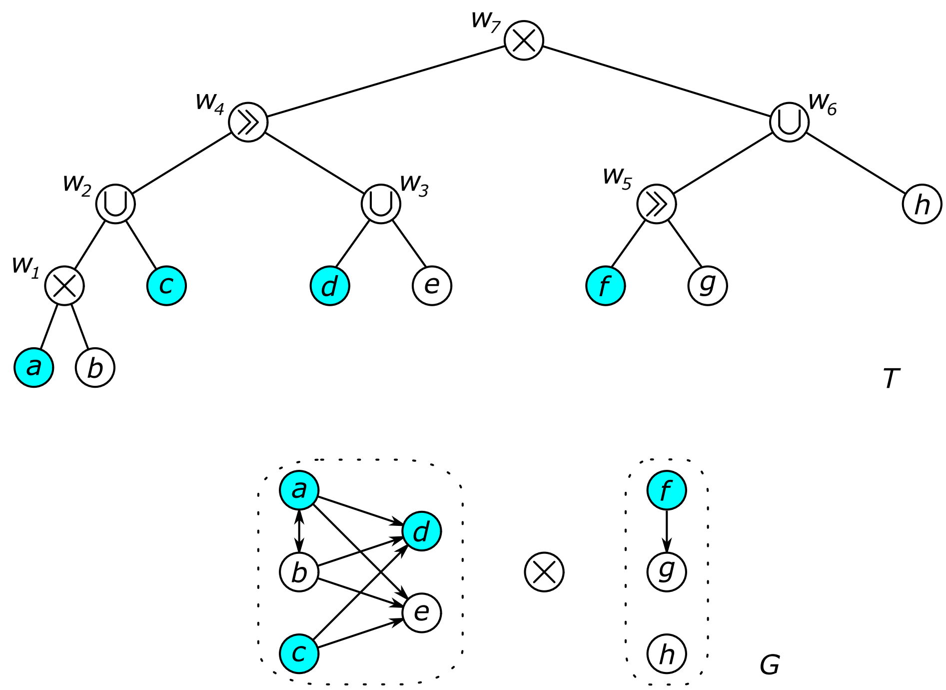


Abbildung 18: Der Graph aus Abbildung 15 mit einer trennenden Menge (blau). Im Teilbaum mit den Blättern $\{a, b, c\}$ ist c ein double remover für den 1-Knoten b und den 2-Knoten c bezüglich der trennenden Menge $\{a\}$. Am nächsten inneren Knoten (mit $\gg$ markiert) wird c als Ankerknoten gewählt, sodass für den entsprechenden Teilbaum die trennende Menge $\{a, c, d\}$ gewählt wird. Dies entspricht dem Fall „12 2 $v_{l,\star}$ 0 $-$ $-$“ aus Tabelle 1 für die Markierung $\gg$. Zur Berechnung der stark trennenden Mengen für jeden inneren Knoten siehe die untenstehende Tabelle.

w	$U_{w,0}$	$U_{w,1}/v_1$	$U_{w,2}/v_2$	$U_{w,12}/v_1, v_2$
w_1	–	$\{a\}/b$	–	–
w_2	–	–	–	$\{a\}/b, c^\star$
w_3	–	–	$\{d\}/e$	–
w_4	$\{a, c, d\}$	–	–	–
w_5	–	$\{f\}/g$	$\{g\}/f$	–
w_6	–	–	$\{f, g\}/h$	$\{f\}/g, h^\star$
w_7	$\{a, c, d, f\}$	–	–	–

t_l	t_r	u	t_w	v_1	v_2
0	0	–	0	–	–
0	1	–	0	–	–
0	2	–	2	–	$v_{r,2}$
0	12	–	2	–	$v_{r,2}$
1	0	–	0	–	–
1	1	–	0	–	–
1	2	–	2	–	$v_{r,2}$
1	12	–	2	–	$v_{r,2}$
2	0	–	2	–	$v_{l,2}$
2	1	–	2	–	$v_{l,2}$
2	2	$v_{l,2}$	2	–	$v_{r,2}$
		$v_{r,2}$	2	–	$v_{l,2}$
2	12	$v_{l,2}$	2	–	$v_{r,2}$
		$v_{r,2}$	2	–	$v_{l,2}$
		$v_{r,\star}$	2	–	$v_{l,2}$
12	0	–	2	–	$v_{l,2}$
12	1	–	2	–	$v_{l,2}$
12	2	$v_{l,2}$	2	–	$v_{r,2}$
		$v_{l,\star}$	2	–	$v_{r,2}$
		$v_{r,2}$	2	–	$v_{l,2}$
12	12	$v_{l,2}$	2	–	$v_{r,2}$
		$v_{l,\star}$	2	–	$v_{r,2}$
		$v_{r,2}$	2	–	$v_{l,2}$
		$v_{r,\star}$	2	–	$v_{l,2}$

$\cup$ Markierung

t_l	t_r	u	t_w	v_1	v_2
0	0	–	0	–	–
0	1	–	1	$v_{r,1}$	–
0	2	–	0	–	–
0	12	–	1	$v_{r,1}$	–
1	0	–	1	$v_{l,1}$	–
1	1	$v_{l,1}$	1	$v_{r,1}$	–
		$v_{r,1}$	1	$v_{l,1}$	–
1	2	–	1	$v_{l,1}$	–
1	12	$v_{l,1}$	1	$v_{r,1}$	–
		$v_{r,1}$	1	$v_{l,1}$	–
		$v_{r,\star}$	1	$v_{l,1}$	–
2	0	–	0	–	–
2	1	–	1	$v_{r,1}$	–
2	2	–	0	–	–
2	12	–	1	$v_{r,1}$	–
12	0	–	1	$v_{l,1}$	–
12	1	$v_{l,1}$	1	$v_{r,1}$	–
		$v_{l,\star}$	1	$v_{r,1}$	–
		$v_{r,1}$	1	$v_{l,1}$	–
12	2	–	1	$v_{l,1}$	–
12	12	$v_{l,1}$	1	$v_{r,1}$	–
		$v_{l,\star}$	1	$v_{r,1}$	–
		$v_{r,1}$	1	$v_{l,1}$	–
		$v_{r,\star}$	1	$v_{l,1}$	–

$\times$ Markierung

t_l	t_r	u	t_w	v_1	v_2
0	0	–	0	–	–
0	1	–	1	$v_{r,1}$	–
0	2	–	0	–	–
0	12	–	1	$v_{r,1}$	–
1	0	–	0	–	–
1	1	–	1	$v_{r,1}$	–
1	2	$v_{l,1}$	0	–	–
		$v_{r,2}$	0	–	–
1	12	$v_{l,1}$	1	$v_{r,1}$	–
		$v_{r,2}$	1	$v_{r,1}$	–
		$v_{r,\star}$	0	–	–
2	0	–	2	–	$v_{l,2}$
2	1	–	12	$v_{r,1}$	$v_{l,2}$
2	2	–	2	–	$v_{l,2}$
2	12	–	12	$v_{r,1}$	$v_{l,2}$
12	0	–	2	–	$v_{l,2}$
12	1	–	12	$v_{r,1}$	$v_{l,2}$
12	2	$v_{l,1}$	2	–	$v_{l,2}$
		$v_{l,\star}$	0	–	–
		$v_{r,2}$	2	–	$v_{l,2}$
12	12	$v_{l,1}$	12	$v_{r,1}$	$v_{l,2}$
		$v_{l,\star}$	1	$v_{r,1}$	–
		$v_{r,2}$	12	$v_{r,1}$	$v_{l,2}$
		$v_{r,\star}$	2	–	$v_{l,2}$

$\gg$ Markierung

Tabelle 1: Die Tabellen für den Fall, dass l und r innere Knoten in T sind.

t_l	t_r	u	t_w	v_1	v_2
0	–	–	2	–	v_r
1	–	–	12	$v_{l,1}$	$v_r^\star$
2	–	$v_{l,2}$	2	–	v_r
		v_r	2	–	$v_{l,2}$
12	–	$v_{l,2}$	12	$v_{l,1}$	$v_r^\star$
		$v_{l,\star}$	2	–	v_r
		v_r	2	–	$v_{l,2}$

$\cup$ Markierung, r ist Blatt

t_l	t_r	u	t_w	v_1	v_2
0	–	–	1	v_r	–
1	–	$v_{l,1}$	1	v_r	–
		v_r	1	$v_{l,1}$	–
2	–	–	12	$v_r^\star$	$v_{l,2}$
12	–	$v_{l,1}$	12	$v_r^\star$	$v_{l,2}$
		$v_{l,\star}$	1	v_r	–
		v_r	1	$v_{l,1}$	–

$\times$ Markierung, r ist Blatt

t_l	t_r	u	t_w	v_1	v_2
0	–	–	1	v_r	–
1	–	$v_{l,1}$	1	v_r	–
		v_r	0	–	–
2	–	–	12	v_r	$v_{l,2}$
12	–	$v_{l,1}$	12	v_r	$v_{l,2}$
		$v_{l,\star}$	1	v_r	–
		v_r	2	–	$v_{l,2}$

$\gg$ Markierung, r ist Blatt

t_l	t_r	u	t_w	v_1	v_2
–	0	–	2	–	v_l
–	1	–	12	$v_{r,1}$	v_l
–	2	v_l	0	–	–
		$v_{r,2}$	2	–	v_l
–	12	v_l	1	$v_{r,1}$	–
		$v_{r,2}$	12	$v_{r,1}$	v_l
		$v_{r,\star}$	2	–	v_l

$\gg$ Markierung, l ist Blatt

Tabelle 2: Die Tabellen für den Fall, dass l oder r ein Blatt in T ist.

U_{w,t_w}	t_w	v_1	v_2
$\{v_l\}$	2	–	v_r
$\{v_r\}$	2	–	v_l

$\cup$ Markierung, zwei Blätter

U_{w,t_w}	t_w	v_1	v_2
$\{v_l\}$	1	v_r	–
$\{v_r\}$	1	v_l	–

$\times$ Markierung, zwei Blätter

U_{w,t_w}	t_w	v_1	v_2
$\{v_l\}$	1	v_r	–
$\{v_r\}$	2	–	v_l

$\gg$ Markierung, zwei Blätter

Tabelle 3: Die Tabellen für den Fall, dass l und r Blätter in T sind.

6 Komplexität der Starken Metrischen Dimension

Oellermann und Peters-Fransen zeigten die Äquivalenz zwischen der Starken Metrischen Dimension eines Graphen G und einem Vertex Cover im G_{SR} [99], welcher wie folgt definiert ist. Für ein Beispiel siehe Abbildung 19.

Definition 21. *[99] Sei G ein ungerichteter Graph.*

- *Ein Knoten $u \in V(G)$ ist* maximal entfernt *von einem Knoten $v \in V(G)$, wenn $\forall u' \in N(u) : d_G(u', v) \leq d_G(u, v)$.*
- *Zwei Knoten $u, v \in V(G)$ sind* gegenseitig maximal entfernt*, wenn u maximal entfernt von v und v maximal entfernt von u ist.*
- *Der* strong resolving Graph *von G, genannt G_{SR}, besitzt die gleiche Knotenmenge wie G und eine Kante zwischen zwei Knoten $u, v \in V(G)$ genau dann, wenn u und v gegenseitig maximal entfernt in G sind.*

Theorem 6. *[99] Sei G ein ungerichteter Graph, dann ist $\mathrm{smd}(G) = \mathrm{VC}(\,\mathrm{G_{SR}}\,)$, wobei $\mathrm{VC}(\,\mathrm{G_{SR}}\,)$ die Größe eines minimalen Vertex Cover von G_{SR} ist.*

Da sich der G_{SR} in polynomieller Zeit konstruieren lässt [99], kann die Starke Metrische Dimension für jeden Graphen effizient gelöst werden, für dessen strong resolving Graphen ein minimales Vertex Cover in polynomieller Zeit bestimmt werden kann.

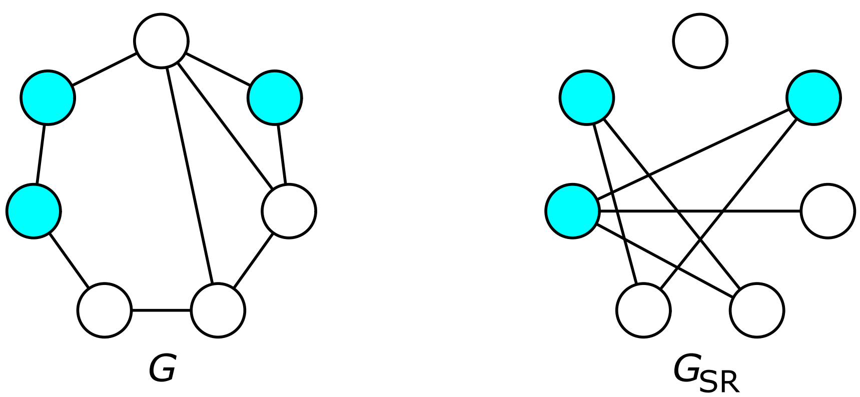


Abbildung 19: Ein Graph G links und sein strong resolving Graph G_{SR} rechts. Die drei markierten Knoten bilden ein Vertex Cover in G_{SR} und sind somit eine stark trennende Menge in G.

Es ist leicht zu erkennen, dass von zwei gegenseitig maximal entfernten Knoten mindestens einer in jeder stark trennenden Menge enthalten sein muss, da es keine Verlängerung des kürzesten Weges zwischen diesen Knoten gibt, auf der ein Ankerknoten liegen könnte. Auch für spezielle Arten von Knoten lassen sich Beobachtungen bezüglich der Starken Metrischen Dimension anstellen.

Beobachtung 1. *Sei G ein ungerichteter Graph.*

- *Eine minimale stark trennende Menge für G enthält keine Separationsknoten.*

- *Sei $d_G(u, v) = D(G)$ für zwei Knoten $u, v \in V(G)$, dann ist $\{u, v\} \in E(G_{\mathrm{SR}})$.*

Im Folgenden werden verschiedene Methoden zur Berechnung der Starken Metrischen Dimension betrachtet. Da die strong resolving Graphen vieler Graphklassen nicht perfekt sein muss (siehe Abbildung 20), werden entweder andere Herangehensweisen (Kapitel 6.1) oder Graphklassen, für deren strong resolving Graphen ein Vertex Cover effizient bestimmt werden kann (Kapitel 6.2.4), verwendet. Im Folgenden sind besonders twins relevant.

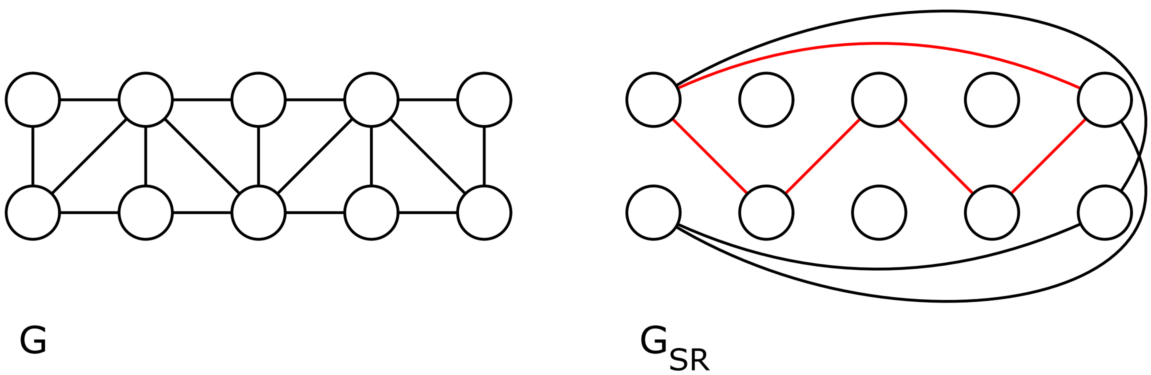


Abbildung 20: Ein G und sein strong resolving Graph G_{SR}. Der Graph G_{SR} enthält einen induzierten Kreis C_5 (rot), sodass er nicht perfekt ist. G kann nach dem gleichen Schema erweitert werden, um induzierte Kreise beliebiger Länge in G_{SR} zu erzeugen. G ist ein Intervallgraph und somit perfekt.

Lemma 4. *Sei G ein ungerichteter Graph.*

- *Wenn $u, v \in V(G)$ twins in G sind, sind sie true twins in G_{SR}.*

- *Sei $\{u, v\} \in E(G)$, dann ist $\{u, v\} \in E(G_{\mathrm{SR}})$ genau dann, wenn u und v true twins in G sind.*

Beweis.

- Da u und v twins sind, besitzen sie dieselben Distanzen zu allen übrigen Knoten in G und alle anderen Knoten besitzen dieselben Distanzen zu u und v. Somit ist u genau dann maximal entfernt von einem Knoten $w \in V(G)$, wenn v auch maximal entfernt von w ist und der Knoten w ist genau dann maximal entfernt von u, wenn er auch maximal entfernt von v ist.

- Da $d_G(u, v) = 1$, darf jeder Knoten aus $N(u)$ auch höchstens die Distanz 1 zu v haben, damit u und v gegenseitig maximal entfernt sind. Analog darf jeder Knoten aus $N(v)$ höchstens die Distanz 1 zu u haben, sodass $N[u] = N[v]$ gelten muss.

$\square$

Mithilfe der zuletzt gezeigten Aussage lässt sich das folgende Lemma beweisen.

Lemma 5. *Sei G ein ungerichteter Graph mit $D(G) \leq 2$. Dann ist*

$$E(G_{\mathrm{SR}}) = E(\overline{G}) \cup \{\{u, v\} \mid u, v \text{ sind true twins in } G\}.$$

Beweis. Da $D(G) \leq 2$ enthält G_{SR} aufgrund von Beobachtung 1 eine Kante zwischen jedem Knotenpaar, das in G nicht adjazent ist. Nach Lemma 4 enthält G_{SR} nur Kanten zwischen Knotenpaaren die adjazent in G sind, wenn sie true twins sind. Zusammen ergibt sich die Kantenmenge von $\overline{G}$ mit zusätzlichen Kanten zwischen true twins von G. □

6.1 Linearzeitalgorithmus für Co-Graphen

Wie bereits in einem vorherigen Kapitel beschrieben, sind Co-Graphen bezüglich der Komplementierung abgeschlossen. Außerdem kann durch das Einfügen einer Kante zwischen zwei false twins kein induzierter P_4 entstehen, sodass die nachstehende Aussage folgt.

Korollar 1. *Wenn G ein Co-Graph ist, dann ist auch G_{SR} ein Co-Graph.*

Diese Aussage ist bereits ausreichend, um die Starke Metrische Dimension eines ungerichteten Co-Graphen G in linearer Zeit zu berechnen. Dazu kann, falls nicht bereits vorhanden, der Co-Baum von G und anschließend der Co-Baum von G_{SR} mithilfe von Lemma 5 bestimmt werden. Zuletzt kann basierend auf diesem Baum ein Vertex Cover berechnet werden. Alle Schritte sind in linearer Zeit durchführbar. Die Starke Metrische Dimension von ungerichteten Co-Graphen lässt sich jedoch aufgrund der folgenden Eigenschaften ohne die Konstruktion des G_{SR} bestimmen.

Lemma 6. *Sei G ein ungerichteter Graph und seien $u, v \in V(G)$ zwei true twins in G. Dann ist* $\operatorname{smd}(G) = \operatorname{smd}(G \setminus v) + 1$.

Beweis. Seien u, v zwei true twins in G und sei $G' = G \setminus v$. Das Entfernen eines twins beeinflusst die Distanzen der übrigen Knoten nicht, sodass zwei Knoten $w_1, w_2 \in V(G) \setminus \{u, v\}$ genau dann gegenseitig maximal entfernt in G sind, wenn sie es in G' sind. Da u und v true twins sind, beeinflusst das Entfernen von v außerdem nicht, von welchen Knoten u maximal entfernt ist. Womit u und ein Knoten $w \in V(G) \setminus \{u, v\}$ genau dann gegenseitig maximal entfernt in G sind, wenn sie es in G' sind. Diese Eigenschaft gilt nicht für false twins, wenn $w \in N(u)$.

Jedes Vertex Cover U' von G'_{SR} kann durch das Hinzufügen von v in ein Vertex Cover für G_{SR} umgewandelt werden, d.h. $\operatorname{smd}(G') + 1 \geq \operatorname{smd}(G)$. Jedes Vertex Cover U von G_{SR} enthält entweder Knoten u, Knoten v oder beide, da diese in G_{SR} adjazent sind. Wenn U den Knoten v nicht enthält, ersetze u durch v in U. Das Ergebnis ist weiterhin ein Vertex Cover für G_{SR} und $U \setminus v$ ist ein Vertex Cover für G'_{SR}, d.h. $\operatorname{smd}(G) - 1 \geq \operatorname{smd}(G')$. □

Das Entfernen eines twins aus einem ungerichteten Graphen beeinflusst außerdem nicht die twin-Eigenschaft der anderen Knoten. Sind also $u_1, u_2 \in V(G)$ und $v_1, v_2 \in V(G)$ zwei Paare von twins in G, so bleiben u_1, u_2 twins in $G \setminus v_1$ und $G \setminus v_2$, was für das folgende Theorem relevant ist.

Theorem 7. *Sei G ein ungerichteter Co-Graph. Sei G' der Graph, der durch wiederholtes Entfernen von true twins in G entsteht. Dann ist*

$$\mathrm{smd}(G) = |V(G)| - \omega(G'),$$

wobei $\omega(G')$ die Größe einer maximalen Clique in G' ist.

Beweis. Sei $n = |V(G)|$ und $n' = |V(G')|$, dann ist

$$\begin{array}{rcll}
\mathrm{smd}(G) & = & \mathrm{smd}(G') + n - n' & \text{durch wiederholtes Anwenden von Lemma 6} \\
& = & \mathrm{VC}(\,G'_{\mathrm{SR}}\,) + n - n' & \text{nach Theorem 6} \\
& = & \mathrm{VC}(\,\overline{G'}\,) + n - n' & \text{nach Lemma 5, da } G' \text{ keine true twins hat} \\
& = & n' - \alpha(\overline{G'}) + n - n' & \text{da } \mathrm{VC}(\,\overline{G'}\,) = n' - \alpha(\overline{G'}) \\
& = & n - \alpha(\overline{G'}) & \\
& = & n - \omega(G') & \text{da } \alpha(\overline{G'}) = \omega(G'),
\end{array}$$

wobei $\mathrm{VC}(\,G\,)$ und $\alpha(G)$ die Größe eines minimalen Vertex Cover und einer maximalen unabhängigen Menge von G sind. □

Mithilfe dieser Aussage kann die Starke Metrische Dimension eines ungerichteten Co-Graphen G in linearer Zeit berechnet werden, ohne den G_{SR} betrachten zu müssen. Stattdessen wird eine maximale Clique in G berechnet, die keine true twins enthält. Die maximale Clique eines ungerichteten Co-Graphen kann einfach mithilfe des Co-Baums berechnet werden, welcher von den Blättern bis zur Wurzel durchlaufen wird. Ein Blatt bekommt den Wert 1 zugewiesen, ein innerer Knoten, der mit $\cup$ markiert ist, das Maximum der Werte seiner Nachfolger, und ein innerer Knoten, der mit $\times$ markiert, ist die Summe der Werte seiner Nachfolger.

Das Entfernen der true twins kann im selben Durchlauf durchgeführt werden. Hierzu wird der bereits in einem früheren Kapitel angesprochene kanonische Co-Baum verwendet. Zwei Knoten sind genau dann true twins in G, wenn die assoziierten Blätter im Co-Baum Nachfolger desselben mit $\times$ markierten Knotens sind. Somit müssen für die Berechnung der maximalen Clique lediglich die Blätter an den mit $\times$ markierten Knoten als ein einzelnes Blatt behandelt werden. Im folgenden Algorithmus 1 geschieht dies durch die Variable t, welche auf 1 gesetzt wird, wenn sich an einem mit $\times$ markierten Knoten ein Blatt befindet, die aber nicht weiter erhöht wird. Für ein Beispiel siehe Abbildung 21.

Theorem 8. *Die Starke Metrische Dimension ungerichteter Co-Graphen kann in linearer Zeit berechnet werden.*

Theorem 7 kann außerdem auf alle Graphklassen mit Durchmesser von höchstens 2 angewendet werden, wenn sie bezüglich des Entfernens von true twins abgeschlossen sind. Wenn zusätzlich eine maximale Clique effizient berechnet werden kann, kann die Starke Metrische Dimension ebenfalls effizient bestimmt werden. Beispiele für diese Graphklassen sind perfekte Graphen mit einem Durchmesser von höchstens 2 oder Graphen mit beschränkter Cliquenweite und einem Durchmesser von höchstens 2.

Algorithm 1 Starke Metrische Dimension ungerichteter Co-Graphen

```
function SMD(Co-Graph G)
    T ← kanonischer Co-Baum von G
    n ← |V(G)|
    return n−MAX_TWINFREIE_CLIQUE(T)
end

function MAX_TWINFREIE_CLIQUE(kanonischer Co-Baum T)
    w ← Wurzel von T
    if w ist mit ∪ markiert then
        m ← 0
        for all Teilbäume T′ an Wurzel w do
            k =MAX_TWINFREIE_CLIQUE(T′)
            if k > m then
                m ← k
        return m
    else if w ist mit × markiert then
        s ← 0
        t ← 0
        for all Teilbäume T′ an Wurzel w do
            if Wurzel von T′ ist ein Blatt then
                t ← 1
            else
                s ← s+MAX_TWINFREIE_CLIQUE(T′)
        return s + t
    else
        return 1
end
```

Korollar 2. *Die Starke Metrische Dimension von perfekten Graphen mit Durchmesser höchstens 2 kann in polynomieller Zeit berechnet werden.*

Beweis. Analog zu Theorem 7 können zuerst alle true twins durch wiederholtes Anwenden von Lemma 6 entfernt und anschließend eine maximale Clique berechnet werden. Da das Entfernen von true twins keine holes oder anti-holes erzeugt, ist der entstehende Graph ebenfalls perfekt und hat einen Durchmesser von höchstens 2. Mithilfe von Lemma 5 kann somit die Starke Metrische Dimension in polynomieller Zeit berechnet werden, da perfekte Graphen bezüglich Komplementierung abgeschlossen sind und eine maximale Clique in polynomieller Zeit bestimmt werden kann. □

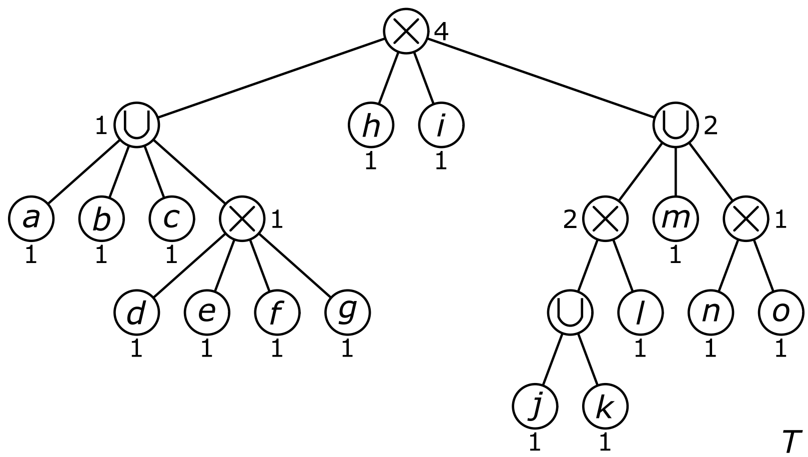


Abbildung 21: Ein Co-Baum T eines ungerichteten Co-Graphen. An den Knoten stehen die Werte, die von Algorithmus 1 berechnet werden. Eine größte twin-freie Clique für diesen Graphen besteht aus 4 Knoten, zum Beispiel $\{a, h, j, l\}$. Der Graph enthält 15 Knoten, sodass seine Starke Metrische Dimension 11 beträgt.

6.2 Definition und Analyse verknüpfter Graphen

Zu Beginn dieses Kapitels wurde bereits erwähnt, dass die Starke Metrische Dimension zum großen Teil für verschiedene Produkte von Graphen untersucht wurde. Das liegt daran, dass diese Graphen eine feste Struktur besitzen, die Aussagen über maximal entfernte Knoten ermöglichen. Insbesondere ist es häufig möglich effizient zu bestimmen, welche Knoten maximal entfernt von einem gegebenen Knoten sind. Demnach stellt sich die Frage, welche Strukturen gegeben sein müssen, damit die Starke Metrische Dimension eines Graphen effizient berechnet werden kann. Im Folgenden wird gezeigt, dass eine minimale stark trennende Menge genau dann für einen ungerichteten Graphen effizient bestimmt werden kann, wenn für jede zweifache Zusammenhangskomponente eine minimale stark trennende Menge effizient bestimmt werden kann. Insbesondere wird auch der Fall betrachtet, in dem diese Komponenten Co-Graphen sind.

Im folgenden werden ungerichtete Graphen betrachtet, die durch das Verknüpfen von Graphen $G_1, \ldots, G_k$ an einen Graphen H entstehen, indem in der disjunkten Vereinigung dieser Graphen Knoten $u_1, \ldots, u_k$ aus $G_1, \ldots, G_k$ mit Knoten $v_1, \ldots, v_k$ aus H verschmolzen werden.

Definition 22. *Seien $G_1, \ldots, G_k$ und H $k+1$ Graphen, $u_i \in V(G_i)$ für $i = 1, \ldots, k$ und $v_1, \ldots, v_k \in V(H)$. Sei $G_{1,\ldots,k}$ die disjunkte Vereinigung von $G_1, \ldots, G_k$. Das heißt $G_{1,\ldots,k}$ besitzt die Knotenmenge $\cup_{i=1}^k V(G_i)$ und die Kantenmenge $\cup_{i=1}^k E(G_i)$.*

Dann ist der Graph

$$G_{1,\ldots,k} \circ_{(u_1,\ldots,u_k \to v_1,\ldots,v_k)} H$$

definiert durch die Knotenmenge

$$(V(G_1) \cup \cdots \cup V(G_k) \cup V(H)) \setminus \{u_1, \ldots, u_k\}$$

und die Kantenmenge

$$\begin{array}{ll} (& E(G_1) \cup \cdots \cup E(G_k) \cup E(H) \\ & \cup \{\{w, v_i\} \mid w \in N_{G_i}(u_i), 1 \leq i \leq k\} \quad) \\ \setminus & \{\{w, u_i\} \mid w \in N_{G_i}(u_i), 1 \leq i \leq k\}. \end{array}$$

Der Graph $G_{1,\ldots,k} \circ_{(u_1,\ldots,u_k \to v_1,\ldots,v_k)} H$ entsteht aus der disjunkten Vereinigung der k Graphen $G_1, \ldots, G_k$ und des Graphen H durch das Entfernen der Knoten $u_1, \ldots, u_k$ und das anschließende Verbinden der Nachbarn von u_i in G_i mit v_i für $1 \leq i \leq k$. Im Folgenden wird nur der in Voraussetzung 1 beschriebene Fall betrachtet.

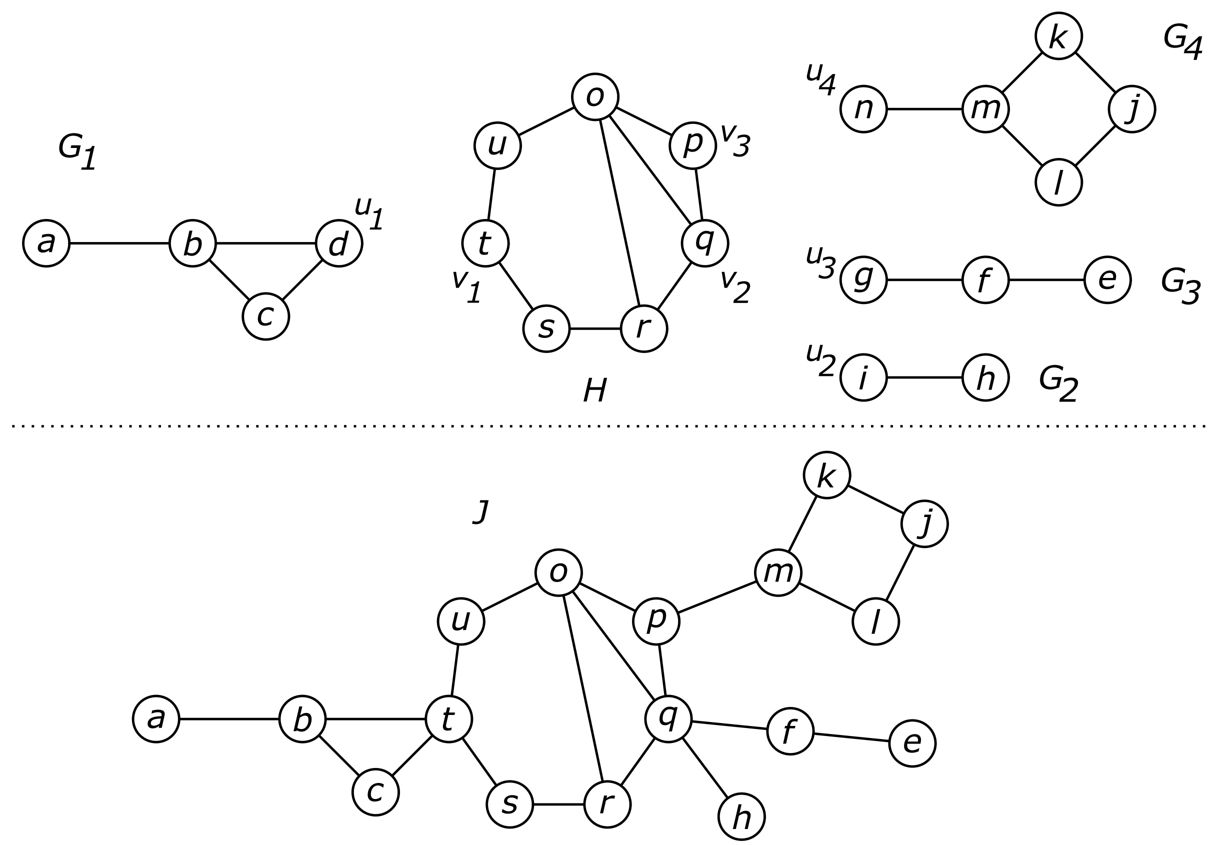


Abbildung 22: Fünf Graphen G_1, G_2, G_3, G_4, H und der Graph $J = G_{1,2,3,4} \circ_{(u_1,u_2,u_3,u_4 \to v_1,v_2,v_2,v_3)} H$. Der Graph H entspricht dem Graphen aus Abbildung 19.

Voraussetzung 1. *Seien $G_1, \ldots, G_k$ und H $k+1$ knotendisjunkte, zusammenhängende Graphen, $|V(G_i)| \geq 2$, $u_i \in V(G_i)$ für $1 \leq i \leq k$, $|V(H)| \geq 2$, $v_1, \ldots, v_k \in V(H)$. Sei $G_{1,\ldots,k}$ die disjunkte Vereinigung von $G_1, \ldots, G_k$ und sei*

$$J = G_{1,\ldots,k} \circ_{(u_1,\ldots,u_k \to v_1,\ldots,v_k)} H.$$

Die Knoten $v_1, \ldots, v_k$ in H müssen nicht disjunkt sein.

6.2.1 Bestimmung des strong resolving Graphen verknüpfter Graphen

Um die Starke Metrische Dimension eines Graphen $J = G_{1,\ldots,k} \circ_{(u_1,\ldots,u_k \to v_1,\ldots,v_k)} H$ zu berechnen, wird zunächst die Kantenmenge von J_{SR} bestimmt. Für die Verbindungsknoten ist die Kantenmenge trivial.

Lemma 7. *Sei Voraussetzung 1 gegeben. Die Knoten $v_1, \ldots, v_k$ besitzen keine inzidenten Kanten in J_{SR}.*

Beweis. Folgt aus Beobachtung 1, da $v_1, \ldots, v_k$ Separationsknoten in J sind. □

Da zur Bestimmung der Kanten in J_{SR} die von einem Knoten maximal entfernten Knoten relevant sind, werden diese mit der folgenden Definition festgehalten.

Definition 23. *Für einen zusammenhängenden Graphen G und einen Knoten $u \in V(G)$ sei* $\mathrm{MD}(\mathrm{G}, \mathrm{u})$ *die Menge an Knoten, die maximal entfernt von u in G sind.*

Zunächst wird die Menge der maximal entfernten Knoten für einen Knoten aus H betrachtet. Das folgende Lemma gilt auch, wenn w einer der Verbindungsknoten ist und ist zudem für Lemma 14 und Lemma 15 relevant. Anschließend wird die Menge der maximal entfernten Knoten für einen Knoten aus einem der Graphen G_j betrachtet. Für ein Beispiel siehe Abbildung 23.

Lemma 8. *Sei Voraussetzung 1 gegeben. Sei $w \in V(H)$, dann ist*

$$\mathrm{MD}(\mathrm{J}, \mathrm{w}) = \left(\begin{array}{ll} & \mathrm{MD}(\mathrm{H}, \mathrm{w}) \setminus \{v_1, \ldots, v_k\} \\ \cup & \bigcup_{1 \leq i \leq k} \mathrm{MD}(\mathrm{G_i}, \mathrm{u_i}) \end{array} \right).$$

Beweis. Die Aussage folgt daraus, dass jeder kürzeste Weg in J zwischen w und einem Knoten aus G_i für $1 \leq i \leq k$ den Knoten v_i passiert. Somit ist ein Knoten aus G_i genau dann maximal entfernt von w, wenn er maximal entfernt von v_i ist. Außerdem führt kein kürzester Weg zwischen w und einem Knoten aus H über Knoten außerhalb von H. Letztere beeinflussen demnach nicht, welche Knoten (außer den Knoten $v_1, \ldots, v_k$) in H maximal entfernt von w sind. Die Knoten $v_1, \ldots, v_k$ sind aufgrund von Lemma 7 von keinem Knoten maximal entfernt. □

Lemma 9. *Sei Voraussetzung 1 gegeben. Sei $u'_j \in V(G_j) \setminus \{u_j\}$ für ein j, $1 \leq j \leq k$. Dann ist*

$$\mathrm{MD}(\mathrm{J}, \mathrm{u'_j}) = \left(\begin{array}{ll} & \mathrm{MD}(\mathrm{G_j}, \mathrm{u'_j}) \setminus \{u_j\} \\ \cup & \mathrm{MD}(\mathrm{H}, \mathrm{v_j}) \setminus \{v_1, \ldots, v_k\} \\ \cup & \bigcup_{1 \leq i \leq k, i \neq j} \mathrm{MD}(\mathrm{G_i}, \mathrm{u_i}) \end{array} \right).$$

Beweis. Analog zum vorherigen Lemma verläuft jeder kürzeste Weg in J zwischen u'_j und einem Knoten aus H oder G_i mit $i \neq j$ über den Knoten u_j. Somit sind diese Knoten genau dann maximal entfernt von u'_j, wenn sie maximal entfernt von v_j bzw. u_i sind. Ebenfalls analog zum vorherigen Lemma haben die Knoten außerhalb von $V(G_j)$ keinen Einfluss auf die von u'_j maximal entfernten Knoten in G_j außer u_j. Der Knoten u_j wird in J durch

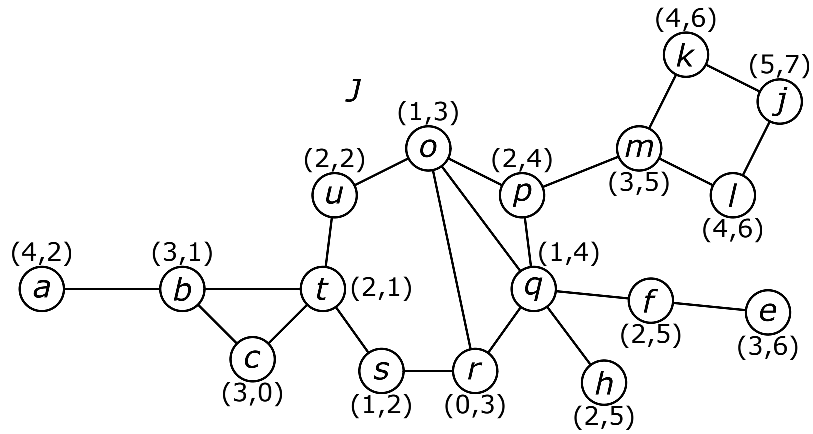


Abbildung 23: Der Graph J aus Abbildung 22. Jeder Knoten $x \in V(J)$ ist mit $(d_J(x,r)/d_J(x,c))$ markiert. Die Knoten a, c, e, h, j sind maximal entfernt von r (sowie jedem anderen Knoten aus $V(H)$), da sie maximal entfernt von ihrem jeweiligen Verbindungsknoten vor der Verknüpfung waren. Die Knoten e, h, j sind somit auch maximal entfernt von c (sowie jedem anderen Knoten aus $V(G_1 \setminus \{d\})$). Zusätzlich ist a maximal entfernt von c, da dies schon in G_1 der Fall war. Aus $V(H)$ ist kein Knoten maximal entfernt von c, da nur p, q maximal entfernt von t in H sind, diese jedoch Separationsknoten in J sind.

v_j ersetzt, welcher ein Separationsknoten in J und somit nicht maximal entfernt ist. Die übrigen Separationsknoten müssen ebenfalls aus der Menge der maximal entfernten Knoten entfernt werden. □

Die beiden vorherigen Lemmata sind ausreichend, um die Kantenmenge zwischen Knoten aus unterschiedlichen verknüpften Graphen im J_{SR} zu bestimmen, was in den folgenden beiden Lemmata festgehalten wird.

Lemma 10. *Sei Voraussetzung 1 gegeben. Dann gelten für jeden Knoten $u'_i \in V(G_i) \setminus \{u_i\}$ für ein i, $1 \leq i \leq k$ und jeden Knoten $v' \in V(J) \setminus V(G_i)$ die folgenden Aussagen.*

1. *Wenn u'_i maximal entfernt von u_i in G_i oder äquivalent maximal entfernt von v_i in J ist, dann ist u'_i maximal entfernt von v' in J.*

2. *Wenn v' maximal entfernt von v_i in J ist, dann ist v' maximal entfernt von u'_i in J.*

Beweis. Folgt aus Lemma 8 und Lemma 9. □

Lemma 11. *Sei Voraussetzung 1 gegeben. Zwei Knoten $u'_i \in V(G_i) \setminus \{u_i\}$ und $u'_j \in V(G_j) \setminus \{u_j\}$ für $i \neq j$ sind genau dann gegenseitig maximal entfernt in J, wenn u'_i maximal entfernt von u_i in G_i ist und u'_j maximal entfernt von u_j in G_j ist.*

Beweis. „⇒“ Folgt aus Lemma 9, da u'_i maximal entfernt von u'_j sein muss und vice versa.

„⇐“ Da jeder kürzeste Weg zwischen u'_i und u'_j in J über v_i und v_j verläuft folgt, dass u'_i maximal entfernt von u_i in G_i sein muss und u'_j maximal entfernt von u_j in G_j. □

Zuletzt gilt es, die Kantenmenge im J_{SR} zwischen Knoten aus demselben Teilgraphen zu bestimmen. Da durch die anderen Teilgraphen jedoch ausschließlich die Nachbarschaft der Separationsknoten beeinflusst wird, hat das Verknüpfen keinen Einfluss auf diese Kantenmenge.

Lemma 12. *Sei Voraussetzung 1 gegeben.*

1. *Zwei Knoten $u_i', u_i'' \in V(G_i) \setminus \{u_i\}$ sind genau dann gegenseitig maximal entfernt in J, wenn sie gegenseitig maximal entfernt in G_i sind.*

2. *Zwei Knoten $v', v'' \in V(H) \setminus \{v_1, \dots, v_k\}$ sind genau dann gegenseitig maximal entfernt in J, wenn sie gegenseitig maximal entfernt in H sind.*

Beweis. Die Aussage folgt daraus, dass kein kürzester Weg in J zwischen zwei Knoten in $(V(G_i) \cup \{v_i\}) \setminus \{u_i\}$ über einen Knoten außerhalb von $(V(G_i) \cup \{v_i\}) \setminus \{u_i\}$ und kein kürzester Weg in J zwischen zwei Knoten aus $V(H)$ über einen Knoten außerhalb von $V(H)$ führt. Separationsknoten sind von keinem Knoten maximal entfernt. □

Die drei vorherigen Aussagen werden in dem folgenden Theorem zusammengefasst, welches alle Kanten des J_{SR} beschreibt, siehe Abbildung 24. Im nächsten Schritt kann basierend auf diesen Informationen ein Vertex Cover für J_{SR} berechnet werden.

Theorem 9. *Sei Voraussetzung 1 gegeben. Der strong resolving Graph J_{SR} besitzt eine Kante $\{w_1, w_2\}$ genau dann wenn $w_1, w_2 \notin \{v_1, \dots, v_k\}$ und*

1. *$w_1, w_2 \in V(G_i)$ für ein i, $1 \leq i \leq k$, und w_1 und w_2 sind gegenseitig maximal entfernt in G_i,*

2. *$w_1, w_2 \in V(H)$ und w_1 und w_2 sind gegenseitig maximal entfernt in H,*

3. *$w_1 \in V(G_i)$ und $w_2 \in V(G_j)$ für i, j, $1 \leq i < j \leq k$, w_1 ist maximal entfernt von u_i in G_i und w_2 ist maximal entfernt von u_j in G_j, oder*

4. *wenn $w_1 \in V(G_i)$ für ein i, $1 \leq i \leq k$, $w_2 \in V(H)$, w_1 ist maximal entfernt von u_i in G_i und w_2 ist maximal entfernt von v_i in H.*

6.2.2 Berechnung eines minimalen Vertex Covers

Theorem 9 beschreibt die Kanten des J_{SR} wie folgt.

- Die Kanten in Punkt 1 sind die Kanten von $(G_i)_{\mathrm{SR}} \setminus \{u_i\}$.
- Die Kanten in Punkt 2 sind die Kanten von $H_{\mathrm{SR}} \setminus \{v_1, \dots, v_k\}$.

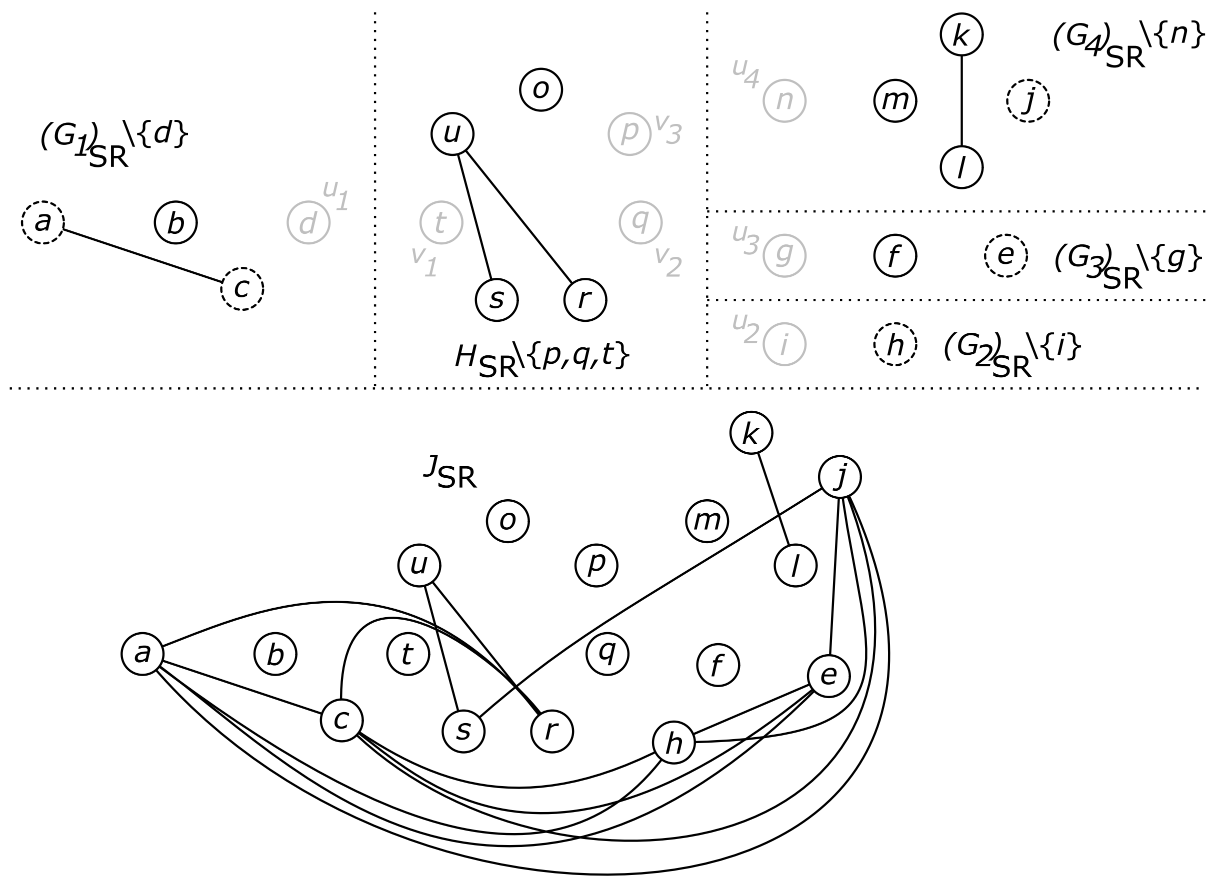


Abbildung 24: Die strong resolving Graphen der Graphen aus Abbildung 22 ohne ihre Verbindungsknoten sowie J_{SR}. Die gestrichelten Knoten sind die, die von den jeweiligen Verbindungsknoten maximal entfernt sind.

- Die Kanten in Punkt 3 sind die Kanten eines vollständigen k-partiten Teilgraphen $K_{n_1,\ldots,n_k}$, $n_i = |\mathrm{MD}(\,\mathrm{G_i},\, \mathrm{u_i}\,)|$ mit Knotenmenge

  $$\bigcup_{1\leq i\leq k} \mathrm{MD}(\,\mathrm{G_i},\, \mathrm{u_i}\,)$$

  und Kantenmenge

  $$\{\{w_1, w_2\} \,|\, w_1, \in \mathrm{MD}(\,\mathrm{G_i},\, \mathrm{u_i}\,), w_2, \in \mathrm{MD}(\,\mathrm{G_j},\, \mathrm{u_j}\,), 1 \leq i < j \leq k\}.$$

- Die Kanten in Punkt 4 sind die Kanten von k vollständig bipartiten Teilgraphen K_{n_i,m_i}, $1 \leq i \leq k$, $n_i = |\mathrm{MD}(\,\mathrm{G_i},\, \mathrm{u_i}\,)|$, $m_i = |\mathrm{MD}(\,\mathrm{H},\, \mathrm{v_i}\,) \setminus \{v_1, \ldots, v_k\}|$ mit Knotenmenge

  $$\mathrm{MD}(\,\mathrm{G_i},\, \mathrm{u_i}\,) \cup (\,\mathrm{MD}(\,\mathrm{H},\, \mathrm{v_i}\,) \setminus \{v_1, \ldots, v_k\}\,)$$

  und Kantenmenge

  $$\{\{w_1, w_2\} \,|\, w_1, \in \mathrm{MD}(\,\mathrm{G_i},\, \mathrm{u_i}\,), w_2, \in \mathrm{MD}(\,\mathrm{H},\, \mathrm{v_i}\,) \setminus \{v_1, \ldots, v_k\}\}.$$

Nach Theorem 6 ist eine stark trennende Menge für J äquivalent zu einem Vertex Cover für J_{SR} und nach Theorem 9 enthält J_{SR} den vollständig k-partiten Graphen $K_{n_1,\dots,n_k}$ und die k vollständig bipartiten Graphen K_{n_i,m_i}, $1 \leq i \leq k$ als Teilgraphen. Somit existiert für jedes Vertex Cover U für J_{SR} ein j, $1 \leq j \leq k$, sodass U alle Knoten der Mengen $\mathrm{MD}(\mathrm{G_i}, \mathrm{u_i})$, $1 \leq i \leq k$, $i \neq j$ enthält und zusätzlich alle Knoten aus $\mathrm{MD}(\mathrm{G_j}, \mathrm{u_j})$ oder alle Knoten aus $\mathrm{MD}(\mathrm{H}, \mathrm{v_j}) \setminus \{v_1, \dots, v_k\}$. Um diese verschiedenen Möglichkeiten für ein Vertex Cover festzuhalten, wird die folgende Notation verwendet.

Notation 1. *Für einen Graphen G sei*

$$\mathrm{VC}(\mathrm{G})$$

ein minimales Vertex Cover für G. Für einen Graphen G und eine Knotenmenge $M \subseteq V(G)$ sei

$$\overline{VC}(G, M)$$

ein Vertex Cover für G, das außerdem alle Knoten aus M enthält.

Ein minimales Vertex Cover für J_{SR} kann damit wie folgt bestimmt werden.

Lemma 13. *Sei Voraussetzung 1 gegeben. Sei*

$$U_0 = \begin{pmatrix} & \mathrm{VC}(\mathrm{H_{SR}} \setminus \{\mathrm{v_1}, \dots, \mathrm{v_k}\}) \\ \cup & \bigcup_{1 \leq i \leq k} \overline{VC}((G_i)_{\mathrm{SR}} \setminus \{u_i\}, \mathrm{MD}(\mathrm{G_i}, \mathrm{u_i})) \end{pmatrix}$$

und

$$U_j = \begin{pmatrix} & \overline{VC}(H_{\mathrm{SR}} \setminus \{v_1, \dots, v_k\}, \mathrm{MD}(\mathrm{H}, \mathrm{v_j}) \setminus \{v_1, \dots, v_k\}) \\ \cup & \mathrm{VC}((\mathrm{G_j})_{\mathrm{SR}} \setminus \{\mathrm{u_j}\}) \\ \cup & \bigcup_{1 \leq i \leq k, i \neq j} \overline{VC}((G_i)_{\mathrm{SR}} \setminus \{u_i\}, \mathrm{MD}(\mathrm{G_i}, \mathrm{u_i})) \end{pmatrix}$$

für $1 \leq j \leq k$.

Jede Menge $U_0, U_1, \dots, U_k$ ist ein Vertex Cover für J_{SR} und mindestens eine ist ein minimales Vertex Cover für J_{SR}.

Beweis. Wie in Theorem 9 erwähnt, enthält J_{SR} einen vollständig k-partiten Teilgraphen mit Knotenmenge

$$\mathrm{MD}(\mathrm{G_1}, \mathrm{u_1}) \cup \cdots \cup \mathrm{MD}(\mathrm{G_k}, \mathrm{u_k})$$

und k vollständig bipartite Teilgraphen mit Knotenmengen

$$\mathrm{MD}(\mathrm{G_i}, \mathrm{u_i}) \cup (\mathrm{MD}(\mathrm{H}, \mathrm{u_i}) \setminus \{u_1, \dots, u_k\})$$

für $1 \leq i \leq k$. Jedes Vertex Cover für J_{SR} muss demnach alle Knotenmengen $\mathrm{MD}(\mathrm{G_1}, \mathrm{u_1}), \dots, \mathrm{MD}(\mathrm{G_k}, \mathrm{u_k})$ bis auf eine $\mathrm{MD}(\mathrm{G_j}, \mathrm{u_j})$, $1 \leq j \leq k$, und zusätzlich alle Knoten aus $\mathrm{MD}(\mathrm{G_j}, \mathrm{u_j})$ oder alle Knoten aus $\mathrm{MD}(\mathrm{H}, \mathrm{v_j}) \setminus \{v_1, \dots, v_k\}$ enthalten, um den verbleibenden vollständig bipartiten Teilgraphen abzudecken. Zuletzt müssen die Kanten in J_{SR}, die nicht inzident zu Knoten der gewählten Mengen sind, abgedeckt werden. Die $k+1$ Möglichkeiten sind durch die verschiedenen Mengen $U_0, U_1, \dots, U_k$ abgedeckt. $\square$

Die folgenden beiden Lemmata betrachten den Fall, dass J einen ausgezeichneten Knoten w besitzt, über den J mit einem weiteren Graphen verknüpft werden kann, siehe Abbildung 25. Für diesen Fall wird ein minimales Vertex Cover für $J_{\mathrm{SR}} \setminus \{w\}$ und ein minimales Vertex Cover für $J_{\mathrm{SR}} \setminus \{w\}$, das alle Knoten aus $\mathrm{MD}(\mathrm{J},\, \mathrm{w})$ enthält, berechnet.

Lemma 14. *Sei Voraussetzung 1 gegeben und $w \in V(H)$. Sei*

$$U_0 = \left(\begin{array}{ll} & \mathrm{VC}(\, \mathrm{H_{SR}} \setminus \{\mathrm{v_1}, \ldots, \mathrm{v_k}, \mathrm{w}\}\,) \\ \cup & \bigcup_{1 \le i \le k} \overline{VC}(\,(G_i)_{\mathrm{SR}} \setminus \{u_i\},\, \mathrm{MD}(\, \mathrm{G_i},\, \mathrm{u_i}\,)\,) \end{array} \right)$$

und

$$U_j = \left(\begin{array}{ll} & \overline{VC}(\, H_{\mathrm{SR}} \setminus \{v_1, \ldots, v_k, w\},\, \mathrm{MD}(\, \mathrm{H},\, \mathrm{v_j}\,) \setminus \{v_1, \ldots, v_k, w\}\,) \\ \cup & \mathrm{VC}(\,(\mathrm{G_j})_{\mathrm{SR}} \setminus \{\mathrm{u_j}\}\,) \\ \cup & \bigcup_{1 \le i \le k, i \ne j} \overline{VC}(\,(G_i)_{\mathrm{SR}} \setminus \{u_i\},\, \mathrm{MD}(\, \mathrm{G_i},\, \mathrm{u_i}\,)\,) \end{array} \right)$$

für $1 \le j \le k$.

Jede Menge $U_0, U_1, \ldots, U_k$ ist ein Vertex Cover für $J_{\mathrm{SR}} \setminus \{w\}$ und mindestens eine ist ein minimales Vertex Cover für $J_{\mathrm{SR}} \setminus \{w\}$.

Beweis. Der einzige Unterschied zwischen Lemma 13 und Lemma 14 ist der Knoten w in H, der jedoch beim Verknüpfen von J mit einem anderen Graphen entfernt, bzw. durch einen Separationsknoten ersetzt wird. Die Korrektheit folgt aus dem Beweis von Lemma 13. $\square$

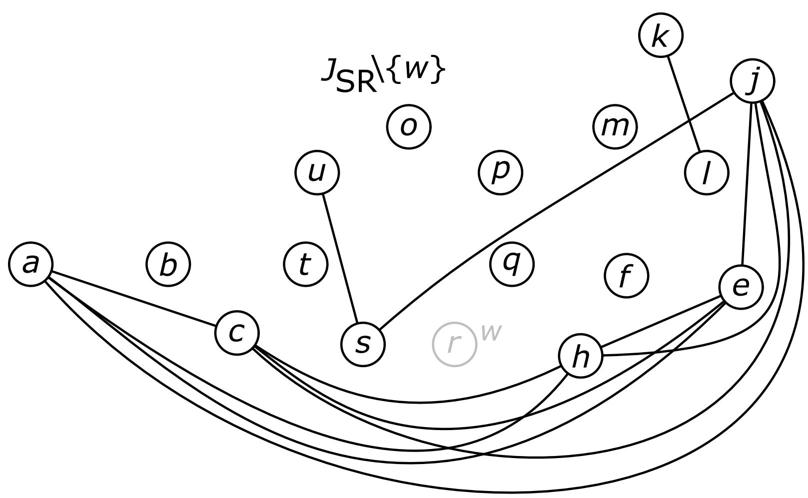


Abbildung 25: Der strong resolving Graphen J_{SR} ohne den Knoten $w = r$. Dieser Knoten wird bei den Berechnungen nicht berücksichtigt, wenn J über r mit einem weiteren Graphen verknüpft wird.

Lemma 15. *Sei Voraussetzung 1 gegeben und $w \in V(H)$. Dann ist*

$$\left(\begin{array}{ll} & \overline{VC}(\, H_{\mathrm{SR}} \setminus \{v_1, \ldots, v_k, w\},\, \mathrm{MD}(\, \mathrm{H},\, \mathrm{w}\,) \setminus \{v_1, \ldots, v_k, w\}\,) \\ \cup & \bigcup_{1 \le i \le k} \overline{VC}(\,(G_i)_{\mathrm{SR}} \setminus \{u_i\},\, \mathrm{MD}(\, \mathrm{G_i},\, \mathrm{u_i}\,)\,). \end{array} \right)$$

ein minimales Vertex Cover für $J_{\mathrm{SR}} \setminus \{w\}$, das alle Knoten aus $\mathrm{MD}(\, \mathrm{J},\, \mathrm{w}\,)$ *enthält.*

Beweis. Nach Lemma 8 enthält $\mathrm{MD}(\mathrm{J}, \mathrm{w})$ alle Knoten aus $\mathrm{MD}(\mathrm{G_i}, \mathrm{u_i})$, $1 \leq i \leq k$. Diese Knoten decken bereits alle in Theorem 9 beschriebenen Kanten des vollständig k-partiten Teilgraphen und der k vollständig bipariten Teilgraphen ab. Somit bilden die angegebenen Knotenmengen auch ein Vertex Cover für $J_{\mathrm{SR}} \setminus \{w\}$. □

Die vorherigen Lemmata über die Möglichkeiten eines Vertex Covers für J_{SR} sind in dem folgenden Theorem zusammengefasst.

Theorem 10. *Sei Voraussetzung 1 gegeben und $w \in V(H)$. Dann sind*

$$\mathrm{VC}(\mathrm{J_{SR}}),$$

$$\mathrm{VC}(\mathrm{J_{SR}} \setminus \{\mathrm{w}\}),$$

und

$$\overline{VC}(J_{\mathrm{SR}} \setminus \{w\}, \mathrm{MD}(\mathrm{J}, \mathrm{w}))$$

berechenbar mithilfe von $G_1, \ldots, G_k, H$, $u_1, \ldots, u_k$, $v_1, \ldots, v_k$, w, und den folgenden Knotenmengen.

1. $\mathrm{VC}((\mathrm{G_i})_{\mathrm{SR}} \setminus \{\mathrm{u_i}\})$, *für* $1 \leq i \leq k$,
   (für Lemma 13: U_j, für Lemma 14: U_j),
2. $\overline{VC}((G_i)_{\mathrm{SR}} \setminus \{u_i\}, \mathrm{MD}(\mathrm{G_i}, \{\mathrm{u_i}\}))$ *für* $1 \leq i \leq k$,
   (für Lemma 13: U_0, U_j, für Lemma 14: U_0, U_j, für Lemma 15),
3. (a) $\mathrm{VC}(\mathrm{H_{SR}} \setminus \{\mathrm{v_1}, \ldots, \mathrm{v_k}\})$,
      (für Lemma 13: U_0),
   (b) $\mathrm{VC}(\mathrm{H_{SR}} \setminus \{\mathrm{v_1}, \ldots, \mathrm{v_k}, \mathrm{w}\})$,
      (für Lemma 14: U_0),
4. (a) $\overline{VC}(H_{\mathrm{SR}} \setminus \{v_1, \ldots, v_k\}, \mathrm{MD}(\mathrm{H}, \{\mathrm{v_i}\}) \setminus \{v_1, \ldots, v_k\})$ *für* $1 \leq i \leq k$,
      (für Lemma 13: U_j),
   (b) $\overline{VC}(H_{\mathrm{SR}} \setminus \{v_1, \ldots, v_k, w\}, \mathrm{MD}(\mathrm{H}, \{\mathrm{v_i}\}) \setminus \{v_1, \ldots, v_k, w\})$ *für* $1 \leq i \leq k$,
      (für Lemma 14: U_j), und
5. $\overline{VC}(H_{\mathrm{SR}} \setminus \{v_1, \ldots, v_k, w\}, \mathrm{MD}(\mathrm{H}, \{\mathrm{w}\}) \setminus \{v_1, \ldots, v_k, w\})$,
   (für Lemma 15).

6.2.3 Algorithmus für verknüpfte Graphen

Basierend auf den vorangegangenen Ergebnissen dieses Kapitels kann nun die Starke Metrische Dimension verknüpfter Graphen berechnet werden. Da die Eingabe des Algorithmus hierfür aus nur einem Graphen G besteht, der durch Verknüpfung mehrerer Graphen entstanden ist, müssen zunächst die Graphen die verknüpft wurden bestimmt werden und anschließend für diese die in Theorem 10 benötigten Mengen berechnet werden. Da die Graphen immer über einen einzelnen Knoten verbunden werden, entsprechen die verknüpften

Graphen den zweifachen Zusammenhangskomponenten von G und die Knoten über die sie verbunden wurden den Separationsknoten. Eine Ausnahme bilden Brücken (d.h. Kanten bei denen beide Endknoten Separationsknoten oder Blätter sind), die zur Vereinfachung auch als zweifache Zusammenhangskomponenten behandelt werden.

Der Algorithmus zur Berechnung der starken Metrischen Dimension arbeitet wie folgt. Zunächst werden die zweifachen Zusammenhangskomponenten des gegebenen Graphen G sowie die Separationsknoten bestimmt. Die Struktur von G wird mithilfe eines Baums T dargestellt, der einen *b-Knoten* für jede zweifache Zusammenhangskomponente von G enthält und einen *s-Knoten* für jeden Separationsknoten. Ein b-Knoten und ein s-Knoten sind genau dann in T mit einer Kante verbunden, wenn der entsprechende Separationsknoten ein Teil der entsprechenden zweifachen Zusammenhangskomponente ist. T enthält keine weiteren Kanten. Anschließend wird ein beliebiger b-Knoten als Wurzel von T gewählt, sodass T von den Blättern zur Wurzel durchlaufen werden kann. Jeder Weg in T besteht hierbei abwechselnd aus b-Knoten und s-Knoten und jedes Blatt von T ist ein b-Knoten. Ein b-Knoten kann mehrere s-Knoten als Nachfolger besitzen, da wie in den vorherigen Kapiteln mehrere Graphen G_i an einen Graphen H geknüpft werden können und ein s-Knoten kann mehrere b-Knoten als Nachfolger besitzen, da die Knoten über die die Graphen verknüpft werden nicht disjunkt sein müssen, siehe Abbildung 26.

Sei b ein Blatt in T, s der Vorgänger von b, und b' der Vorgänger von s in T. Um Theorem 10 anzuwenden wird die mit b assoziierte zweifache Zusammenhangskomponente als Teilgraph G_i behandelt, der zu s gehörende Knoten als u_i und die Komponente von b' als Teilgraph H. Im nächsten Schritt wird der gesamte Graph, der durch die entsprechenden Komponenten von b' und seinen Nachfolgern entsteht, als Teilgraph $G_{i'}$ betrachtet und über den mit dem Vorgänger von b' assoziierten Knoten weiter verknüpft.

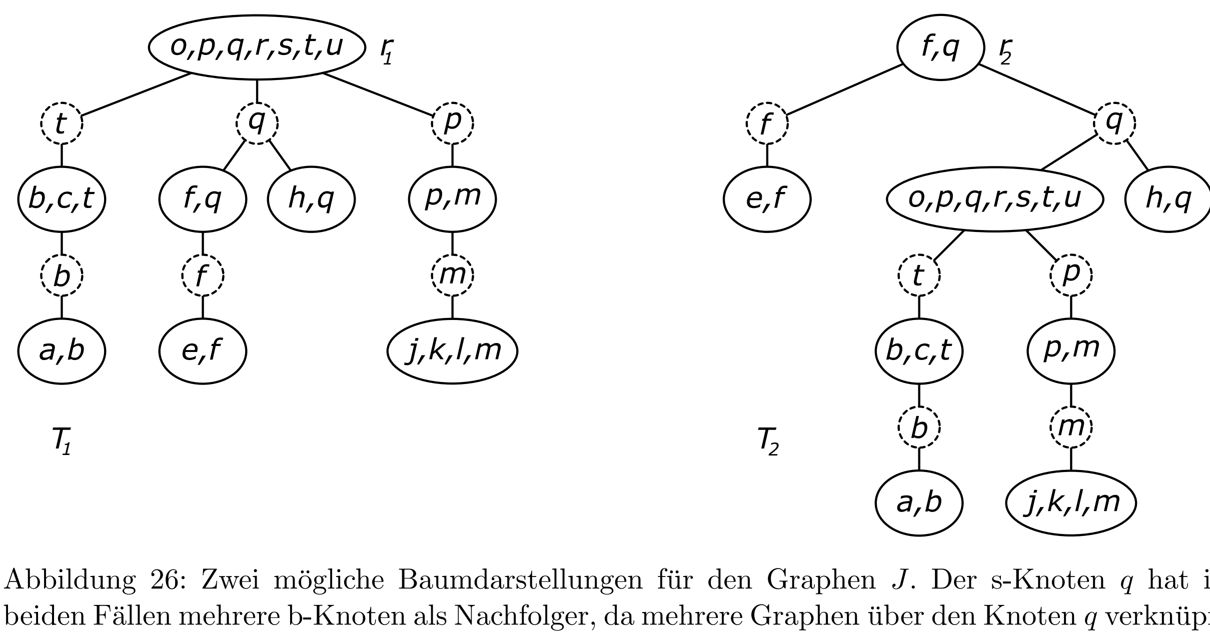


Abbildung 26: Zwei mögliche Baumdarstellungen für den Graphen J. Der s-Knoten q hat in beiden Fällen mehrere b-Knoten als Nachfolger, da mehrere Graphen über den Knoten q verknüpft werden.

6.2.4 Der Algorithmus am Beispiel dreier Graphklassen

Die in Theorem 10 benötigten Knotenmengen lassen sich in zwei Kategorien zusammenfassen. Die Starke Metrische Dimension eines Graphen H lässt sich in linearer Zeit berechnen, wenn für jede zweifache Zusammenhangskomponente G von H und jede Knotenmenge $W \subseteq V(G)$ die Mengen

1. $\mathrm{VC}(\, \mathrm{G}_{\mathrm{SR}} \setminus \mathrm{W}\,)$ und
2. $\overline{\mathrm{VC}}(\, G_{\mathrm{SR}} \setminus W,\ \mathrm{MD}(\, \mathrm{G},\ \{\mathrm{u}\}\,) \setminus W\,)$ für alle $u \in V(G)$

in linearer Zeit bestimmt werden können. Wichtig hierbei ist, dass die Mengen im zweiten Punkt nicht nur für jedes $u \in V(G)$, sondern für beliebig viele Knoten aus $V(G)$ zusammengenommen in linearer Zeit bestimmt werden können müssen. Diese Mengen werden in Punkt 4 von Theorem 10 verwendet und um eine lineare Laufzeit zu garantieren, darf jede zweifache Zusammenhangskomponente nur konstant oft durchlaufen werden, obwohl beliebig viele Graphen mit einer Komponente verknüpft werden können. Hierbei sei ebenfalls hervorgehoben, dass die Laufzeit linear in der Größe der betrachteten Komponente und nicht in der des gesamten Graphen sein muss.

In diesem Kapitel wird für einige Graphklassen gezeigt, wie die oben genannten Mengen in linearer Zeit berechnet werden können. Wie zuvor behandelt folgt daraus, dass die Starke Metrische Dimension eines Graphen H in linearer Zeit berechnet werden kann, wenn seine zweifachen Zusammenhangskomponenten diesen Graphklassen entsprechen.

Gitter

An dieser Stelle werden lediglich 2-dimensionale Gitter betrachtet; die Ergebnisse lassen sich jedoch einfach auf d-dimensionale Gitter mit $d > 2$ übertragen.

Ein 2-dimensionales $n \times m$ Gitter G besitzt die Knotenmenge

$$V(G) = \{x_{1,1}, \dots, x_{1,m}, \dots, x_{n,1}, \dots, x_{n,m}\}, \quad n, m \geq 2,$$

und die Kantenmenge

$$E(G) = \{\{x_{i,j}, x_{i',j'}\} \mid (i = i' \wedge |j - j'| = 1) \vee (|i - i'| = 1 \wedge j = j')\}.$$

Es ist leicht zu erkennen, dass der strong resolving Graph G_{SR} lediglich die beiden Kanten $\{x_{1,1}, x_{n,m}\}$ und $\{x_{1,m}, x_{n,1}\}$ enthält, also die Kanten zwischen je zwei gegenüberliegenden Eckknoten des Gitters. Die kürzesten Wege zwischen jedem anderen Knotenpaar u, v des Gitters können stets zu längeren kürzesten Wegen zwischen anderen Knoten erweitert werden, sodass weder u maximal entfernt von v ist, noch umgekehrt. Daraus folgt unmittelbar die nachstehende Beobachtung. Für eine Illustration der Distanzen in einem Gitter siehe Abbildung 27.

Beobachtung 2. *Sei G ein 2-dimensionales $n \times m$ Gitter und sei $x_{i,j} \in V(G)$.*

1. *Wenn $i \in \{1, n\}$ und $j \in \{1, m\}$, dann ist* $\mathrm{MD}(\mathrm{G}, \mathrm{x}_{\mathrm{i,j}}) = \{x_{n+1-i,m+1-j}\}$. *Das heißt, wenn $x_{i,j}$ ein Eckknoten des Gitters ist, dann enthält* $\mathrm{MD}(\mathrm{G}, \mathrm{x}_{\mathrm{i,j}})$ *nur den gegenüberliegenden Eckknoten.*

2. *Wenn $i \in \{1, n\}$ und $j \notin \{1, m\}$, dann ist* $\mathrm{MD}(\mathrm{G}, \mathrm{x}_{\mathrm{i,j}}) = \{x_{n+1-i,1}, x_{n+1-i,m}\}$. *Analog gilt, wenn $i \notin \{1, n\}$ und $j \in \{1, m\}$, dann ist* $\mathrm{MD}(\mathrm{G}, \mathrm{x}_{\mathrm{i,j}}) = \{x_{1,m+1-j}, x_{n,m+1-j}\}$. *Das heißt, wenn $x_{i,j}$ auf einer äußeren Kante des Gitters liegt, dann enthält* $\mathrm{MD}(\mathrm{G}, \mathrm{x}_{\mathrm{i,j}})$ *die beiden Eckknoten der gegenüberliegenden Kante.*

3. *Wenn $i \notin \{1, n\}$ und $j \notin \{1, m\}$, dann ist* $\mathrm{MD}(\mathrm{G}, \mathrm{x}_{\mathrm{i,j}}) = \{x_{1,1}, x_{n,1}, x_{1,m}, x_{n,m}\}$. *Das heißt, wenn $x_{i,j}$ ein innerer Knoten des Gitters ist, dann enthält* $\mathrm{MD}(\mathrm{G}, \mathrm{x}_{\mathrm{i,j}})$ *alle vier Eckknoten.*

Somit ist leicht zu erkennen, dass für ein 2-dimensionales Gitter G die Mengen $\mathrm{VC}(\mathrm{G}_{\mathrm{SR}} \setminus \mathrm{W})$ und $\overline{\mathrm{VC}}(G_{\mathrm{SR}} \setminus W, \mathrm{MD}(\mathrm{G}, \{\mathrm{u}\}) \setminus W)$ für alle $W \subseteq V(G)$ und beliebig viele $u \in V(G)$ insgesamt in linearer Zeit berechnet werden können. Wie angesprochen können diese Ergebnisse für mehrdimensionale Gitter erweitert werden, woraus das nächste Theorem folgt.

Theorem 11. *Die Starke Metrische Dimension eines Graphen G, dessen zweifache Zusammenhangskomponenten Gitter sind, kann in linearer Zeit berechnet werden.*

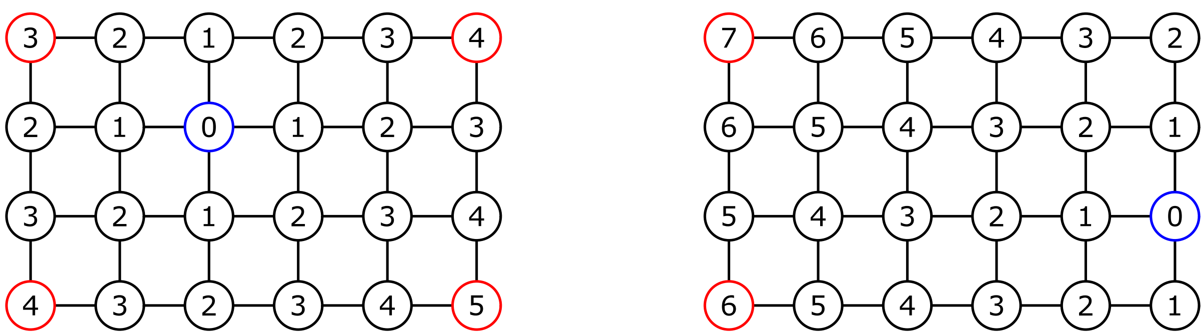


Abbildung 27: Ein Gitter $G_{4,6}$. Im linken Bild ist der Knoten $x_{2,3}$ (blau) markiert, sowie die von im maximal entfernten Knoten (rot). In den Knoten stehen die Distanzen zu dem Knoten $x_{2,3}$. Im rechten Bild analog für den Knoten $x_{3,6}$.

Kreise

Als zweites Beispiel werden Kreise betrachtet. Ein Kreis $G = C_n$ besitzt die Knotenmenge

$$V(G) = x_0, x_1, \ldots, x_{n-1}, \quad n \geq 3,$$

und die Kantenmenge

$$E(G) = \{\{x_0, x_1\}, \{x_1, x_2\}, \ldots, \{x_{n-1}, x_0\}\}.$$

Auch für Kreise ist der strong resolving Graph einfach zu bestimmen, da nur gegenüberliegende Knoten gegenseitig maximal entfernt sind. Hierbei ist lediglich zu unterscheiden, ob die Anzahl der Knoten im Kreis gerade oder ungerade ist. Bei geraden Kreisen liegt jedem Knoten genau ein anderer Knoten gegenüber, sodass der G_{SR} aus $n/2$ disjunkten K_2 besteht. Bei ungeraden Kreisen liegen jedem Knoten genau zwei andere Knoten gegenüber, sodass der G_{SR} selbst ebenfalls ein Kreis ist, siehe Abbildung 28. In beiden Fällen besteht $G_{\mathrm{SR}} \setminus W$ für jedes $W \subseteq V(G)$ aus einer Sammlung von Wegen (außer wenn n ungerade und $W = \emptyset$ ist), von denen jeweils jeder zweite Knoten in einem minimalen Vertex Cover enthalten ist. $\overline{\mathrm{VC}}(\,G_{\mathrm{SR}} \setminus W,\ \mathrm{MD}(\,\mathrm{G},\ \{\mathrm{x_i}\}\,) \setminus W\,)$ kann jedoch noch weitere Knoten enthalten, zum Beispiel wenn $\mathrm{MD}(\,\mathrm{G},\ \{\mathrm{x_i}\}\,)$ zwei Knoten enthält, die beide Endknoten desselben Wegs in $G_{\mathrm{SR}} \setminus W$ sind. Insbesondere kann dies dadurch auftreten, dass die Knoten aus $\mathrm{MD}(\,\mathrm{G},\ \{\mathrm{x_i}\}\,)$ immer Endknoten von Wegen in $G_{\mathrm{SR}} \setminus W$ sind, wenn $x_i \in W$.

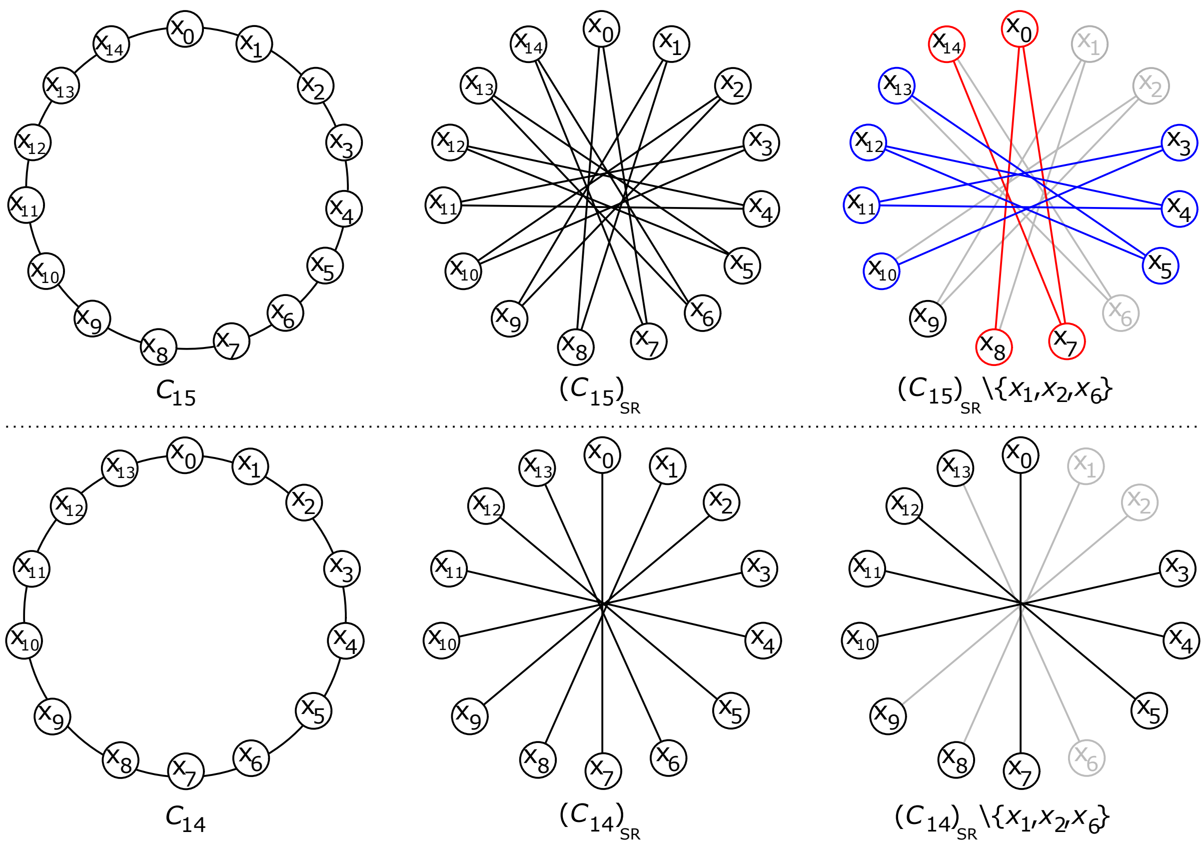


Abbildung 28: Die Graphen C_{15} und C_{14} mit ihren strong resolving Graphen sowie den strong resolving Graphen ohne die Knoten $\{x_1, x_2, x_6\}$. Der strong resolving Graph von C_{15} zerfällt ohne die Knoten in zwei Wege (rot und blau), der strong resolving Graph von C_{14} besteht aus disjunkten Kanten und einzelnen Knoten.

Zur Bestimmung der benötigten Vertex Cover werden lediglich die Wege und ihre Längen in $G_{\mathrm{SR}} \setminus W$ sowie die Positionen der Knoten aus $\mathrm{MD}(\,\mathrm{G},\ \{\mathrm{x_i}\}\,)$ für alle $x_i \in W$ auf den Wegen benötigt. Diese können mithilfe der Nummerierung der Knoten in G

in linearer Zeit errechnet werden, die, falls nicht bereits gegeben, ebenfalls in linearer Zeit erzeugt werden kann. Somit können auch für Kreise die Mengen $\mathrm{VC}(\mathrm{G}_{\mathrm{SR}} \setminus \mathrm{W})$ und $\overline{\mathrm{VC}}(G_{\mathrm{SR}} \setminus W, \mathrm{MD}(\mathrm{G}, \{\mathrm{u}\}) \setminus W)$ für alle $W \subseteq V(G)$ und beliebig viele $u \in V(G)$ insgesamt in linearer Zeit berechnet werden, woraus das folgende Theorem folgt.

Theorem 12. *Die Starke Metrische Dimension eines Graphen G, dessen zweifache Zusammenhangskomponenten Kreise sind, kann in linearer Zeit berechnet werden.*

Da die Bestimmung der benötigten Mengen und damit auch der Vertex Cover für vollständige Graphen trivial ist, deckt das Theorem beispielsweise Kaktus- und Kaktusblockgraphen ab.

Co-Graphen

Als letztes Beispiel werden Co-Graphen betrachtet, welche bereits in Kapitel 5 definiert wurden. Dort wurden ebenfalls Co-Bäume und kanonische Co-Bäume definiert und erwähnt, dass diese in linearer Zeit berechnet werden können. In diesem Kapitel werden ausschließlich kanonische Co-Bäume verwendet. Außerdem sind die Ergebnisse aus Kapitel 6 und Kapitel 6.1 entscheidend für das Beispiel dieser Graphklasse.

Zur Erinnerung entsprechen die Blätter des Co-Baums T für einen Co-Graphen G den Knoten aus $V(G)$ und die inneren Knoten von T sind mit $\cup$ oder $\times$ markiert. Im kanonischen Co-Baum können diese inneren Knoten beliebig viele Nachfolger haben. Zwei Knoten aus $V(G)$ sind genau dann in G benachbart, wenn ihr kleinster gemeinsamer Vorgänger in T mit $\times$ markiert ist.

Nach Korollar 1 ist der strong resolving Graph eines Co-Graphen ebenfalls ein Co-Graph, sodass für G_{SR} ebenfalls ein eindeutiger kanonischer Co-Baum T_{SR} existiert. Nach Lemma 5 entspricht der G_{SR} eines Co-Graphen G dem Komplement $\overline{G}$ mit zusätzlichen Kanten zwischen den true twins in G. Der Co-Baum T_{SR} kann ausgehend von T erzeugt werden, indem im ersten Schritt alle mit $\times$ markierten Knoten stattdessen mit $\cup$ markiert werden und vice versa. Im zweiten Schritt werden die Kanten zwischen den true twins von G betrachtet. Zwei Knoten $u, v \in V(G)$ sind genau dann true twins in G, wenn die entsprechenden Knoten in T Blätter am selben mit $\times$ markierten Knoten sind. Nach Vertauschen der Markierungen sind diese Knoten Blätter an einem $\cup$-Knoten, sodass u und v im entstehenden Co-Graphen nicht benachbart wären. Um dies zu ändern, werden die Blätter vom $\cup$-Knoten abgetrennt und erhalten einen neu hinzugefügten, mit $\times$ markierten Knoten als Vorgänger. Dieser $\times$-Knoten wird an den $\cup$-Knoten angehängt und als *twin-join* Knoten markiert. Hierdurch kann es sein, dass der $\cup$-Knoten lediglich diesen twin-join Knoten als Nachfolger hat, siehe Abbildung 29. Für die folgenden Algorithmen ist es jedoch notwendig, diese Struktur beizubehalten und den Co-Baum nicht zu vereinfachen. Die zusätzliche Markierung wird verwendet, da die twin-join Knoten zum Co-Baum hinzugefügt werden und keinem Knoten in T zugeordnet sind. Außerdem kann der $\cup$-Knoten nicht einfach zu einem $\times$-Knoten ummarkiert werden, da er außer den Blättern auch noch innere Knoten als Nachfolger besitzen kann.

Anschließend kann ein minimales Vertex Cover für G_{SR} mithilfe eines einfachen bottom-up Algorithmus, wie er in Algorithmus 2 gezeigt ist, in linearer Zeit berechnet werden.

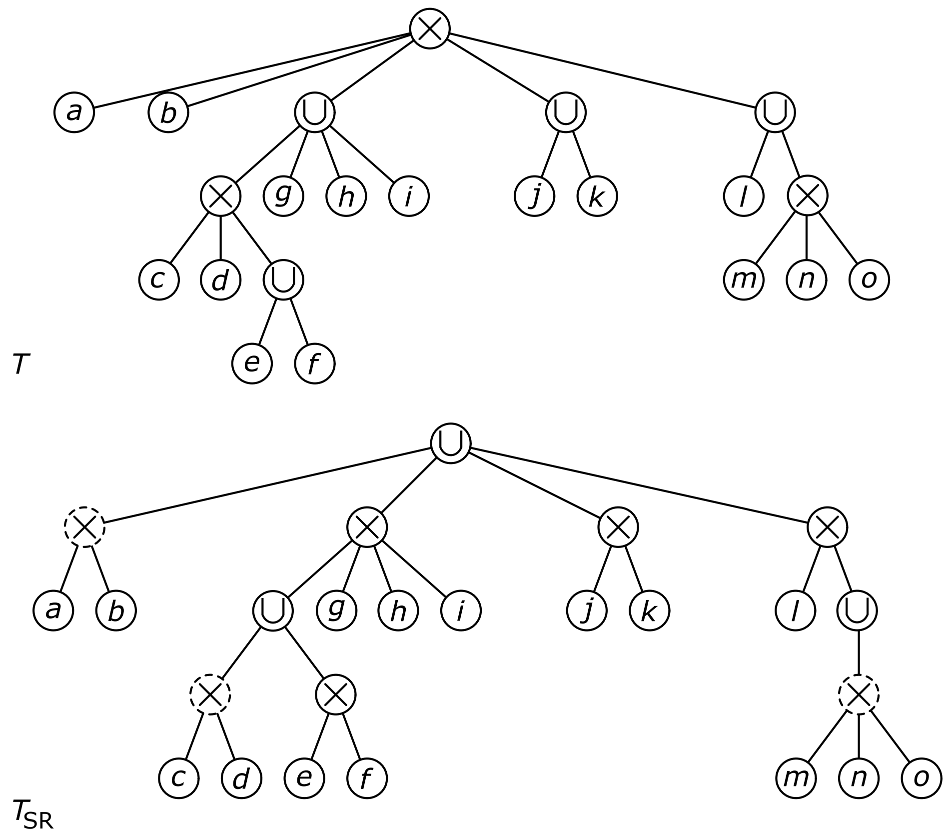


Abbildung 29: Der kanonische Co-Baum T eines Co-Graphen sowie der Co-Baum T_{SR} des strong resolving Graphen. Bis auf die twin-join Knoten entspricht der Baum T_{SR} dem Baum T mit vertauschten Markierungen an den inneren Knoten. In T_{SR} besitzt der $\cup$-Knoten unten rechts lediglich einen twin-join Knoten als Nachfolger, wodurch der Baum vereinfacht werden könnte; diese Struktur wird jedoch für nachfolgende Algorithmen beibehalten.
Betrachte den kanonischen Co-Baum T. Der Knoten e wird mit dem Knoten f vereint, sodass $d(e, f) = 2$. Alle anderen Knoten außer g, h, i werden mit e und f verbunden, sodass sie die Distanz 1 zu e besitzen, jedoch mit f einen Nachbarn haben, der weiter von e entfernt ist. Somit ist keiner dieser Knoten maximal entfernt von e. Die Knoten d, e, f sind maximal entfernt von Knoten c, da sie alle in dem Teilbaum von T enthalten sind, dessen Wurzel der Vorgänger von c ist. Somit ist c mit all diesen Knoten benachbart, wodurch sie keinen Nachbarn besitzen, der weiter von c entfernt ist als sie selbst. Bezüglich der Knoten außerhalb dieses Teilbaums besitzen alle Knoten dieselbe Nachbarschaft.

Hierbei bezeichnet $\mathrm{n}(u)$ die Anzahl der Blätter im Teilbaum T_u, also dem Teilbaum von T_{SR} mit Wurzel u, siehe Abbildung 30. Der Algorithmus kann zudem so angepasst werden, dass er ein minimales Vertex Cover für $G_{\mathrm{SR}} \setminus W$ berechnet, indem zum Beispiel der Wert $\mathrm{n}(u)$ als Anzahl der Blätter in T_u, die nicht in W sind, definiert wird, siehe Abbildung 31.

Somit bleibt zu zeigen, dass $\overline{\mathrm{VC}}(\, G_{\mathrm{SR}} \setminus W, \mathrm{MD}(\,\mathrm{G}, \{\mathrm{u}\}\,) \setminus W\,)$ für alle $W \subseteq V(G)$ und beliebig viele $u \in V(G)$ für Co-Graphen insgesamt in linearer Zeit berechnet werden können. Zunächst wird für einen einzelnen Knoten u betrachtet, welche Knoten maximal

Algorithm 2 Vertex Cover eines Co-Graphen

function VC(Co-Baum T_{SR})
  for all Knoten u von T_{SR} **do**
    $\mathrm{vc}(u) \leftarrow -1$ ▷ -1 steht für undefiniert
  for all Blätter u von T_{SR} **do**
    $\mathrm{vc}(u) \leftarrow 0$
  end
  while $\left(\begin{array}{l}\text{es existiert Knoten } u \text{ in } T_{\mathrm{SR}} \text{ mit Nachfolgern } u_1, \ldots, u_k \text{, sodass}\\ \mathrm{vc}(u) = -1 \text{ und } \mathrm{vc}(u_i) \geq 0, 1 \leq i \leq k\end{array}\right)$ **do**
    if u ist ein $\cup$-Knoten **then**
      $\mathrm{vc}(u) \leftarrow \mathrm{vc}(u_1) + \cdots + \mathrm{vc}(u_k)$
    else
      $\mathrm{vc}(u) \leftarrow \min_{1 \leq i \leq k} (\mathrm{n}(u) - \mathrm{n}(u_i) + \mathrm{vc}(u_i))$ ▷ u ist ein $\times$-Knoten
    end
  end
  return $\mathrm{vc}(r)$
end

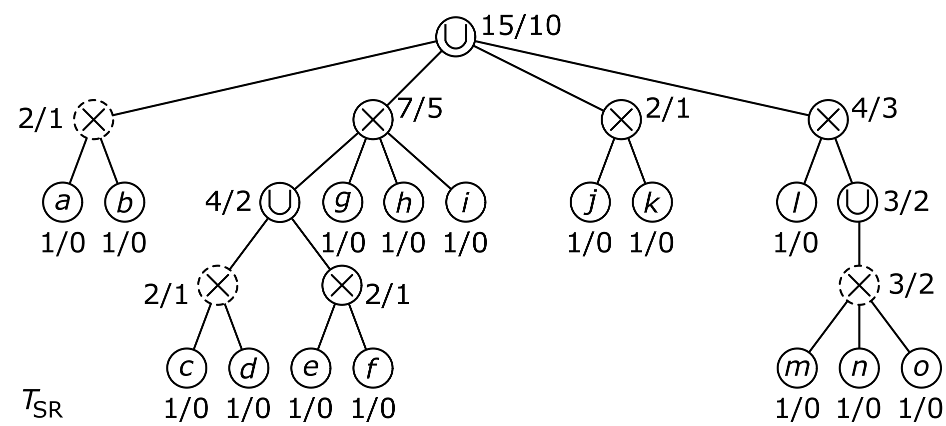


Abbildung 30: Die mit Algorithmus 2 berechnete Größe eines Vertex Covers mithilfe des Co-Baums T_{SR}. Die Knoten des Baums sind mit $\mathrm{n}(u)/\mathrm{vc}(u)$ beschriftet.

entfernt von u sind.

Da der Durchmesser von Co-Graphen maximal 2 ist, sind alle Knoten, die nicht mit u benachbart sind, maximal entfernt von diesem. Damit ein Nachbarknoten v von u maximal entfernt ist, muss per Definition gelten, dass $d_G(v', u) \leq d_G(v, u)$ für alle $v' \in N(v)$. Da $d_G(v, u) = 1$, müssen somit alle Nachbarn von v auch Nachbarn von u sein.

Im kanonischen Co-Baum T sind die von u maximal entfernten Knoten wie folgt zu finden. Betrachte die inneren Knoten auf dem Weg von u zur Wurzel von T. Für jeden $\cup$-Knoten wird u mit mindestens einem weiteren Knoten w vereint, aber ist nicht mit diesem benachbart. Jeder Knoten v, der danach aufgrund eines $\times$-Knotens mit u verbunden wird,

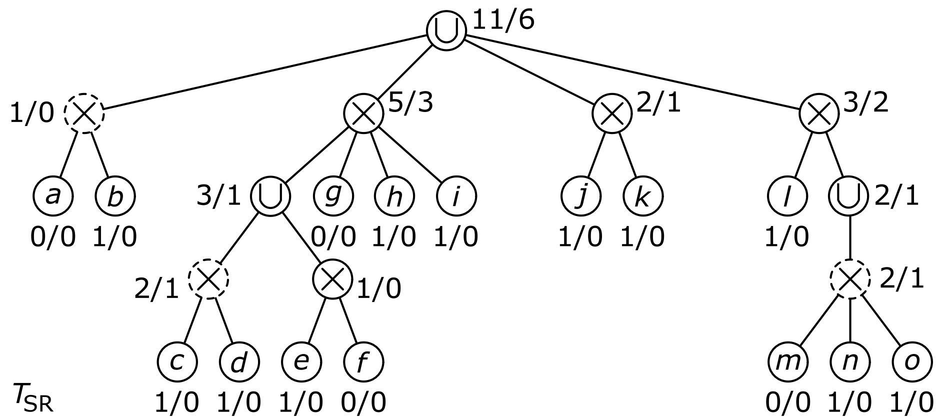


Abbildung 31: Die mit Algorithmus 2 berechnete Größe eines Vertex Covers mithilfe des Co-Baums T_{SR}, wobei die Knoten a, f, g, m nicht berücksichtigt werden, da sie für die Verknüpfung verwendet werden. Die Knoten des Baums sind wie in Abbildung 30 mit $\mathrm{n}(u)/\mathrm{vc}(u)$ beschriftet.

wird auch mit w verbunden und ist somit nicht maximal entfernt von u. Wird auf dem Weg von u zur Wurzel ein $\cup$-Knoten besucht, kommen folglich an den $\times$-Knoten keine von u maximal entfernten Knoten in Teilbäumen, die nicht u enthalten, vor. Demnach ist der direkte Vorgänger w von u in T der einzig relevante $\times$-Knoten, wenn er entsprechend markiert ist. Es gilt, dass u mit jedem Knoten v aus den Teilbäumen von w verbunden wird und anschließend alle Knoten im Teilbaum T_w dieselbe Nachbarschaft bezüglich aller anderen Knoten besitzen. Demnach ist die Nachbarschaft jedes Knotens in T_w eine Teilmenge der Nachbarschaft von u. Es sei hervorgehoben, dass bisher der Co-Baum von G und nicht der Co-Baum von G_{SR} verwendet wurde, sodass es keine $\cup$-Knoten gibt, die nur einen twin-join Knoten als Nachfolger besitzen.

Zusammengefasst gilt: Sei T der kanonische Co-Baum des Co-Graphen G. Seien u und v aus T und sei w der kleinste gemeinsame Vorgänger von u und v.

- Wenn w mit $\cup$ markiert ist, dann ist v maximal entfernt von u in G.
- Wenn w mit $\times$ markiert ist, dann ist v genau dann maximal entfernt von u in G, wenn u ein direkter Nachfolger von w ist.

Der folgende Algorithmus berechnet die Größe eines minimalen Vertex Cover für G_{SR}, das zusätzlich alle Knoten enthält, die maximal entfernt von einem Knoten $u \in V(G)$ sind. Hierbei sei darauf hingewiesen, dass Algorithmus 3 zu Beginn den Vorgänger von u in T statt in T_{SR} verwendet, um die twin-join Knoten nicht gesondert behandeln zu müssen. Die Knoten in T und T_{SR} werden zudem äquivalent verwendet und es wird davon ausgegangen, dass der Wert $\mathrm{vc}(w)$ bereits für jeden Knoten w in T_{SR} mithilfe von Algorithmus 2 berechnet wurde.

Die von Algorithmus 3 berechneten Werte $\mathrm{h}(v)$ entsprechen der Anzahl der von u maximal entfernten Knoten im Teilbaummit Wurzel v von T_{SR}. Besteht G nur aus dem Knoten u, so ist auch nur u von sich selbst maximal entfernt. Wenn G mindestens zwei

Algorithm 3 Vertex Cover eines Co-Graphen inklusive MD(G, u)

```
1: function VC̅(Co-Baum T, Co-Baum T_SR, Knoten u ∈ T)
2:     if (u = r) then
3:         return 1
4:     end
5:     sei v der Vorgänger von u in T
6:     h(v) ← n(v) − 1
7:     while (v ≠ r) do
8:         u ← v
9:         sei v der Vorgänger von u in T_SR
10:        seien u_1, ..., u_k die Nachfolger von v ohne u
11:        if v ist ein ×-Knoten in T_SR then
12:            h(v) ← h(u) + n(u_1) + ··· + n(u_k)
13:        else
14:            h(v) ← h(u) + vc(u_1) + ··· + vc(u_k)
15:        end
16:    end
17:    return h(r)
18: end
```

Knoten enthält und zusammenhängend ist, so ist u nicht maximal entfernt von sich selbst, weswegen im Algorithmus (Zeile 6) der Wert $\mathrm{h}(v)$ um eins verringert wird. Somit wird der Wert aufgrund der vorherigen Überlegungen zunächst unabhängig von der Markierung des ersten inneren Knotens auf $\mathrm{n}(v) - 1$ gesetzt. Anschließend wird er an jedem $\times$-Knoten in T_{SR} um die Anzahl der Knoten in den anderen Teilbäumen erhöht, da dieser $\times$-Knoten ein $\cup$-Knoten in T ist und die Knoten in diesen Teilbäumen wie beschrieben alle maximal entfernt von u sind. An jedem $\cup$-Knoten in T_{SR} wird der Wert um die Größen der Vertex Cover der anderen Teilbäume erhöht, wie es auch in Algorithmus 2 geschieht.

Da der Wert $\mathrm{n}(u)$ wie auch schon für Algorithmus 2 so angepasst werden kann, dass er der Anzahl der Blätter in T_u entspricht, die nicht Teil der Menge $W \subseteq V(G)$ sind, kann mit Algorithmus 3 die Größe einer Menge
$\overline{\mathrm{VC}}(\, G_{\mathrm{SR}} \setminus W, \mathrm{MD}(\,\mathrm{G}, \{\mathrm{u}\}\,) \setminus W\,)$ für alle $W \subseteq V(G)$ und ein $u \in V(G)$ in linearer Zeit berechnet werden. Folglich bleibt zu zeigen, wie die Größen dieser Mengen für beliebig viele $u \in V(G)$ insgesamt in linearer Zeit bestimmt werden können.

Hierfür wird in Algorithmus 4 der Co-Baum in einem top-down Verfahren durchlaufen und für jeden inneren Knoten u berechnet, um welchen Wert sich der h-Wert aus Algorithmus 3 für die Blätter an u auf dem Weg zur Wurzel erhöhen würde. Dieser Wert wird genauso berechnet wie die h-Werte; letztere werden lediglich auf den Wegen von der Wurzel zu den Blättern zu einem Wert $\mathrm{m}(u)$ aufsummiert, siehe Abbildung 32. Erneut kann der Wert $\mathrm{n}(u)$ an die Menge W angepasst werden und die Werte $\mathrm{vc}(u)$ können im Voraus durch Algorithmus 2 berechnet werden, siehe Abbildung 33. Anschließend kann wie folgt die Größe für $\overline{\mathrm{VC}}(\, G_{\mathrm{SR}} \setminus W,\, \mathrm{MD}(\,\mathrm{G},\, \{\mathrm{u}\}\,) \setminus W\,)$ für jeden Knoten $u \in V(G)$ in konstanter Zeit bestimmt werden. Sei $u \in T$ und sei v der Vorgänger von u in T. Dann ist

$$|\overline{\mathrm{VC}}(\, G_{\mathrm{SR}} \setminus W,\, \mathrm{MD}(\,\mathrm{G},\, \{\mathrm{u}\}\,) \setminus W\,)| = \mathrm{n}(v) - \mathrm{n}(u) + \mathrm{m}(v).$$

Hierbei ist $\mathrm{n}(u) = 0$, wenn $u \in W$, sodass beide Fälle ($u \in W$ und $u \notin W$) in dieser Formel zusammengefasst werden können. Außerdem sei darauf hingewiesen, dass die m-Werte für die Knoten in T und nicht in T_{SR} berechnet werden, sodass, obwohl die zusammengehörigen Knoten in beiden Bäumen äquivalent behandelt werden, die twin-join Knoten in T_{SR} keinen Wert zugewiesen bekommen.

Algorithm 4 Berechnung der m-Werte

function VC(Co-Baum T, Co-Baum T_{SR})
    for all innere Knoten u von T **do**
        $\mathrm{m}(u) \leftarrow -1$ ▷ -1 steht für undefiniert
    end
    $\mathrm{m}(r) \leftarrow 0$
    while $\left(\begin{array}{l} \text{es existiert innerer Knoten } u \text{ in } T_{\mathrm{SR}} \text{ mit Vorgänger } v \text{, sodass} \\ \mathrm{m}(u) = -1 \text{ und } \mathrm{m}(v) \geq 0 \end{array} \right)$ **do**
        seien $u_1, \ldots, u_k$ die Nachfolger von v in T_{SR} ohne u
        if v ist ein $\times$-Knoten in T_{SR} **then**
            $\mathrm{m}(u) \leftarrow \mathrm{m}(v) + \mathrm{n}(u_1) + \cdots + \mathrm{n}(u_k)$
        else
            $\mathrm{m}(u) \leftarrow \mathrm{m}(v) + \mathrm{vc}(u_1) + \cdots + \mathrm{vc}(u_k)$
        end
    end
end

Mithilfe von Algorithmus 4 können die Mengen $\overline{\mathrm{VC}}(\, G_{\mathrm{SR}} \setminus W,\, \mathrm{MD}(\,\mathrm{G},\, \{\mathrm{u}\}\,) \setminus W\,)$ für alle $W \subseteq V(G)$ und beliebig viele $u \in V(G)$ insgesamt in linearer Zeit berechnet werden, woraus das folgende Theorem folgt.

Theorem 13. *Die Starke Metrische Dimension eines Graphen G, dessen zweifache Zusammenhangskomponenten Co-Graphen sind, kann in linearer Zeit berechnet werden.*

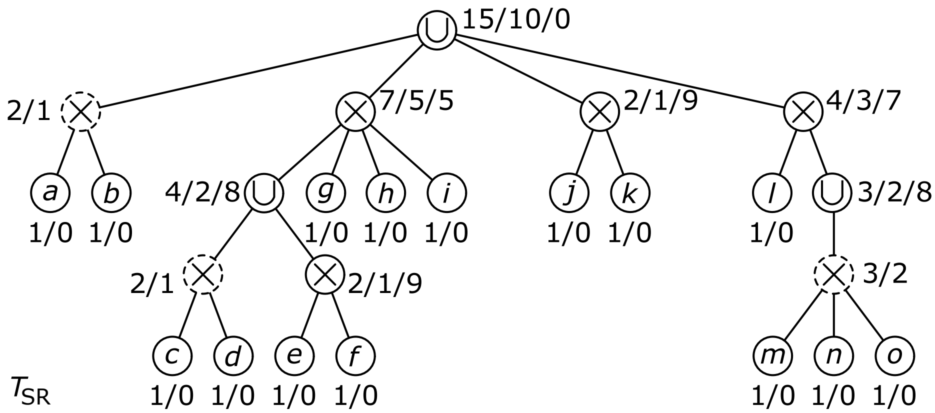


Abbildung 32: Die mit Algorithmus 4 berechnete Anzahl an maximal entfernten Knoten. Die Knoten des Baums sind mit $\mathrm{m}(u)/\mathrm{n}(u)/\mathrm{vc}(u)$ beschriftet. Die twin-join Knoten sind nur mit $\mathrm{n}(u)/\mathrm{vc}(u)$ beschriftet.

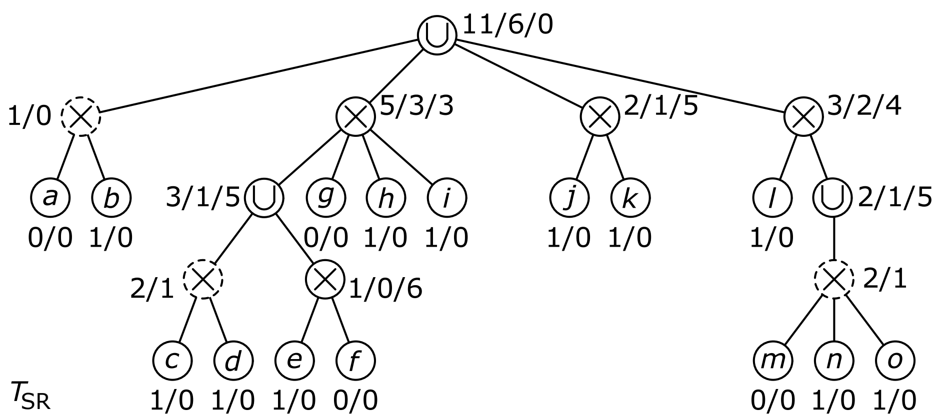


Abbildung 33: Derselbe Baum T_{SR} wie in Abbildung 32, wobei die Knoten a, f, g, m nicht berücksichtigt werden, da sie für die Verknüpfung verwendet werden.

6.3 Bestimmung des strong resolving Graphen distanzerhaltender Graphen

Distanzerhaltende Graphen wurden in [62] eingeführt. Für die Starke Metrische Dimension distanzerhaltender Graphen wurde in [93] ein Polyinomialzeit-Algorithmus angegeben. Als Grundlage für einen möglichen effizienteren Algorithmus, werden in diesem Kapitel die strong resolving Graphen distanzerhaltender Graphen untersucht. Gemäß Theorem 14 sind distanzerhaltende Graphen äquivalent zu Twinset-Graphen, die wie folgt definiert sind.

Definition 24. *[21]*

- *Der Graph mit nur einem Knoten v ist ein ungerichteter Twinset-Graph mit Twinset v, bezeichnet mit $G = (\{v\}, \emptyset, \{v\})$.*
- *Seien G_1 und G_2 zwei ungerichtete Twinset-Graphen. Die* Vereinigung $G = G_1 \cup$

G_2 dieser Graphen ist ein ungerichteter Twinset-Graph mit Knotenmenge $V(G_1) \cup V(G_2)$, Kantenmenge $E(G_1) \cup E(G_2)$ und Twinset $TS(G_1) \cup TS(G_2)$.

- *Seien G_1 und G_2 zwei ungerichtete Twinset-Graphen. Der* Join *$G = G_1 \times G_2$ dieser Graphen ist ein ungerichteter Twinset-Graph mit Knotenmenge $V(G_1) \cup V(G_2)$, Kantenmenge $E(G_1) \cup E(G_2) \cup \{\{u, v\} \mid u \in TS(G_1), v \in TS(G_2)\}$ und Twinset $TS(G_1) \cup TS(G_2)$.*

- *Seien G_1 und G_2 zwei ungerichtete Twinset-Graphen. Das* Anhängen *$G = G_1 \gg G_2$ dieser Graphen ist ein ungerichteter Twinset-Graph mit Knotenmenge $V(G_1) \cup V(G_2)$, Kantenmenge $E(G_1) \cup E(G_2) \cup \{\{u, v\} \mid u \in TS(G_1), v \in TS(G_2)\}$ und Twinset $TS(G_1)$.*

Ohne Beschränkung der Allgemeinheit wird davon ausgegangen, dass die letzte Operation bei einem zusammenhängenden Twinset-Graphen eine $\times$-Operation ist.

Theorem 14. *[7, 53] Ein Graph ist genau dann distanzerhaltend, wenn er ein Twinset-Graph ist.*

Definition 25. *Sei G ein Twinset-Graph. Die vier Mengen $A(G), B(G), B'(G)$ und $D(G)$ für G seien wie folgt definiert.*

- $G = (\{v\}, \emptyset, \{v\})$:
  - $A(G) = \{v\}$,
  - $B(G) = \emptyset$,
  - $B'(G) = \emptyset$ *und*
  - $D(G) = \{v\}$.
- $G = G_1 \cup G_2$:
  - $A(G) = A(G_1) \cup A(G_2)$,
  - $B(G) = \emptyset$,
  - $B'(G) = B(G_1) \cup B(G_2) \cup B'(G_1) \cup B'(G_2)$ *und*
  - $D(G) = \emptyset$.
- $G = G_1 \times G_2$:
  - $A(G) = A(G_1) \cup A(G_2)$,
  - $B(G) = B(G_1) \cup B(G_2)$,
  - $B'(G) = B'(G_1) \cup B'(G_2)$ *und*
  - $D(G) = D(G_1) \cup D(G_2)$.
- $G = G_1 \gg G_2$:

- $A(G) = \emptyset$,
- $B(G) = B(G_1) \cup B(G_2) \cup B'(G_2) \cup A(G_2)$,
- $B'(G) = B'(G_1)$ *und*
- $D(G) = \emptyset$.

Für eine einfachere Notation sei $R(G) = V(G) \setminus TS(G)$. Per Definition gilt $D(G) \subseteq A(G) \subseteq TS(G)$ und $B(G) \cup B'(G) \subseteq R(G)$.

Diese vier Mengen enthalten alle Knoten, die an den Kanten im strong resolving Graphen beteiligt sind, wie die folgenden Lemmata zeigen.

Lemma 16. *Sei G ein Twinset-Graph. Dann ist $A(G) = \{v \mid N[v] \subseteq TS(G)\}$ und $D(G) = \{v \mid N[v] = TS(G)\}$.*

Beweis. Beide Aussagen treffen offensichtlich zu, wenn G nur aus einem Knoten besteht. Es sei angemerkt, dass ein Knoten v mit $N[v] \subseteq TS(G)$ oder $N[v] = TS(G)$ nach den drei Operationen $\cup, \times, \gg$ diese Eigenschaft der Nachbarschaft behalten kann; ein solcher Knoten kann jedoch nicht durch die drei Operationen entstehen, da der Knoten selbst in $TS(G)$ sein muss. Außerdem ist ein Knoten, durch den die Nachbarschaft von v nicht diese Eigenschaft besitzt, entweder ein Nachbar von v, der Teil von $R(G)$ ist, welcher nach den drei Operationen bestehen bleibt und nicht mehr Teil des Twinsets werden kann. Oder (nur für den Fall, dass $v \in D(G)$) es ist ein Knoten aus $TS(G)$, der kein Nachbar von v ist, es auch nicht mehr werden kann und der nur (durch die $\gg$-Operation) aus dem Twinset entfernt wird, wenn v auch entfernt wird.

Sei $G = G_1 \cup G_2$. Die Knoten aus $A(G_1)$ und $A(G_2)$ behalten ihre Eigenschaft, da die Twinsets von G_1 und G_2 vereinigt werden. Da G_1 und G_2 beide aus mindestens einem Knoten bestehen und ihre Twinsets nicht leer sein können, existiert kein Knoten v mit $N[v] = TS(G)$ in G.

Sei $G = G_1 \times G_2$. Auch in diesem Fall behalten die Knoten aus $A(G_1)$ und $A(G_2)$ ihre Eigenschaft, die Knoten aus $D(G_1)$ und $D(G_2)$ jedoch auch, da jeder Knoten aus $TS(G_1)$ Nachbar von jedem Knoten aus $TS(G_2)$ wird.

Sei $G = G_1 \gg G_2$. Alle Knoten aus $A(G_1)$, $A(G_2)$, $D(G_1)$ und $D(G_2)$ verlieren ihre Eigenschaft, da sie entweder aus G_2 und somit nicht in $TS(G)$ sind oder sie bekommen alle Knoten aus $TS(G_2)$ als Nachbarn, welche anschließend in $R(G)$ sind. □

Lemma 17. *Sei G ein Twinset-Graph. Dann ist $B(G) \cup B'(G) = R(G) \cap \{v \mid d(v, TS(G)) \geq d(v', TS(G)), \forall v' \in N(v)\}$.*

Beweis. „$\subseteq$“: Die Mengen $B(G)$ und $B'(G)$ werden nur durch Knoten v aus $A(G_2)$ erweitert, wenn $G = G_1 \gg G_2$. Für diese Knoten gilt vor der Operation, dass $N[v] \subseteq TS(G_2)$ und somit $d(v, TS(G_2)) \geq d(v', TS(G_2)), \forall v' \in N(v)$. Nach der $\gg$-Operation gilt $d(w, TS(G)) = d(w, TS(G_2)) + 1$ für jeden Knoten $w \in V(G_2)$ und damit auch für v und seine Nachbarn. Außerdem gilt für jeden Nachbarn $v'' \in TS(G_1)$ von v, dass $1 = d(v, TS(G)) > d(v'', TS(G)) = 0$. Jede der drei Operationen behält diese Eigenschaften bezüglich der

Nachbarschaft von v bei, da die Operationen die kürzesten Wege aller Knoten zum Twinset entweder um 0 ($\cup$ und $\times$) oder um 1 ($\gg$) verlängern.

„$\supseteq$": Sei $v \in R(G) \cap \{v \mid d(v, TS(G)) \geq d(v', TS(G))$. Die Menge $R(G)$ wird nur durch das Twinset von G_2 erweitert, wenn $G = G_1 \gg G_2$. Wie oben beschrieben verlängern die drei Operationen die kürzesten Wege alle um 0 oder 1, womit v die Eigenschaft $d(v, TS(G)) \geq d(v', TS(G))$, $\forall v' \in N(v)$ bereits erfüllen muss, bevor er in $R(G)$ aufgenommen wird. Die Knoten des Twinsets, die diese Eigenschaft erfüllen, sind nach Lemma 16 genau die Knoten aus $A(G_2)$. Da jeder Knoten aus $A(G_2)$ in $B(G) \cup B'(G)$ aufgenommen wird, sobald er Teil von $R(G)$ wird, folgt die Aussage. □

Lemma 18. *Sei $G = G_1 \times (\{z\}, \emptyset, \{z\})$ ein Twinset-Graph und sei $v \in R(G_1)$. Dann ist v maximal entfernt von z genau dann, wenn $v \in B(G_1) \cup B'(G_1)$.*

Beweis. Diese Aussage folgt aus Lemma 17, da $d(z, v) = d(z, TS(G_1)) + d(TS(G_1), v) = 1 + d(TS(G_1), v)$ für alle $v \in V(G_1)$ und somit die Knoten, die maximal entfernt von z sind, genau die Knoten sind, die die Eigenschaft des vorherigen Lemmas erfüllen. □

Lemma 19. *Sei $G = G_1 \times (\{z\}, \emptyset, \{z\})$ ein Twinset-Graph und sei $v \in R(G_1)$. Dann sind v und z genau dann gegenseitig maximal entfernt, wenn $v \in B(G_1)$.*

Beweis. „$\Rightarrow$:" Sei $v \in R(G_1)$ und seien v und z gegenseitig maximal entfernt. Aus Lemma 18 folgt, dass $v \in B(G_1) \cup B'(G_1)$. Angenommen es sei $v \in B'(G_1)$. Da nur durch die $\cup$-Operation Knoten zu B' hinzugefügt werden, sei $G_1 = G' \cup G''$ für zwei Twinset-Graphen G' und G''. Dann existiert ein Knoten $w \in TS(G')$, sodass $d(w, u) = d(w, z) + d(z, u) = 1 + d(z, u)$ für alle Knoten $u \in V(G'')$ in G, wodurch z nicht maximal entfernt von Knoten in G'' ist. Dasselbe gilt für alle Knoten aus G'. Daher ist z nicht von v maximal entfernt, solange $w \in TS(G_1)$, was sich durch die $\cup$- und die $\times$-Operation nicht ändert. Lediglich durch die $\gg$-Operation wird w aus dem Twinset entfernt, wenn w ein Knoten des zweiten Graphen dieser Operation ist. In diesem Fall ist v jedoch auch ein Knoten des zweiten Graphen und wird somit aus $B'(G_1)$ entfernt.

„$\Leftarrow$:" Aus Lemma 18 folgt, dass v maximal entfernt von z ist. Da $N(z) = TS(G_1)$, ist z maximal entfernt von den Knoten in $B(G_1)$. Die Korrektheit gilt auch nach den anderen beiden Operationen, da Knoten nach einer $\cup$-Operation aus B entfernt und nach einer $\gg$-Operation zu B hinzugefügt werden. □

Die bisherigen Ergebnisse sind ausreichend, um die Kantenmenge des strong resolving Graphen zu bestimmen. Dies geschieht in den folgenden drei Lemmata.

Lemma 20. *Sei H ein zusammenhängender Twinset-Graph und sei $G = G_1 \cup G_2$ ein induzierter Teilgraph von H. Dann ist*

$$\begin{aligned} E(H_{\mathrm{SR}}[G]) = &\quad \{\{u, v\} \mid u \in A(G_1), v \in A(G_2) \cup B(G_2) \cup B'(G_2)\} \\ &\cup \{\{u, v\} \mid u \in B(G_1), v \in A(G_2) \cup B(G_2) \cup B'(G_2)\} \\ &\cup \{\{u, v\} \mid u \in B'(G_1), v \in A(G_2) \cup B(G_2) \cup B'(G_2)\}. \end{aligned}$$

Beweis. Da H zusammenhängend ist und da $TS(G) = TS(G_1) \cup TS(G_2)$, sind die Knoten aus G_1 und G_2 in H über einen Knoten verbunden, der durch eine $\times$- oder $\gg$-Operation hinzukam. Somit ist $d_H(u, v) = 2$ für alle $u \in TS(G_1)$ und $v \in TS(G_2)$. Zusammen mit Lemmata 16 bis 18 folgt die Aussage. Da $TS(G) = TS(G_1) \cup TS(G_2)$, werden beide Twinsets in allen nachfolgenden Operationen mit denselben Knoten verbunden, sodass keine der drei Operationen einen Einfluss auf die Kanten zwischen Knoten aus G_1 und G_2 in H_{SR} haben. $\square$

Lemma 21. *Sei H ein zusammenhängender Twinset-Graph und sei $G = G_1 \times G_2$ ein induzierter Teilgraph von H. Dann ist*

$$\begin{aligned} E(H_{\mathrm{SR}}[G]) = \quad & \{\{u, v\} \mid u \in A(G_1), v \in B(G_2\} \\ \cup \quad & \{\{u, v\} \mid u \in B(G_1), v \in A(G_2) \cup B(G_2) \cup B'(G_2)\} \\ \cup \quad & \{\{u, v\} \mid u \in B'(G_1), v \in B(G_2) \cup B'(G_2)\} \\ \cup \quad & \{\{u, v\} \mid u \in D(G_1), v \in D(G_2)\}. \end{aligned}$$

Beweis. Die Kanten zwischen den B- und B'-Mengen folgen erneut aus Lemma 18. Da jedoch $d_H(u, v) = 1$ für alle $u \in TS(G_1)$ und $v \in TS(G_2)$ und da $TS(G_2) \subseteq N_G(u)$ und $TS(G_1) \subseteq N_G(v)$, sind u und v nur dann in H_{SR} adjazent, wenn $N_G[u] = N_G[v] = TS(G)$. Auch hier gilt, dass die nachfolgenden Operationen die Kanten in H_{SR} nicht beeinflussen. $\square$

Lemma 22. *Sei H ein zusammenhängender Twinset-Graph und sei $G = G_1 \gg G_2$ ein induzierter Teilgraph von H. Dann ist*

$$\begin{aligned} E(H_{\mathrm{SR}}[G]) = \quad & \{\{u, v\} \mid u \in B(G_1), v \in A(G_2) \cup B(G_2) \cup B'(G_2)\} \\ \cup \quad & \{\{u, v\} \mid u \in B'(G_1), v \in B(G_2) \cup B'(G_2)\}. \end{aligned}$$

Beweis. Der Beweis verläuft analog zum vorherigen. Da jedoch davon ausgegangen wird, dass die letzte Operation bei zusammenhängenden Twinset-Graphen eine $\times$-Operation ist, besitzen die Knoten aus $TS(G_1)$ weitere Nachbarn, welche nicht in G und somit nicht adjazent zu den Knoten in $TS(G_2)$ sind. Dadurch ist keiner der Knoten aus $TS(G_1)$ maximal entfernt von Knoten aus G_2. $\square$

7 Linearzeitalgorithmus für die gerichtete Starke Metrische Dimension gerichteter Co-Graphen

In diesem Kapitel wird die gerichtete Starke Metrische Dimension gerichteter Co-Graphen berechnet. Hierbei werden ausschließlich stark zusammenhängende Graphen betrachtet, da diese Variante der Metrischen Dimension nur für solche definiert ist. Außerdem bilden die folgende Definition und das folgende Theorem, welche ebenfalls nur für stark zusammenhängende Graphen gelten, die Grundlage zur Untersuchung der gerichteten Starken Metrischen Dimension. Für ein Beispiel siehe Abbildung 34.

Definition 26. *[99] Sei G ein stark zusammenhängender gerichteter Graph und $u, v \in V(G)$.*

- *Der Knoten u ist* maximal entfernt zu *dem Knoten v (eng.: maximally distant to), abgekürzt $u \operatorname{MDT} v$, wenn $\forall u' \in N^-(u) : d(u', v) \leq d(u, v)$.*
- *Der Knoten v ist* maximal entfernt von *dem Knoten u (eng.: maximally distant from), abgekürzt $v \operatorname{MDF} u$, wenn $\forall v' \in N^+(v) : d(u, v') \leq d(u, v)$.*
- *Der Knoten u ist* gegenseitig maximal entfernt von *dem Knoten v (eng.: mutally maximally distant to), abgekürzt $u \operatorname{MMDT} v$, wenn $u \operatorname{MDT} v$ und $v \operatorname{MDF} u$.*
- *Der* strong resolving graph *G_{SR} von G ist ein ungerichteter Graph mit gleicher Knotenmenge wie G und enthält die Kante $\{u, v\}$ genau dann, wenn $u \operatorname{MMDT} v$ oder $v \operatorname{MMDT} u$.*

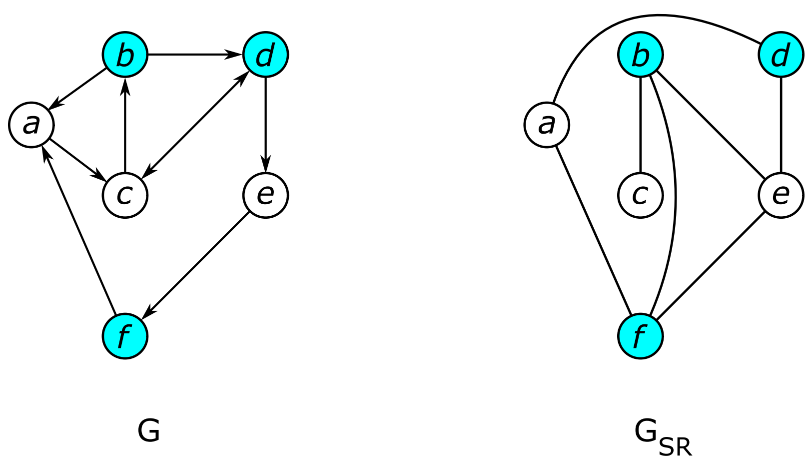


Abbildung 34: Der Graph aus Abbildung 11 (links) und sein strong resolving Graph (rechts). Die blau markierten Knoten bilden ein Vertex Cover in G_{SR} und sind somit eine stark trennende Menge für G. Beispielsweise ist der Knoten d gegenseitig maximal entfernt von a, jedoch nicht umgekehrt. Denn a ist aufgrund von f nicht maximal entfernt zu d und d ist aufgrund von e nicht maximal entfernt von a.

Theorem 15. *[99] Sei G ein stark zusammenhängender gerichteter Graph, dann ist* $\mathrm{smd}(G) = \mathrm{VC}(\mathrm{G}_{\mathrm{SR}})$*, wobei* $\mathrm{VC}(\mathrm{G}_{\mathrm{SR}})$ *die Größe eines minimalen Vertex Cover von* G_{SR} *ist.*

In diesem Kapitel wird die Starke Metrische Dimension von gerichteten Co-Graphen, wie sie in Kapitel 5.3 definiert sind, betrachtet. Es werden ausschließlich stark zusammenhängende Graphen betrachtet, sodass im Fall von gerichteten Co-Graphen die letzte Operation ein Join war. Das heißt die Wurzel jedes Co-Baums ist mit $\times$ markiert. Hierfür wird der gleiche Ansatz wie in Kapitel 6.1 verwendet, bei dem die Größe einer maximalen Clique im Komplement des strong resolving Graphen bestimmt wird. Die Methode aus Kapitel 6.2.4 lässt sich nicht übernehmen, da der strong resolving Graph von gerichteten Co-Graphen kein Co-Graph sein muss, siehe Abbildung 35.

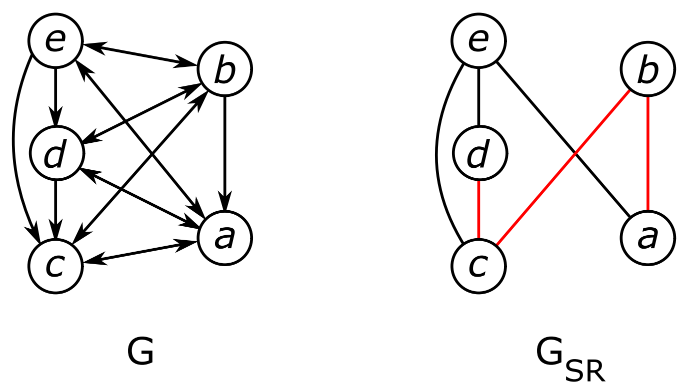


Abbildung 35: Ein gerichteter Co-Graph $G = (e \gg (d \gg c) \times (b \gg a))$ und sein strong resolving Graph. Letzterer enthält den induzierten Weg $\{a, b, c, d\}$ (rot) und ist somit kein Co-Graph.

Für den übrigen Teil dieses Kapitels sei G ein stark zusammenhängender Co-Graph mit binärem Co-Baum T. Sei w ein Knoten aus T und T_w der Teilbaum von T mit Wurzel w. Seien T_l und T_r der linke und rechte Teilbaum von T_w. Seien G_w, G_l und G_r die Teilgraphen von G, die durch die mit den Blättern in T_w, T_l und T_r assoziierten Knoten induziert werden.

Offensichtlich gilt auch für gerichtete Graphen, dass zwei Knoten im strong resolving Graph adjazent sind, wenn die Distanz zwischen ihnen dem Durchmesser des Graphen entspricht. Der Unterschied zum ungerichteten Fall besteht lediglich darin, dass in gerichteten Graphen für jedes Knotenpaar zwei Distanzen betrachtet werden müssen.

Da der Durchmesser von stark zusammenhängenden gerichteten Co-Graphen ≤ 2 ist, ist jedes Knotenpaar, zwischen dem es nicht beide gerichteten Kanten gibt, im strong resolving Graph adjazent. Diese Aussage wird im folgenden Lemma festgehalten, wobei in Hinblick auf den Algorithmus zur Bestimmung der Starken Metrischen Dimension das Komplement des strong resolving Graphen verwendet wird.

Lemma 23. *Wenn w mit $\cup$ oder $\gg$ markiert ist, dann existieren in $\overline{G_{\mathrm{SR}}}$ keine Kanten zwischen Knoten aus G_l und Knoten aus G_r.*

Beweis. Sei $u \in V(G_l)$ und $v \in V(G_r)$. Da w mit $\cup$ oder $\gg$ markiert ist, gilt $d_G(v, u) = 2$. Da G ein stark zusammenhängender gerichteter Co-Graph ist, ist $D(G) \leq 2$ und es folgt, dass $v \,\mathrm{MMDT}\, u$, was bereits als Beweis ausreichend ist. Wenn w mit $\cup$ markiert ist, gilt auch $d_G(u,v) = 2$ und somit $u \,\mathrm{MMDT}\, v$. Zudem sei angemerkt, dass wenn w mit $\gg$ markiert ist, dieser gerichtete Join auch eine Kante von jedem in-Nachbarn von u zu v erzeugt und eine Kante von u zu jedem out-Nachbarn von v. Da u und v dieselbe Nachbarschaft bezüglich der Knoten außerhalb von T_w besitzen kann folglich der kürzeste Weg von u nach v in keine Richtung erweitert werden, wodurch $u \,\mathrm{MMDT}\, v$ gilt. $\square$

Betrachte nun den Fall, dass w mit $\times$ markiert ist. Per Definition enthält G_{SR} genau dann eine Kante zwischen $u \in V(G_l)$ und $v \in V(G_r)$, wenn $u \,\mathrm{MMDT}\, v$ oder $v \,\mathrm{MMDT}\, u$. Da hier das Komplement $\overline{G_{\mathrm{SR}}}$ betrachtet wird, darf keine der beiden Eigenschaften erfüllt sein, damit die Kante zwischen u und v existiert. Somit enthält $\overline{G_{\mathrm{SR}}}$ genau dann die Kante $\{u, v\}$, wenn

$$\neg((u \,\mathrm{MMDT}\, v) \vee (v \,\mathrm{MMDT}\, u))$$

oder umgeformt

$$\neg(u \,\mathrm{MMDT}\, v) \wedge \neg(v \,\mathrm{MMDT}\, u)$$

oder umgeformt

$$(\neg(u \,\mathrm{MDT}\, v) \vee \neg(v \,\mathrm{MDF}\, u)) \wedge (\neg(v \,\mathrm{MDT}\, u) \vee \neg(u \,\mathrm{MDF}\, v))$$

oder umgeformt mithilfe der Definition von MDT und MDF

1. (a) $\neg(\forall y \in N^-(u) : d(y,v) \leq d(u,v))$ oder
   (b) $\neg(\forall x \in N^+(v) : d(u,x) \leq d(u,v))$

   und

2. (a) $\neg(\forall x \in N^-(v) : d(x,u) \leq d(v,u))$ oder
   (b) $\neg(\forall y \in N^+(u) : d(v,y) \leq d(v,u))$,

oder, da bei einem Join $d_G(u,v) = 1$ gilt

1. (a) $\exists y \in V(G)$, sodass $(y,u) \in E(G)$ und $(y,v) \notin E(G)$ oder
   (b) $\exists x \in V(G)$, sodass $(v,x) \in E(G)$ und $(u,x) \notin E(G)$

   und

2. (a) $\exists x \in V(G)$, sodass $(x,v) \in E(G)$ und $(x,u) \notin E(G)$ oder
   (b) $\exists y \in V(G)$, sodass $(u,y) \in E(G)$ und $(v,y) \notin E(G)$.

Da bei einem Join alle Kanten zwischen den Knoten aus G_l und G_r existieren, muss in den Fällen 1.a und 2.b der Knoten y Teil von G_r und der Knoten x in den Fällen 1.b und 2.a Teil von G_l sein.

Wie in den oben beschriebenen Fällen zu erkennen ist, reicht es aus, wenn der kürzeste Weg von u nach v in eine der beiden Richtungen erweitert werden kann (entweder durch einen in-Nachbarn von u oder einen out-Nachbarn von v), damit nicht $u \,\mathrm{MMDT}\, v$ gilt. Da auch nicht $v\,\mathrm{MMDT}\,u$, gilt eine analoge Aussage für den kürzesten Weg von v nach u. Als nächstes werden spezielle Arten von Knoten definiert, die eine Nachbarschaft besitzen, die das Erweitern dieser kürzesten Wege ermöglicht. Somit sind diese Knoten diejenigen, die für maximale Cliquen in $\overline{G_{\mathrm{SR}}}$ in Frage kommen.

Definition 27. *Sei G ein stark zusammenhängender gerichteter Co-Graph. Ein Knoten $u \in V(G)$ heißt*

- solitary-Knoten *in G, wenn ein Knoten $z \in V(G)$ existiert, sodass $(u,z) \notin E(G)$ und $(z,u) \notin E(G)$, das heißt, wenn z und u nicht benachbart sind.*
- in-Knoten *in G, wenn ein Knoten $z \in V(G)$ existiert, sodass $(z,u) \in E(G)$ und $(u,z) \notin E(G)$, das heißt, wenn z nur ein in-Nachbar von u ist.*
- out-Knoten *in G, wenn ein Knoten $z \in V(G)$ existiert, sodass $(u,z) \in E(G)$ und $(z,u) \notin E(G)$, das heißt, wenn z nur ein out-Nachbar von u ist.*
- in-out-Knoten *in G, wenn u ein in-Knoten und ein out-Knoten in G ist.*

Ein Knoten u kann gleichzeitig ein solitary-Knoten, ein in-Knoten, ein out-Knoten und ein in-out-Knoten in G sein. Wenn u ein in-out-Knoten ist, dann ist u auch ein in-Knoten und ein out-Knoten.

Lemma 24. *Wenn w mit $\times$ markiert ist, dann sind zwei Knoten $u \in V(G_l)$ und $v \in V(G_r)$ genau dann in $\overline{G_{\mathrm{SR}}}$ mit einer Kante verbunden, wenn*

1. *u ein solitary-Knoten in G_l ist oder*
2. *v ein solitary-Knoten in G_r ist oder*
3. *u ein in-out-Knoten in G_l ist oder*
4. *v ein in-out-Knoten in G_r ist oder*
5. *u ein in-Knoten in G_l und v ein in-Knoten in G_r ist oder*
6. *u ein out-Knoten in G_l und v ein out-Knoten in G_r ist.*

Beweis. „$\Rightarrow$:“

1. Wenn die Fälle 1.a und 2.a erfüllt sind, dann sind u und v solitary-, out- oder in-out-Knoten in G. Beide Knoten können zu unterschiedlichen Typen und zu mehreren gleichzeitig gehören.
2. Wenn die Fälle 1.a und 2.b erfüllt sind, dann ist v ein solitary-Knoten oder ein in-out-Knoten in G.
3. Wenn die Fälle 1.b und 2.a erfüllt sind, dann ist u ein solitary-Knoten oder ein in-out-Knoten in G.

4. Wenn die Fälle 1.b und 2.b erfüllt sind, dann sind u und v solitary-, in- oder in-out-Knoten in G. Beide Knoten können zu unterschiedlichen Typen und zu mehreren gleichzeitig gehören.

„$\Leftarrow$:“

1. Wenn u ein solitary-Knoten ist, dann sind die Fälle 1.b und 2.a erfüllt.
2. Wenn v ein solitary-Knoten ist, dann sind die Fälle 1.a und 2.b erfüllt.
3. Wenn u ein in-out-Knoten ist, dann sind die Fälle 1.b und 2.a erfüllt.
4. Wenn v ein in-out-Knoten ist, dann sind die Fälle 1.a und 2.b erfüllt.
5. Wenn u ein in-Knoten ist, dann ist der Fall 1.b und wenn v ein in-Knoten ist, der Fall 2.b erfüllt.
6. Wenn u ein out-Knoten ist, dann ist der Fall 1.a und wenn v ein out-Knoten ist, der Fall 2.a erfüllt.

□

Das vorherige Lemma gilt für alle Knoten, die paarweise eine der geforderten Eigenschaften erfüllen. So sind beispielsweise alle solitary-Knoten aus G_l mit allen Knoten aus G_r in $\overline{G_{\text{SR}}}$ verbunden und alle in-Knoten aus G_l mit allen in-Knoten aus G_r. Damit diese Knoten eine Clique in $\overline{G_{\text{SR}}}$ bilden, müssen jedoch auch die Knoten aus G_l sowie die aus G_r jeweils untereinander adjazent sein. Daher werden im Folgen bottom-up Algorithmus für jeden Knoten des Co-Baums T die Größen mehrerer Cliquen bestimmt. Für jeden Knoten w aus T geschieht dies mit den vier Werten m, s, i, o, wobei

1. m die Größe einer größten Clique in $\overline{G_{\text{SR}}}|_{V(G_w)}$ ist,
2. s die Größe einer größten Clique in $\overline{G_{\text{SR}}}|_{V(G_w)}$ ist, die ausschließlich aus solitary- oder in-out-Knoten besteht,
3. i die Größe einer größten Clique in $\overline{G_{\text{SR}}}|_{V(G_w)}$ ist, die ausschließlich aus in-Knoten besteht und
4. o die Größe einer größten Clique in $\overline{G_{\text{SR}}}|_{V(G_w)}$ ist, die ausschließlich aus out-Knoten besteht.

Entsprechend der vorherigen Lemmata können diese vier Werte aus den jeweiligen Werten m_l, s_l, i_l, o_l und m_r, s_r, i_r, o_r des linken und rechten Teilbaums T_l und T_r berechnet werden. Gleichzeitig ist es möglich, die Knotenmengen der dazugehörigen Cliquen mitzuführen, sodass auch eine minimale stark trennende Menge bestimmt werden kann.

Wie die Werte der Teilbäume kombiniert werden, ist von der Markierung von w abhängig. Jedoch sei angemerkt, dass m immer mindestens genauso groß ist wie das Maximum der anderen drei Werte, sodass einige Kombinationsmöglichkeiten nicht betrachtet werden müssen.

Wenn w ein Blatt ist, dann ist $m = 1, s = 0, i = 0, o = 0$.

Wenn w mit $\cup$ markiert ist, existieren gemäß Lemma 23 keine Kanten zwischen Knoten aus G_l und G_r in $\overline{G_{\text{SR}}}$. Somit können die Cliquen der beiden Teilgraphen nicht zu größeren Cliquen kombiniert werden. Da sowohl $V(G_l)$ als auch $V(G_r)$ mindestens einen Knoten enthalten, werden jedoch alle Knoten zu solitary-Knoten. Folglich ist

1. m das Maximum von m_l und m_r,
2. s das Maximum von m_l und m_r,
3. i das Maximum von i_l und i_r und
4. o das Maximum von o_l und o_r.

Wenn w mit $\gg$ markiert ist, existieren ebenfalls gemäß Lemma 23 keine Kanten zwischen Knoten aus G_l und G_r in $\overline{G_{\text{SR}}}$. Die Cliquen können nicht kombiniert werden, jedoch wird jeder Knoten aus G_l durch die Knoten aus G_r zu einem out-Knoten in G_w und jeder Knoten aus G_r durch die Knoten aus G_l zu einem in-Knoten in G_w. Das heißt auch, dass alle in-Knoten aus G_l und alle out-Knoten aus G_r zu in-out-Knoten werden. Folglich ist

1. m das Maximum von m_l und m_r,
2. s das Maximum von s_l, s_r, i_l und o_r,
3. i das Maximum von m_r und i_l und
4. o das Maximum von m_l und o_r.

Wenn w mit $\times$ markiert ist, so können gemäß Lemma 24 die Cliquen kombiniert werden, deren Knoten paarweise einen der sechs Fälle erfüllen. Durch den Join verlieren die speziellen Knoten keinen ihrer Typen, erhalten jedoch auch keine weiteren. Folglich ist

1. m das Maximum von $m_l + s_r, m_r + s_l, i_l + i_r$ und $o_l + o_r$,
2. $s = s_l + s_r$,
3. $i = i_l + i_r$ und
4. $o = o_l + o_r$.

Der bottom-up Prozess ist in Algorithmus 5 dargestellt. Für ein Beispiel siehe Abbildung 36.

Da der Co-Baum in linearer Zeit gefunden werden kann und nur einmal durchlaufen werden muss, folgt das nächste Theorem.

Theorem 16. *Die Gerichtete Starke Metrische Dimension eines stark zusammenhängenden Co-Graphens kann in linearer Zeit berechnet werden.*

Algorithm 5 Gerichtete Starke Metrische Dimension gerichteter Co-Graphen

function DSMD_DI_CO-GRAPH(gerichteter Co-graph G)
  $T \leftarrow$ binärer Co-Baum von G
  $n \leftarrow |V(G)|$
  $(m, s, i, o) \leftarrow$ CLIQUEN_VEKTOR(T)
  return $n - m$
end

function CLIQUEN_VEKTOR(Co-Baum T)
  $w \leftarrow$ Wurzel von T
  if w ist ein Blatt **then**
    return $(1, 0, 0, 0)$
  else
    $T_l \leftarrow$ linker Teilbaum von w
    $(m_l, s_l, i_l, o_l) \leftarrow$ CLIQUEN_VEKTOR(T_l)
    $T_r \leftarrow$ rechter Teilbaum von w
    $(m_r, s_r, i_r, o_r) \leftarrow$ CLIQUEN_VEKTOR(T_r)
    if w ist mit $\cup$ markiert **then**
      return $(\max\{m_l, m_r\}, \max\{m_l, m_r\}, \max\{i_l, i_r\}, \max\{o_l, o_r\})$
    else if w ist mit $\gg$ markiert **then**
      return $(\max\{m_l, m_r\}, \max\{s_l, s_r, i_l, o_r\}, \max\{m_r, i_l\}, \max\{m_l, o_r\})$
    else $\triangleright$ w ist mit $\times$ markiert
      return $(\max\{m_l + s_r, m_r + s_l, i_l + i_r, o_l + o_r\}, s_l + s_r, i_l + i_r, o_l + o_r)$
    end
  end
end

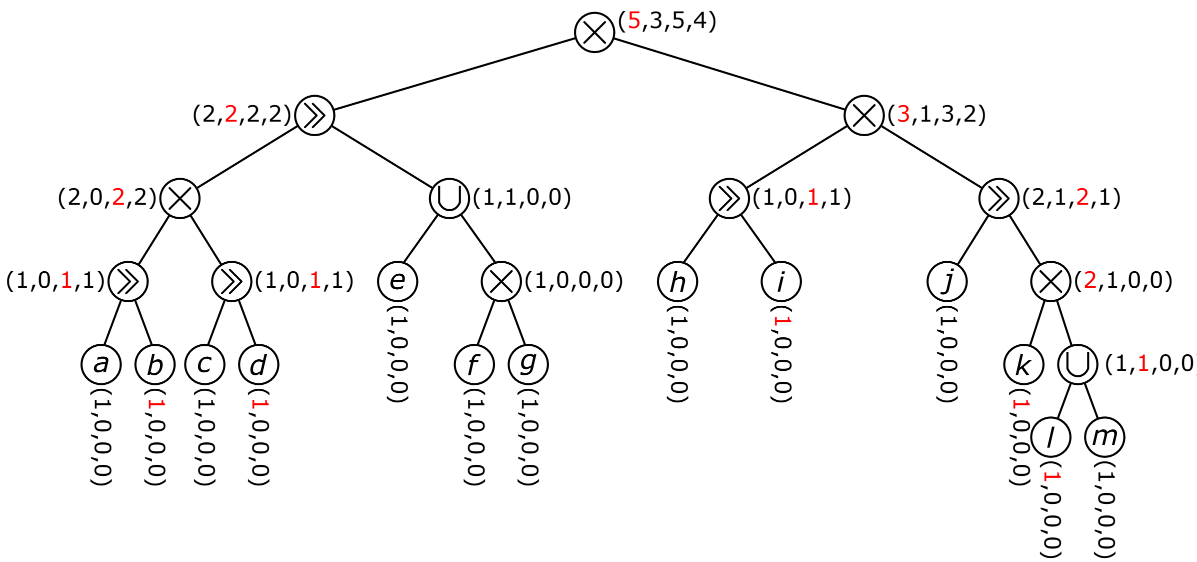


Abbildung 36: Ein Co-Baum eines gerichteten Co-Graphen mit den von Algorithmus 5 berechneten Werten. Eine größte Clique im Komplement des strong resolving Graphen hat die Größe 5. Eine Möglichkeit für eine solche Clique und ihre Berechnung ist in rot dargestellt. An der Wurzel werden in diesem Fall die zwei in-out-Knoten b, d des linken Teilbaums mit der Clique $\{i, k, l\}$ aus dem rechten Teilbaum kombiniert. Alle übrigen Knoten bilden eine stark trennende Menge der Größe 13.

8 Fazit und Ausblick

In dieser Arbeit wurde die Metrische Dimension und einige ihrer Varianten für spezielle Graphklassen untersucht. Nach einer Einleitung und einigen Grundlagen in Kapitel 1 und Kapitel 2, wurden in Kapitel 3 die verbreitetsten Varianten der Metrischen Dimension vorgestellt. Unter anderem wurde die k-Metrische Dimension aufgeführt, für die in Kapitel 4 die NP-Vollständigkeit gezeigt wurde. Der Beweis basiert auf einer Reduktion von 3DkM, einer Verallgemeinerung des bekannten Problems 3-Dimensionales Matching. Für 3DkM wurde zuvor ebenfalls die NP-Vollständigkeit gezeigt.

In Kapitel 5 wurde die ebenfalls zu Beginn vorgestellte gerichtete Metrische Dimension untersucht. Wie in 3.6.1 diskutiert, gibt es mehrere Möglichkeiten die Metrische Dimension auf gerichtete Graphen zu übertragen, sodass in Kapitel 5.1 zunächst die NP-Vollständigkeit der hier verwendeten Version für DAGs gezeigt wurde. Der Beweis basiert auf einer Reduktion von Hitting Set. Anschließend wurden stark zusammenhängende Co-Graphen untersucht. Für diese Graphen wurde ein Algorithmus vorgestellt, der die gerichtete Metrische Dimension in linearer Zeit berechnet. Dieser Algorithmus durchläuft den Co-Baum von den Blättern bis zur Wurzel und wählt, zusätzlich zu der Vereinigung der Teillösungen des linken und rechten Teilbaums, an jedem inneren Knoten maximal einen weiteren Ankerknoten.

Kapitel 6 beschäftigt sich mit der Starken Metrischen Dimension. Zu Beginn des Kapitels wurden ungerichtete Co-Graphen betrachtet, für die ebenfalls ein Linearzeitalgorithmus angegeben wurde. Dieser basiert darauf, dass die strong resolving Graphen von Co-Graphen auch Co-Graphen sind, für die ein minimales Vertex Cover in linearer Zeit berechnet werden kann. Darüber hinaus muss der strong resolving Graph für Co-Graphen nicht explizit bestimmt werden, da eine maximale Clique im ursprünglichen Graphen ausreichend ist, um die Starke Metrische Dimension zu berechnen. Anschließend wurden in Kapitel 6.2 Graphen behandelt, die durch die Verknüpfung mehrerer Graphen entstehen. Für diese Graphen wurde der strong resolving Graph bestimmt sowie dessen Vertex Cover. Jeder Graph kann als Verknüpfung seiner zweifachen Zusammenhangskomponente gesehen werden. Folglich wurde gezeigt, dass die Starke Metrische Dimension eines Graphen in polynomieller Zeit berechnet werden kann, wenn die Starke Metrische Dimension seiner zweifachen Zusammenhangskomponenten effizient bestimmt werden kann. Ein Algorithmus der die Starke Metrische Dimension basierend auf einer Zerlegung in die zweifachen Zusammenhangskomponenten bestimmt wurde ebenfalls beschrieben. Als Beispiele für diesen Algorithmus wurden Graphen betrachtet, deren zweifache Zusammenhangskomponenten Gitter, Kreise oder Co-Graphen sind.

In Kapitel 7 wurde die gerichtete Starke Metrische Dimension untersucht. Hierfür wurden erneut gerichtete Co-Graphen betrachtet und auch für diese Variante konnte ein Linearzeitalgorithmus angegeben werden. Dieser basiert, wie der zur Starken Metrischen Dimension, auf der Berechnung einer maximalen Clique, wodurch der strong resolving Graph nicht bestimmt werden muss. Auch dieser Algorithmus durchläuft den Co-Baum von den Blättern bis zur Wurzel und vereint Teillösungen der beiden Teilbäume.

Die Metrische Dimension bietet zahlreiche Möglichkeiten für weitere Arbeiten zu diesem Thema. Allen voran ist die Frage nach der Komplexität der Metrischen Dimension für Graphen mit einer Baumweite > 1 und < 24 offen. In Kapitel 2 wurden einige Ergebnisse zu speziellen Graphklassen vorgestellt, jedoch wurden nicht alle Varianten der Metrischen Dimension für alle diese Graphklassen untersucht. Dies gilt auch für die gerichtete Metrische Dimension, deren Problematiken in Kapitel 3.6.1 angesprochen wurden. Insbesondere durch die Vielzahl an Varianten und durch die Möglichkeit des Kombinierens der Varianten bleiben viele Graphklassen zu untersuchen. Vor allem sind jedoch die Starke und die gerichtete Starke Metrische Dimension von Interesse, da diese eine eindeutige Bestimmung der Kanten eines Graphen ermöglichen und damit beispielsweise optimales Routen in Sensornetzwerken erlauben. Speziell die gerichtete Starke Metrische Dimension wurde selten untersucht, sodass wenig über ihre Komplexität bekannt ist. Mögliche Ansätze zur Untersuchung dieser beiden Varianten bieten die Verknüpfung von Graphen oder die Berechnung einer maximalen Clique im Komplement des strong resolving Graphen, wie sie in dieser Arbeit verwendet wurden. Außerdem wurden in Kapitel 6.3 die strong resolving Graphen distanzerhaltender Graphen bestimmt, was die Grundlage für einen Linearzeitalgorithmus für die Starke Metrische Dimension dieser Graphklasse bilden könnte. In diesem Zuge könnte auch die gerichtete Starke Metrische Dimension von gerichteten Twinset-Graphen untersucht werden.

Literatur

[1] Marcel Abas and Tomás Vetrík. Metric dimension of cayley digraphs of split metacyclic groups. *Theor. Comput. Sci.*, 809:61–72, 2020.

[2] Ron Adar and Leah Epstein. An algorithm for the weighted metric dimension of two-dimensional grids. *Computing Research Repository (CoRR)*, abs/1602.05899, 2016.

[3] Ron Adar and Leah Epstein. The k-metric dimension. *Journal of Combinatorial Optimization*, 34(1):1–30, 2017.

[4] Júlio Araújo, Julien Bensmail, Victor A. Campos, Frédéric Havet, Ana Karolinna Maia, Nicolas Nisse, and Ana Silva. On finding the best and worst orientations for the metric dimension. *Algorithmica*, 85(10):2962–3002, 2023.

[5] S. Arumugam and Varughese Mathew. The fractional metric dimension of graphs. *Discrete Mathematics*, 312(9):1584–1590, 2012.

[6] S. Arumugam, Varughese Mathew, and Jian Shen. On fractional metric dimension of graphs. *Discrete Mathematics, Algorithms and Applications*, 5(4), 2013.

[7] Hans-Jürgen Bandelt and Henry Martyn Mulder. Distance-hereditary graphs. *J. Comb. Theory, Ser. B*, 41(2):182–208, 1986.

[8] Gabriel A. Barragán-Ramírez, Alejandro Estrada-Moreno, Yunior Ramírez-Cruz, and Juan A. Rodríguez-Velázquez. The simultaneous local metric dimension of graph families. *Symmetry*, 9(8):132, 2017.

[9] Gabriel A. Barragán-Ramírez, Carlos García Gómez, and Juan Alberto Rodríguez-Velázquez. Closed formulae for the local metric dimension of corona product graphs. *Electronic Notes in Discrete Mathematics*, 46:27–34, 2014.

[10] Gabriel A. Barragán-Ramírez and Juan A. Rodríguez-Velázquez. The local metric dimension of strong product graphs. *Graphs and Combinatorics*, 32(4):1263–1278, 2016.

[11] Zsolt Bartha, Júlia Komjáthy, and Järvi Raes. Sharp bound on the truncated metric dimension of trees. *Discret. Math.*, 346(8):113410, 2023.

[12] Denis Bechet, Philippe De Groote, and Christian Retoré. A complete axiomatisation for the inclusion of series-parallel partial orders. In *International Conference on Rewriting Techniques and Applications*, pages 230–240. Springer, 1997.

[13] Zuzana Beerliova, Felix Eberhard, Thomas Erlebach, Alexander Hall, Michael Hoffmann, Matúš Mihaľák, and L. Shankar Ram. Network discovery and verification. In Dieter Kratsch, editor, *Graph-Theoretic Concepts in Computer Science*, pages 127–138. Springer Berlin Heidelberg, 2005.

[14] Rémy Belmonte, Fedor V. Fomin, Petr A. Golovach, and M. S. Ramanujan. Metric dimension of bounded tree-length graphs. *SIAM Journal on Discrete Mathematics*, 31(2):1217–1243, 2017.

[15] Nadia Benakli, Novi H. Bong, Shonda Dueck, Linda Eroh, Beth Novick, and Ortrud R. Oellermann. The threshold strong dimension of a graph. *Discret. Math.*, 344(7):112402, 2021.

[16] Julien Bensmail, Fionn Mc Inerney, and Nicolas Nisse. Metric dimension: from graphs to oriented graphs. *Electronic Notes in Theoretical Computer Science*, 346:111–123, 2019.

[17] Édouard Bonnet and Nidhi Purohit. Metric dimension parameterized by treewidth. *Computing Research Repository (CoRR)*, abs/1907.08093, 2019.

[18] Robert C. Brigham, Gary Chartrand, Ronald D. Dutton, and Ping Zhang. Resolving domination in graphs. *Mathematica Bohemica*, 128(1):25–36, 2003.

[19] José Cáceres, M. Carmen Hernando, Mercè Mora, Ignacio M. Pelayo, María Luz Puertas, Carlos Seara, and David R. Wood. On the metric dimension of cartesian products of graphs. *SIAM Journal on Discrete Mathematics*, 21(2):423–441, 2007.

[20] Antonio Caruso, Stefano Chessa, Swades De, and A. Urpi. GPS free coordinate assignment and routing in wireless sensor networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, 13-17 March 2005, Miami, FL, USA*, pages 150–160, 2005.

[21] Maw-Shang Chang, Sun-Yuan Hsieh, and Gen-Huey Chen. Dynamic programming on distance-hereditary graphs. In Hon Wai Leong, Hiroshi Imai, and Sanjay Jain, editors, *Algorithms and Computation, 8th International Symposium, ISAAC '97, Singapore, December 17-19, 1997, Proceedings*, volume 1350 of *Lecture Notes in Computer Science*, pages 344–353. Springer, 1997.

[22] Gary Chartrand, Linda Eroh, Mark A. Johnson, and Ortrud Oellermann. Resolvability in graphs and the metric dimension of a graph. *Discrete Applied Mathematics*, 105(1-3):99–113, 2000.

[23] Gary Chartrand, Michael Raines, Ping Zhang, and Kalamazoo. The directed distance dimension of oriented graphs. *Mathematica Bohemica*, 125(02):155–168, 2000.

[24] Gary Chartrand, Varaporn Saenpholphat, and Ping Zhang. The independent resolving number of a graph. *Mathematica Bohemica*, 128(4):379–393, 2003.

[25] Derek G Corneil, Helmut Lerchs, and L Stewart Burlingham. Complement reducible graphs. *Discrete Applied Mathematics*, 3(3):163–174, 1981.

[26] Christophe Crespelle and Christophe Paul. Fully dynamic recognition algorithm and certificate for directed cographs. *Discrete Applied Mathematics*, 154(12):1722–1741, 2006.

[27] Antoine Dailly, Florent Foucaud, and Anni Hakanen. Algorithms and hardness for metric dimension on digraphs. In Daniël Paulusma and Bernard Ries, editors, *Graph-Theoretic Concepts in Computer Science - 49th International Workshop, WG 2023, Fribourg, Switzerland, June 28-30, 2023, Revised Selected Papers*, volume 14093 of *Lecture Notes in Computer Science*, pages 232–245. Springer, 2023.

[28] Milica Milivojevic Danas. The difference between several metric dimension graph invariants. *Discret. Appl. Math.*, 332:1–6, 2023.

[29] Bhaskar DasGupta and Nasim Mobasheri. On optimal approximability results for computing the strong metric dimension. *Computing Research Repository (CoRR)*, abs/1408.1390, 2014.

[30] Josep Díaz, Olli Pottonen, Maria J. Serna, and Erik Jan van Leeuwen. Complexity of metric dimension on planar graphs. *Journal of Computer and System Sciences*, 83(1):132–158, 2017.

[31] Leah Epstein, Asaf Levin, and Gerhard J. Woeginger. The (weighted) metric dimension of graphs: Hard and easy cases. In *Graph-Theoretic Concepts in Computer Science - 38th International Workshop, WG 2012, Jerusalem, Israel, June 26-28, 2012, Revised Selcted Papers*, pages 114–125, 2012.

[32] Leah Epstein, Asaf Levin, and Gerhard J. Woeginger. The (weighted) metric dimension of graphs: Hard and easy cases. *Algorithmica*, 72(4):1130–1171, 2015.

[33] Linda Eroh, Cong X. Kang, and Eunjeong Yi. The connected metric dimension at a vertex of a graph. *Theor. Comput. Sci.*, 806:53–69, 2020.

[34] Alejandro Estrada-Moreno, Juan Alberto Rodríguez-Velázquez, and Ismael González Yero. The k-metric dimension of a graph. *arXiv preprint arXiv:1312.6840*, 2013.

[35] Alejandro Estrada-Moreno, IG Yero, and JA Rodríguez-Velázquez. On the (k, t)-metric dimension of graphs. *The Computer Journal*, 2016.

[36] Alejandro Estrada-Moreno, Ismael González Yero, and Juan A. Rodríguez-Velázquez. The k-metric dimension of the lexicographic product of graphs. *Discrete Mathematics*, 339(7):1924–1934, 2016.

[37] Alejandro Estrada-Moreno, Ismael González Yero, and Juan A Rodríguez-Velázquez. Relationships between the 2-metric dimension and the 2-adjacency dimension in the lexicographic product of graphs. *Graphs and Combinatorics*, 32:2367–2392, 2016.

[38] Melodie Fehr, Shonda Gosselin, and Ortrud R. Oellermann. The metric dimension of cayley digraphs. *Discrete Mathematics*, 306(1):31–41, 2006.

[39] Min Feng, Benjian Lv, and Kaishun Wang. On the fractional metric dimension of graphs. *Discrete Applied Mathematics*, 170:55–63, 2014.

[40] Min Feng, Kaishun Wang, and Yuefeng Yang. On weak metric dimension of digraphs. *Discret. Math. Algorithms Appl.*, 15(3):2250088:1–2250088:17, 2023.

[41] Min Feng, Min Xu, and Kaishun Wang. On the metric dimension of line graphs. *Discrete Applied Mathematics*, 161(6):802–805, 2013.

[42] Henning Fernau and Juan A. Rodríguez-Velázquez. On the (adjacency) metric dimension of corona and strong product graphs and their local variants: Combinatorial and computational results. *Discrete Applied Mathematics*, 236:183–202, 2018.

[43] Rodrigo Fonseca, Sylvia Ratnasamy, Jerry Zhao, Cheng Tien Ee, David E. Culler, Scott Shenker, and Ion Stoica. Beacon vector routing: Scalable point-to-point routing in wireless sensornets. In *2nd Symposium on Networked Systems Design and Implementation (NSDI 2005), May 2-4, 2005, Boston, Massachusetts, USA, Proceedings*, pages 329–342, 2005.

[44] Florent Foucaud, George B. Mertzios, Reza Naserasr, Aline Parreau, and Petru Valicov. Identification, location-domination and metric dimension on interval and permutation graphs. II. algorithms and complexity. *Algorithmica*, 78(3):914–944, 2017.

[45] Rafael M. Frongillo, Jesse Geneson, Manuel E. Lladser, Richard C. Tillquist, and Eunjeong Yi. Truncated metric dimension for finite graphs. *Discret. Appl. Math.*, 320:150–169, 2022.

[46] Michael Randolph Garey and David Stifler Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[47] Jesse Geneson. Metric dimension and pattern avoidance in graphs. *Discret. Appl. Math.*, 284:1–7, 2020.

[48] Jesse Geneson, Suchir Kaustav, and Antoine Labelle. Extremal results for graphs of bounded metric dimension. *Discret. Appl. Math.*, 309:123–129, 2022.

[49] Paul Gutkovich and Zi Song Yeoh. Computing truncated metric dimension of trees. *CoRR*, abs/2302.05960, 2023.

[50] Michel Habib and Christophe Paul. A simple linear time algorithm for cograph recognition. *Discrete Applied Mathematics*, 145(2):183–197, 2005.

[51] Anni Hakanen, Ville Junnila, and Tero Laihonen. The solid-metric dimension. *Theor. Comput. Sci.*, 806:156–170, 2020.

[52] Anni Hakanen and Tero Laihonen. On {l}-metric dimensions in graphs. *Fundamenta Informaticae*, 162(2-3):143–160, 2018.

[53] Peter L. Hammer and Frédéric Maffray. Completely separable graphs. *Discret. Appl. Math.*, 27(1-2):85–99, 1990.

[54] Frank Harary and Robert A. Melter. On the metric dimension of a graph. *Ars Combinatoria*, 2:191–195, 1976.

[55] Sepp Hartung and André Nichterlein. On the parameterized and approximation hardness of metric dimension. In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, K.lo Alto, California, USA, 5-7 June, 2013*, pages 266–276, 2013.

[56] Mathias Hauptmann, Richard Schmied, and Claus Viehmann. Approximation complexity of metric dimension problem. *Journal of Discrete Algorithms*, 14:214–222, 2012.

[57] Sakander Hayat. Computing distance-based topological descriptors of complex chemical networks: New theoretical techniques. *Chemical Physics Letters*, 688(1):51–58, 1977.

[58] M. Carmen Hernando, Mercè Mora, Ignacio M. Pelayo, Carlos Seara, José Cáceres, and María Luz Puertas. On the metric dimension of some families of graphs. *Electronic Notes in Discrete Mathematics*, 22:129–133, 2005.

[59] M. Carmen Hernando, Mercè Mora, Peter J. Slater, and David R. Wood. Fault-tolerant metric dimension of graphs. *Convexity in Discrete Structures*, 5:81–85, 2008.

[60] Stefan Hoffmann, Alina Elterman, and Egon Wanke. A linear time algorithm for metric dimension of cactus block graphs. *Theoretical Computer Science*, 630:43–62, 2016.

[61] Stefan Hoffmann and Egon Wanke. Metric dimension for gabriel unit disk graphs is np-complete. In *Algorithms for Sensor Systems, 8th International Symposium on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile Entities, ALGOSENSORS 2012, Ljubljana, Slovenia, September 13-14, 2012. Revised Selected Papers*, pages 90–92, 2012.

[62] Edward Howorka. A characterization of distance-hereditary graphs. *The quarterly journal of mathematics*, 28(4):417–420, 1977.

[63] Yufei Huang, Bo Hou, Wen Liu, Lidong Wu, Stephen Rainwater, and Suogang Gao. On approximation algorithm for the edge metric dimension problem. *Theor. Comput. Sci.*, 853:2–6, 2021.

[64] H. Iswadi, Edy Tri Baskoro, A.N.M. Salman, and Rinovia Simanjuntak. The metric dimension of amalgamation of cycles. *Far East Journal of Mathematical Sciences (FJMS)*, 41(1):19–31, 2010.

[65] Beverly Jamison and Stephan Olariu. A linear-time recognition algorithm for p4-reducible graphs. *Theor. Comput. Sci.*, 145(1&2):329–344, 1995.

[66] Mohsen Jannesari and Behnaz Omoomi. The metric dimension of the lexicographic product of graphs. *Discrete Mathematics*, 312(22):3349–3356, 2012.

[67] Zilin Jiang and Nikita Polyanskii. On the metric dimension of cartesian powers of a graph. *Computing Research Repository (CoRR)*, abs/1712.02723, 2017.

[68] Cong X. Kang. On the fractional strong metric dimension of graphs. *Discrete Applied Mathematics*, 213:153–161, 2016.

[69] Cong X. Kang, Ismael González Yero, and Eunjeong Yi. The fractional strong metric dimension in three graph products. *Discrete Applied Mathematics*, 251:190–203, 2018.

[70] Na Kang, Zhiquan Li, Lihang Hou, and Jing Qu. Mixed metric dimension of some plane graphs. In Qiufen Ni and Weili Wu, editors, *Algorithmic Aspects in Information and Management - 16th International Conference, AAIM 2022, Guangzhou, China, August 13-14, 2022, Proceedings*, volume 13513 of *Lecture Notes in Computer Science*, pages 363–375. Springer, 2022.

[71] Richard M. Karp. *Reducibility among Combinatorial Problems.* Springer US, 1972.

[72] Aleksander Kelenc, Dorota Kuziak, Andrej Taranenko, and Ismael González Yero. Mixed metric dimension of graphs. *Applied Mathematics and Computation*, 314:429–438, 2017.

[73] Aleksander Kelenc, Niko Tratnik, and Ismael González Yero. Uniquely identifying the edges of a graph: The edge metric dimension. *Discrete Applied Mathematics*, 251:204–220, 2018.

[74] Samir Khuller, Balaji Raghavachari, and Azriel Rosenfeld. Localization in graphs. 1994.

[75] Samir Khuller, Balaji Raghavachari, and Azriel Rosenfeld. Landmarks in graphs. *Discrete Applied Mathematics*, 70(3):217–229, 1996.

[76] Sandi Klavžar, Freydoon Rahbarnia, and Mostafa Tavakoli. Some binary products and integer linear programming for k-metric dimension of graphs. *Applied Mathematics and Computation*, 409:126420, 2021.

[77] Sandi Klavzar and Mostafa Tavakoli. Local metric dimension of graphs: Generalized hierarchical products and some applications. *Applied Mathematics and Computation*, 364, 2020.

[78] Sandi Klavzar and Mostafa Tavakoli. Edge metric dimensions via hierarchical product and integer linear programming. *Optim. Lett.*, 15(6):1993–2003, 2021.

[79] Martin Knor, Snjezana Majstorovic, Aoden Teo Masa Toshi, Riste Skrekovski, and Ismael G. Yero. Graphs with the edge metric dimension smaller than the metric dimension. *Appl. Math. Comput.*, 401:126076, 2021.

[80] Martin Knor, Riste Skrekovski, and Ismael G. Yero. A note on the metric and edge metric dimensions of 2-connected graphs. *Discret. Appl. Math.*, 319:454–460, 2022.

[81] Jozef Kratica, Vera Kovacevic-Vujcic, Mirjana Cangalovic, and Milica Stojanovic. Minimal doubly resolving sets and the strong metric dimension of some convex polytopes. *Applied Mathematics and Computation*, 218(19):9790–9801, 2012.

[82] Jozef Kratica, Vera Kovačević-Vujčić, Mirjana Cangalovic, and Nenad Mladenovic. Strong metric dimension: A survey. *Yugoslav Journal of Operations Research*, 24(2):187–198, 2014.

[83] Dorota Kuziak, Juan Alberto Rodríguez-Velázquez, and Ismael González Yero. On the strong metric dimension of product graphs. *Electronic Notes in Discrete Mathematics*, 46:169–176, 2014.

[84] Dorota Kuziak, Juan Alberto Rodríguez-Velázquez, and Ismael González Yero. Closed formulae for the strong metric dimension of lexicographic product graphs. *Discussiones Mathematicae Graph Theory*, 36(4):1051–1064, 2016.

[85] Dorota Kuziak, Juan Alberto Rodríguez-Velázquez, and Ismael González Yero. Computing the metric dimension of a graph from primary subgraphs. *Discussiones Mathematicae Graph Theory*, 37(1):273–293, 2017.

[86] Dorota Kuziak and Ismael G Yero. Metric dimension related parameters in graphs: A survey on combinatorial, computational and applied results. *arXiv preprint arXiv:2107.04877*, 2021.

[87] Dorota Kuziak, Ismael González Yero, and Juan A. Rodríguez-Velázquez. On the strong metric dimension of cartesian sum graphs. *Fundamenta Informaticae*, 141(1):57–69, 2015.

[88] Dorota Kuziak, Ismael González Yero, and Juan A. Rodríguez-Velázquez. Strong metric dimension of rooted product graphs. *International Journal of Computer Mathematics*, 93(8):1265–1280, 2016.

[89] Dorota Kuziak, Ismael González Yero, and Juan Alberto Rodríguez-Velázquez. On the strong metric dimension of corona product graphs and join graphs. *Discrete Applied Mathematics*, 161(7-8):1022–1027, 2013.

[90] Sohan Lal and Vijay Kumar Bhat. On the dominant local metric dimension of some planar graphs. *Discret. Math. Algorithms Appl.*, 15(7):2250152:1–2250152:20, 2023.

[91] Shaohua Li and Marcin Pilipczuk. Hardness of metric dimension in graphs of constant treewidth. *Algorithmica*, 84(11):3110–3155, 2022.

[92] Ke Liu and Nael B. Abu-Ghazaleh. Virtual coordinates with backtracking for void traversal in geographic routing. In *Ad-Hoc, Mobile, and Wireless Networks, 5th International Conference, ADHOC-NOW 2006, Ottawa, Canada, August 17-19, 2006, Proceedings*, pages 46–59, 2006.

[93] Teresa R. May and Ortrud R. Oellermann. The strong dimension of distance-hereditary graphs. *JCMCC-Journal of Combinatorial Mathematics and Combinatorial Computing*, 76:59, 2011.

[94] Nenad Mladenovic, Jozef Kratica, Vera Kovacevic-Vujcic, and Mirjana Cangalovic. Variable neighborhood search for metric dimension and minimal doubly resolving set problems. *European Journal of Operational Research*, 220(2):328–337, 2012.

[95] Lucas Mol, Matthew J. H. Murphy, and Ortrud R. Oellermann. The threshold dimension of a graph. *Discret. Appl. Math.*, 287:118–133, 2020.

[96] Gaia Moravcik, Ortrud R. Oellermann, and Samuel Yusim. Comparing the metric and strong dimensions of graphs. *Discrete Applied Mathematics*, 220:68–79, 2017.

[97] Kairui Nie and Kexiang Xu. Mixed metric dimension of some graphs. *Appl. Math. Comput.*, 442:127737, 2023.

[98] Ortrud R. Oellermann, Charlene D. Pawluck, and Anna Stokke. The metric dimension of cayley digraphs of abelian groups. *Ars Combinatoria*, 81, 2006.

[99] Ortrud R. Oellermann and Joel Peters-Fransen. The strong metric dimension of graphs and digraphs. *Discrete Applied Mathematics*, 155(3):356–364, 2007.

[100] Futaba Okamoto, Bryan Phinezy, and Ping Zhang. The local metric dimension of a graph. *Mathematica Bohemica*, 135:239–255, 01 2010.

[101] Iztok Peterin and Ismael G Yero. Edge metric dimension of some graph operations. *Bulletin of the Malaysian Mathematical Sciences Society*, 43(3):2465–2477, 2020.

[102] Jing Qu and Nanbin Cao. Edge metric dimension and mixed metric dimension of planar graph qn. *Discret. Appl. Math.*, 320:462–475, 2022.

[103] Bharati Rajan, Indra Rajasingh, Jude Annie Cynthia, and Paul D. Manuel. Metric dimension of directed graphs. *International Journal of Computer Mathematics*, 91(7):1397–1406, 2014.

[104] Yunior Ramírez-Cruz, Alejandro Estrada-Moreno, and Juan Alberto Rodríguez-Velázquez. The simultaneous metric dimension of families composed by lexicographic product graphs. *Graphs and Combinatorics*, 32(5):2093–2120, 2016.

[105] Yunior Ramírez-Cruz, Ortrud R. Oellermann, and Juan A. Rodríguez-Velázquez. The simultaneous metric dimension of graph families. *Discrete Applied Mathematics*, 198:241–250, 2016.

[106] Juan Alberto Rodríguez-Velázquez, Ismael González Yero, Dorota Kuziak, and Ortrud R. Oellermann. On the strong metric dimension of cartesian and direct products of graphs. *Discrete Mathematics*, 335:8–19, 2014.

[107] Varaporn Saenpholphat and Ping Zhang. Conditional resolvability in graphs: a survey. *International Journal of Mathematics and Mathematical Sciences*, 2004(38):1997–2017, 2004.

[108] Muhammad Salman, Imran Javaid, and Muhammad Anwar Chaudhry. Minimum fault-tolerant, local and strong metric dimension of graphs. *Ars Combinatoria*, 138:333–353, 2018.

[109] Suhadi Wido Saputro, Rinovia Simanjuntak, Saladin Uttunggadewa, Hilda Assiyatun, Edy Tri Baskoro, A. N. M. Salman, and Martin Baca. The metric dimension of the lexicographic product of graphs. *Discrete Mathematics*, 313(9):1045–1051, 2013.

[110] Yannick Schmitz, Duygu Vietz, and Egon Wanke. A note on the complexity of k-metric dimension. *Applied Mathematics and Computation*, 457:128204, 2023.

[111] Yannick Schmitz and Egon Wanke. On the strong metric dimension of directed co-graphs. *arXiv preprint arXiv:2111.13054*, 2021.

[112] Yannick Schmitz and Egon Wanke. The directed metric dimension of directed co-graphs. *arXiv preprint arXiv:2306.08594*, 2023.

[113] András Sebö and Eric Tannier. On metric generators of graphs. *Mathematics of Operations Research*, 29(2):383–393, 2004.

[114] Jelena Sedlar and Riste Skrekovski. Mixed metric dimension of graphs with edge disjoint cycles. *Discret. Appl. Math.*, 300:1–8, 2021.

[115] Jelena Sedlar and Riste Skrekovski. Vertex and edge metric dimensions of cacti. *Discret. Appl. Math.*, 320:126–139, 2022.

[116] Jelena Sedlar and Riste Skrekovski. Vertex and edge metric dimensions of unicyclic graphs. *Discret. Appl. Math.*, 314:81–92, 2022.

[117] Rinovia Simanjuntak, Saladin Uttunggadewa, and Suhadi Wido Saputro. Metric dimension for amalgamations of graphs. In *Combinatorial Algorithms - 25th International Workshop, IWOCA 2014, Duluth, MN, USA, October 15-17, 2014, Revised Selected Papers*, pages 330–337, 2014.

[118] Peter J. Slater. Leaves of trees. *Congressum Numerantium*, 14:549–559, 1975.

[119] B. Suganya and S. Arumugam. Independent resolving number of convex polytopes. In *Theoretical Computer Science and Discrete Mathematics - First International Conference, ICTCSDM 2016, Krishnankoil, India, December 19-21, 2016, Revised Selected Papers*, pages 401–408, 2016.

[120] Liliek Susilowati, Siti Istikhomah, Mohammad Imam Utoyo, and S. Slamin. The local complement metric dimension of graphs. *Discret. Math. Algorithms Appl.*, 15(2):2250073:1–2250073:19, 2023.

[121] Liliek Susilowati, Imroatus Sa'adah, and Utami Dyah Purwati. On the joint product graphs with respect to dominant metric dimension. *Discret. Math. Algorithms Appl.*, 13(2):2150010:1–2150010:11, 2021.

[122] Mostafa Tavakoli, Meysam Korivand, Ahmad Erfanian, Gholamreza Abrishami, and Edy Tri Baskoro. The dominant edge metric dimension of graphs. *Electron. J. Graph Theory Appl.*, 11(1):197–208, 2023.

[123] Richard C. Tillquist, Rafael M. Frongillo, and Manuel E. Lladser. Getting the lay of the land in discrete space: A survey of metric dimension and its applications. *SIAM Rev.*, 65(4):919–962, 2023.

[124] Duygu Vietz. *Die Metrische Dimension spezieller Graphklassen*. PhD thesis, 2020.

[125] Duygu Vietz, Stefan Hoffmann, and Egon Wanke. Computing the metric dimension by decomposing graphs into extended biconnected components - (extended abstract). In *WALCOM: Algorithms and Computation - 13th International Conference, WALCOM 2019, Guwahati, India, February 27 - March 2, 2019, Proceedings*, pages 175–187, 2019.

[126] Duygu Vietz and Egon Wanke. The fault-tolerant metric dimension of cographs. In *Fundamentals of Computation Theory - 22nd International Symposium, FCT 2019, Copenhagen, Denmark, August 12-14, 2019, Proceedings*, pages 350–364, 2019.

[127] Marcel Wagner, Yannick Schmitz, and Egon Wanke. On the strong metric dimension of composed graphs. *arXiv preprint arXiv:2212.04166*, 2022.

[128] Changcheng Wei, Muhammad Salman, Syed Shahzaib, Masood Ur Rehman, and Juanyan Fang. Classes of planar graphs with constant edge metric dimension. *Complex.*, 2021:5599274:1–5599274:10, 2021.

[129] Meiqin Wei, Jun Yue, and Lily Chen. The effect of vertex and edge deletion on the edge metric dimension of graphs. *J. Comb. Optim.*, 44(1):331–342, 2022.

[130] Ismael González Yero, Alejandro Estrada-Moreno, and Juan A. Rodríguez-Velázquez. Computing the k-metric dimension of graphs. *Applied Mathematics and Computation*, 300:60–69, 2017.

[131] Ismael González Yero, Dorota Kuziak, and Juan Alberto Rodríguez-Velázquez. On the metric dimension of corona product graphs. *Computers & Mathematics with Applications*, 61(9):2793–2798, 2011.

[132] Eunjeong Yi. The fractional k-truncated metric dimension of graphs. In Ding-Zhu Du, Donglei Du, Chenchen Wu, and Dachuan Xu, editors, *Combinatorial Optimization and Applications - 15th International Conference, COCOA 2021, Tianjin, China, December 17-19, 2021, Proceedings*, volume 13135 of *Lecture Notes in Computer Science*, pages 568–578. Springer, 2021.

[133] Eunjeong Yi. On the connected metric dimension of graphs and their complements. *Discret. Math. Algorithms Appl.*, 13(5):2150059:1–2150059:17, 2021.

[134] Enqiang Zhu, Andrej Taranenko, Zehui Shao, and Jin Xu. On graphs with the maximum edge metric dimension. *Discrete Applied Mathematics*, 257:317–324, 2019.

[135] Nina Zubrilina. On the edge dimension of a graph. *Discrete Mathematics*, 341(7):2083–2088, 2018.

9 Anhang

In diesem Kapitel werden die Veröffentlichungen und Manuskripte aufgeführt, auf denen diese Arbeit basiert. Die Ergebnisse wurden zu einer Gesamtarbeit zusammengefügt, durch neue Abbildungen verdeutlicht, Beweise wurden angepasst, Notationen vereinheitlicht und noch unveröffentlichte Ergebnisse hinzugefügt.

9.1 A note on the complexity of k-Metric Dimension

A note on the complexity of k-Metric Dimension
Yannick Schmitz, Duygu Vietz, Egon Wanke
Applied Mathematics and Computation (2023)

Diese Arbeit enthält einen Beweis für die NP-Vollständigkeit der k-Metrischen Dimension, der auch in Kapitel 4 zu finden ist. Alle Autoren haben gleichermaßen zur Erarbeitung und zur Formulierung der Ergebnisse beigetragen. Den Hauptteil meines Beitrags bildet der NP-Vollständigkeitsbeweis von 3DkM.

Applied Mathematics and Computation 457 (2023) 128204

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

A note on the complexity of k-METRIC DIMENSION

Yannick Schmitz*, Duygu Vietz, Egon Wanke

Heinrich-Heine-Universität Düsseldorf, Universitätsstraße 1, Düsseldorf, 40225, Germany

ARTICLE INFO

Article history:
Received 29 November 2022
Revised 14 June 2023
Accepted 21 June 2023

Keywords:
Metric dimension
K-Metric dimension
Resolving set
Metric generator

ABSTRACT

Two vertices $u, v \in V$ of an undirected connected graph $G = (V, E)$ are *resolved* by a vertex w if the distance between u and w and the distance between v and w are different. A set $R \subseteq V$ of vertices is a *k-resolving set* for G if for each pair of vertices $u, v \in V$ there are at least k distinct vertices $w_1, \ldots, w_k \in R$ such that each of them resolves u and v. The *k-Metric Dimension* of G is equal to the size of a smallest k-resolving set for G. The decision problem k-Metric Dimension is the question whether G has a k-resolving set of size at most r, for a given graph G and a given number r. In this paper, we proof the NP-completeness of k-Metric Dimension for bipartite graphs and each $k \geq 2$, which corrects the proof in [1].

1. Introduction

The metric dimension of graphs has been introduced in the 1970s independently by Slater [2] and by Harary and Melter [3]. We consider simple undirected and connected graphs $G = (V, E)$, where V is the set of vertices and $E \subseteq \{\{u, v\} \mid u, v \in V, u \neq v\}$ is the set of edges. Such a graph *has metric dimension* at most r if there is a vertex set $R \subseteq V$ such that $|R| \leq r$ and $\forall u, v \in V$, $u \neq v$, there is a vertex $w \in R$ such that $d(w, u) \neq d(w, v)$, where $d(u, v)$ is the distance (the length of a shortest path in an unweighted graph) between u and v. The *metric dimension* of G is the smallest integer r such that G has metric dimension at most r.

If $d(w, u) \neq d(w, v)$, for three vertices u, v, w, we say that u and v are *resolved* or *distinguished* by vertex w. If every pair of vertices is resolved by at least one vertex of vertex set R, then R is a *resolving set* or *metric generator* for G. In certain applications, the vertices of a resolving set are also called *landmark nodes* or *anchor nodes*. This is a common naming, particularly in the theory of sensor networks.

The metric dimension finds applications in various areas, including network discovery and verification [4], geographical routing protocols [5], combinatorial optimization [6], sensor networks [7], robot navigation [8] and chemistry [9,10].

There are several algorithms for computing a minimum resolving set in polynomial time for special classes of graphs, for example trees [8,9], wheels [11], grid graphs [12], k-regular bipartite graphs [13], amalgamation of cycles [14] and outerplanar graphs [15]. The approximability of the metric dimension has been studied for bounded degree, dense and general graphs in [16]. Upper and lower bounds on the metric dimension are considered in [17,18] for further classes of graphs.

In this paper, we consider the *k-Metric Dimension* for some positive integer k. A set $R \subseteq V$ of vertices is a *k-resolving set* for G if for each pair of vertices $u, v \in V$ there are at least k vertices $w_1, \ldots, w_k \in R$ such that each of them resolves u and v. The *k-Metric Dimension* of G is equal to the size of a smallest k-resolving set for G. The k-Metric Dimension problem

* Corresponding author.
E-mail address: yannick.schmitz@hhu.de (Y. Schmitz).

https://doi.org/10.1016/j.amc.2023.128204
0096-3003/

was introduced by Estrada-Moreno et al. in [19]. The 1-metric dimension is simply called metric dimension. The 2-metric dimension is also called *fault-tolerant metric dimension* and was introduced in [20].

Estrada-Moreno et al. analysed the (k,t)-Metric Dimension [21]. The (k,t)-Metric Dimension is the k-Metric Dimension, with the addition, that the distance between two vertices u, v of G is defined as the minimum of $d(u,v)$ and t. Therefore, if t is greater than or equal to the diameter of G, then (k,t)-Metric Dimension is equivalent to k-Metric Dimension. Estrada-Moreno et al. showed the NP-completeness of (k,t)-Metric Dimension for odd values of k and $t \geq 2$.

The decision problem k-Metric Dimension is defined as follows.

k-Metric Dimension	
Instance:	An undirected connected graph $G = (V, E)$ and a number r.
Question:	Is there a k-resolving set $R \subseteq V$ for G of size at most r?

The complexity of k-Metric Dimension has only been investigated for very few graph classes, such as trees and other simple graph classes. For general graph classes, k-Metric Dimension is assumed to be NP-complete if k is given as part of the input. The decision problem 1-Metric Dimension is known to be NP-complete, see [22]. A proof can be found in [8]. In this paper, we show the NP-completeness of k-Metric Dimension for bipartite graphs and each $k \geq 2$ by an alternative approach to [1], whose proof unfortunately is incorrect and does not offer any simple correction options.

2. The NP-completeness of *k*-Metric dimension

In this section, k-Metric Dimension is shown to be NP-complete for bipartite graphs and each $k \geq 2$ by a reduction from 3-Dimensional k-Matching, which is defined as follows.

3-Dimensional k-Matching (3DkM)	
Instance:	A set $S \subseteq A \times B \times C$, where A, B and C are disjoint sets of the same size n.
Question:	Does S contain a k-matching, i.e. a subset M of size $k \cdot n$ such that each element of A, B and C is contained in exactly k triples of M?

For $k = 1$, the 3D1M problem is the well-known NP-complete 3-Dimensional Matching (3DM) problem, see [22]. The next theorem shows that 3DkM is also NP-complete for each $k \geq 2$.

Theorem 1. 3DkM is NP-complete for each $k \geq 2$.

Proof. The 3DkM problem is obviously in NP, because it can be checked in polynomial time whether a selection of triples from S is a k-matching.

The NP-hardness is shown by a reduction from 3DM. Let

$$\begin{array}{ll} A = \{a_1, \ldots, a_n\}, & B = \{b_1, \ldots, b_n\}, \\ C = \{c_1, \ldots, c_n\}, \text{ and } & S = \{s_1, \ldots, s_m\} \end{array}$$

be an instance for 3DM. Without loss of generality, n is assumed to be a multiple of $(k-1)$, that is $n = r(k-1)$ for a positive integer r. If this is not the case, then expand A, B and C by at most $k-2$ elements each and S by at most $k-2$ triples, which cover every additional element exactly once and none of the originally given elements.

Now consider the following instance for 3DkM defined by

$$\begin{array}{ll} A' = A \cup \{a_{n+1}, \ldots, a_{3n}\}, & B' = B \cup \{b_{n+1}, \ldots, b_{3n}\}, \\ C' = C \cup \{c_{n+1}, \ldots, c_{3n}\}, \text{ and } & S' = S \cup R \cup T \end{array}$$

where $R = \{(a_i, b_i, c_i) \mid n+1 \leq i \leq 3n\}$ and $T \subseteq A' \times B' \times C'$. Set T has exactly $3n(k-1)$ triples, which will be defined later.

The sets A', B' and C' contain the elements of A, B and C, respectively, and $2n$ additional elements. Set S' contains the elements of S, R and T

Let $U = A \cup B \cup C$ and $U' = A' \cup B' \cup C'$. The $2n$ triples of R cover each element of $U' \setminus U$ exactly once and no element of U. Set T is defined such that its $3n(k-1)$ triples cover each element of U' exactly $k-1$ times. Each triple of T covers exactly one element from U and two elements from $U' \setminus U$.

If M is a matching for U, then $M \cup R \cup T$ is obviously a k-matching for U' for any $k \geq 2$. Any k-matching M' for U' contains all triples from R and T, because otherwise it is not possible to cover the elements of $U' \setminus U$ at least k times. The triples of T cover the elements of U' exactly $k-1$ times. That is, if M' is a k-matching for U', then $M = M' \setminus (R \cup T)$ is a matching for U.

Set T can be easily defined with the help of

$$T_{p,q} \subseteq (A \times B) \cup (A \times C) \cup (B \times C)$$

defined by

$$T_{p,q} = \begin{array}{cl} & \{(a_i, b_j) \mid i \in \{p, \ldots, p+q-1\},\ j \in \{p+q, \ldots, p+2q-1\}\} \\ \cup & \{(b_i, c_j) \mid i \in \{p, \ldots, p+q-1\},\ j \in \{p+q, \ldots, p+2q-1\}\} \\ \cup & \{(c_i, a_j) \mid i \in \{p, \ldots, p+q-1\},\ j \in \{p+q, \ldots, p+2q-1\}\}. \end{array}$$

These $3q^2$ tuples cover each element of

$$\{a_p, \ldots, a_{p+2q-1},\ b_p, \ldots, b_{p+2q-1},\ c_p, \ldots, c_{p+2q-1}\}$$

exactly q times. There are

- q^2 tuples between the elements of $\{a_p, \ldots, a_{p+q-1}\}$ and $\{b_{p+q}, \ldots, b_{p+2q-1}\}$,
- q^2 tuples between the elements of $\{b_p, \ldots, b_{p+q-1}\}$ and $\{c_{p+q}, \ldots, c_{p+2q-1}\}$, and
- q^2 tuples between the elements of $\{c_p, \ldots, c_{p+q-1}\}$ and $\{a_{p+q}, \ldots, a_{p+2q-1}\}$.

Now let T' be defined by

$$T' = \bigcup_{i=0}^{r-1} T_{n+1+i2(k-1),\,k-1}, \quad \text{with } r = \frac{n}{k-1}.$$

T' has $r3(k-1)^2 = \frac{n}{k-1} \cdot 3(k-1)^2 = 3n(k-1)$ tuples. It is the union of $r = \frac{n}{k-1}$ sets $T_{p,q}$ where index p is running from $n+1$ to $3n+1-2(k-1)$ in steps of width $2(k-1)$ and $q = k-1$. These tuples of T' cover each element of $U' \setminus U$ exactly $(k-1)$ times.

In the last step, the $3n(k-1)$ tuples of T' are expanded to $3n(k-1)$ triples for T, by including each element from U to exactly $k-1$ tuples from T', such that each generated triple is from $A' \times B' \times C'$. Each tuple from T' is extended by exactly one element from U. The result is the set T of triples with the required properties. This transformation can obviously be done in polynomial time, see also Example 1. □

Example 1. Let $A = \{a_1, \ldots, a_4\}$, $B = \{b_1, \ldots, b_4\}$, $C = \{c_1, \ldots, c_4\}$ and

$$S = \{(a_1, b_1, c_1), (a_1, b_2, c_3), (a_2, b_3, c_3), (a_2, b_4, c_1), (a_3, b_1, c_2), (a_4, b_3, c_4)\}$$

be an instance for 3DM. The triple $(a_1, b_2, c_3), (a_2, b_4, c_1), (a_3, b_1, c_2), (a_4, b_3, c_4)$ form a 3-dimensional matching and thus a solution for 3DM.

It follows the construction of an instance for 3DkM for $k = 4$ as defined in the proof of Theorem 1. Integer n has to be a multiple of $k-1 = 3$. To ensure this, A is extended by a_5 and a_6, B is extended by b_5 and b_6, C is extended by c_5 and c_6 and S is extended by (a_5, b_5, c_5) and (a_6, b_6, c_6). Now $n = 6$ and $r = \frac{n}{k-1} = 2$.

Then $A' = \{a_1, \ldots, a_{18}\}$, $B' = \{b_1, \ldots, b_{18}\}$, $C' = \{c_1, \ldots, c_{18}\}$, $R = \{(a_i, b_i, c_i) \mid i = 7, \ldots, 18\}$, $T' = T_{7,3} \cup T_{13,3}$ and $S' = S \cup R \cup T$, where, for example,

$$T_{7,3} = \left\{ \begin{matrix} (a_7, b_{10}), & (a_7, b_{11}), & (a_7, b_{12}), & (a_8, b_{10}), & (a_8, b_{11}), & (a_8, b_{12}), & (a_9, b_{10}), & (a_9, b_{11}), & (a_9, b_{12}), \\ (b_7, c_{10}), & (b_7, c_{11}), & (b_7, c_{12}), & (b_8, c_{10}), & (b_8, c_{11}), & (b_8, c_{12}), & (b_9, c_{10}), & (b_9, c_{11}), & (b_9, c_{12}), \\ (c_7, a_{10}), & (c_7, a_{11}), & (c_7, a_{12}), & (c_8, a_{10}), & (c_8, a_{11}), & (c_8, a_{12}), & (c_9, a_{10}), & (c_9, a_{11}), & (c_9, a_{12}) \end{matrix} \right\},$$

$$T_{13,3} = \left\{ \begin{matrix} (a_{13}, b_{16}), & (a_{13}, b_{17}), & (a_{13}, b_{18}), & (a_{14}, b_{16}), & (a_{14}, b_{17}), & (a_{14}, b_{18}), & (a_{15}, b_{16}), & (a_{15}, b_{17}), & (a_{15}, b_{18}), \\ (b_{13}, c_{16}), & (b_{13}, c_{17}), & (b_{13}, c_{18}), & (b_{14}, c_{16}), & (b_{14}, c_{17}), & (b_{14}, c_{18}), & (b_{15}, c_{16}), & (b_{15}, c_{17}), & (b_{15}, c_{18}), \\ (c_{13}, a_{16}), & (c_{13}, a_{17}), & (c_{13}, a_{18}), & (c_{14}, a_{16}), & (c_{14}, a_{17}), & (c_{14}, a_{18}), & (c_{15}, a_{16}), & (c_{15}, a_{17}), & (c_{15}, a_{18}) \end{matrix} \right\},$$

$$T = \left\{ \begin{matrix} (a_1, b_7, c_{10}), & (a_1, b_7, c_{11}), & (a_1, b_7, c_{12}), & (a_2, b_8, c_{10}), & (a_2, b_8, c_{11}), & (a_2, b_8, c_{12}), \\ (a_3, b_9, c_{10}), & (a_3, b_9, c_{11}), & (a_3, b_9, c_{12}), & (a_4, b_{13}, c_{16}), & (a_4, b_{13}, c_{17}), & (a_4, b_{13}, c_{18}), \\ (a_5, b_{14}, c_{16}), & (a_5, b_{14}, c_{17}), & (a_5, b_{14}, c_{18}), & (a_6, b_{15}, c_{16}), & (a_6, b_{15}, c_{17}), & (a_6, b_{15}, c_{18}), \\ \\ (a_7, b_{10}, c_1), & (a_7, b_{11}, c_1), & (a_7, b_{12}, c_1), & (a_8, b_{10}, c_2), & (a_8, b_{11}, c_2), & (a_8, b_{12}, c_2), \\ (a_9, b_{10}, c_3), & (a_9, b_{11}, c_3), & (a_9, b_{12}, c_3), & (a_{10}, b_1, c_7), & (a_{10}, b_2, c_8), & (a_{10}, b_3, c_9), \\ (a_{11}, b_1, c_7), & (a_{11}, b_2, c_8), & (a_{11}, b_3, c_9), & (a_{12}, b_1, c_7), & (a_{12}, b_2, c_8), & (a_{12}, b_3, c_9), \\ \\ (a_{13}, b_{16}, c_4), & (a_{13}, b_{17}, c_4), & (a_{13}, b_{18}, c_4), & (a_{14}, b_{16}, c_5), & (a_{14}, b_{17}, c_5), & (a_{14}, b_{18}, c_5), \\ (a_{15}, b_{16}, c_6), & (a_{15}, b_{17}, c_6), & (a_{15}, b_{18}, c_6), & (a_{16}, b_4, c_{13}), & (a_{16}, b_5, c_{14}), & (a_{16}, b_6, c_{15}), \\ (a_{17}, b_4, c_{13}), & (a_{17}, b_5, c_{14}), & (a_{17}, b_6, c_{15}), & (a_{18}, b_4, c_{13}), & (a_{18}, b_5, c_{14}), & (a_{18}, b_6, c_{15}) \end{matrix} \right\},$$

see also Fig. 1.

Theorem 2. *k-MD is NP-complete for bipartite graphs G and each $k \geq 2$.*

Proof. The k-MD problem is obviously in NP, because it can be checked in polynomial time whether a set of vertices is a k-resolving set.

The NP-hardness is proven by a reduction from 3D(k-1)M. Let $A = \{a_1, \ldots, a_n\}$, $B = \{b_1, \ldots, b_n\}$, $C = \{c_1, \ldots, c_n\}$, $S = \{s_1, \ldots, s_m\}$ be an instance I for 3D$(k-1)$M where $k \geq 2$ and $n > k$. The aim is to define a graph $G = (V, E)$ and a number x such that G has a k-resolving set of size x if and only if instance I has a (k-1)-matching.

Graph G is defined as follows, see also Fig. 2. It has a vertex a_i, b_i and c_i for $i = 1, \ldots, n$ and a vertex s_i for $i = 1, \ldots, m$. Graph G additionally contains vertices denoted by $a_0, b_0, c_0, v_0, v_A, v_B, v_C$ and $d_1, \ldots, d_{m'}$ where $m' = \lceil \log(m) \rceil$.

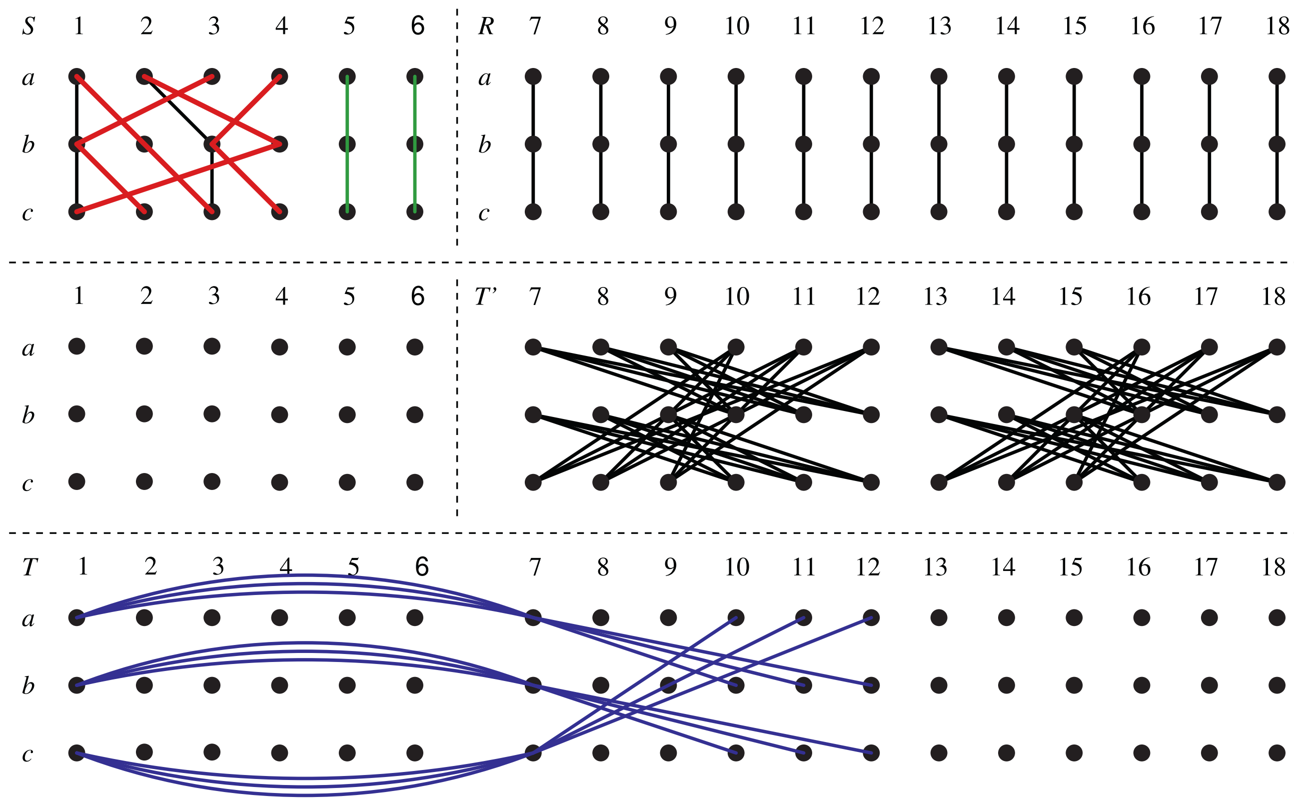


Fig. 1. This graphic illustrates the transformation from 3DM to 3DkM for $k = 4$ as explained in Example 1. The drawing on the top left visualizes an instance with 6 triples in S that cover the elements $\{a_1, \ldots, a_4, b_1, \ldots, b_4, c_1, \ldots, c_4\}$. The triples are indicated by 6 red and 2 black lines, each covering 3 elements. Set S contains a matching indicated by the red lines. Each set A, B and C is extended by two element a_5, a_6, b_5, b_6 and c_5, c_6 respectively, and set S is extended by two triples (a_5, b_5, c_5), (a_6, b_6, c_6), such that the number of elements in the new sets A, B and C is a multiple of $(k-1) = 3$. These two triples are indicated by green lines. The drawing in the middle right visualizes the $2 \cdot 6 = 12$ triples of R indicated by black lines. The drawing at the bottom visualizes the 54 tuples of $T' = T_{7,3} \cup T_{13,3}$, also indicated by black lines, each covering 2 elements. The set T is formed from set T' by adding each element of A, B and C to $k-1 = 3$ tuples of T'. For the sake of clarity, only the triples from T for the elements a_1, b_1 and c_1 are shown in the figure. These triples are indicated by blue lines. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

1. Each vertex a_i, $0 \le i \le n$, is connected with
   (a) vertex v_A,
   (b) vertex v_0, and
   (c) vertex s_j, $1 \le j \le m$ if and only if triple s_j contains element a_i.
2. Each vertex b_i, $0 \le i \le n$, is connected with
   (a) vertex v_B,
   (b) vertex v_0, and
   (c) vertex s_j, $1 \le j \le m$ if and only if triple s_j contains element b_i.
3. Each vertex c_i, $0 \le i \le n$, is connected with
   (a) vertex v_C,
   (b) vertex v_0, and
   (c) vertex s_j, $1 \le j \le m$ if and only if triple s_j contains element c_i.
4. Each vertex d_i, $1 \le i \le m'$, is connected with
   (a) vertex v_0 and
   (b) vertex s_j, $1 \le j \le m$, if and only if the i-th bit of the binary representation of j is 1.

Graph G additionally contains so-called *leg vertices*. These leg vertices form paths (*legs*) with $\lceil k/2 \rceil$ or $\lfloor k/2 \rfloor$ vertices. Two such legs, one with $\lceil k/2 \rceil$ vertices and one with $\lfloor k/2 \rfloor$ vertices, are attached to each vertex of $L_{\text{root}} = \{v_A, v_B, v_C, v_0, d_1, \ldots, d_{m'}\}$, see Fig. 2. Set L_{root} consists of the *root vertices* of the legs. Let L_v be the set of vertices of the two legs at vertex v and

$$L = L_{v_A} \cup L_{v_B} \cup L_{v_C} \cup L_{v_0} \cup L_{d_1} \cup \cdots \cup L_{d_{m'}}$$

be the set of all leg vertices of G. Set L_{root} has $4 + m'$ vertices, each set L_v, $v \in L_{\text{root}}$, has k vertices and L has $(4 + m')k$ vertices.

The graph G can obviously be constructed in polynomial time from instant I.

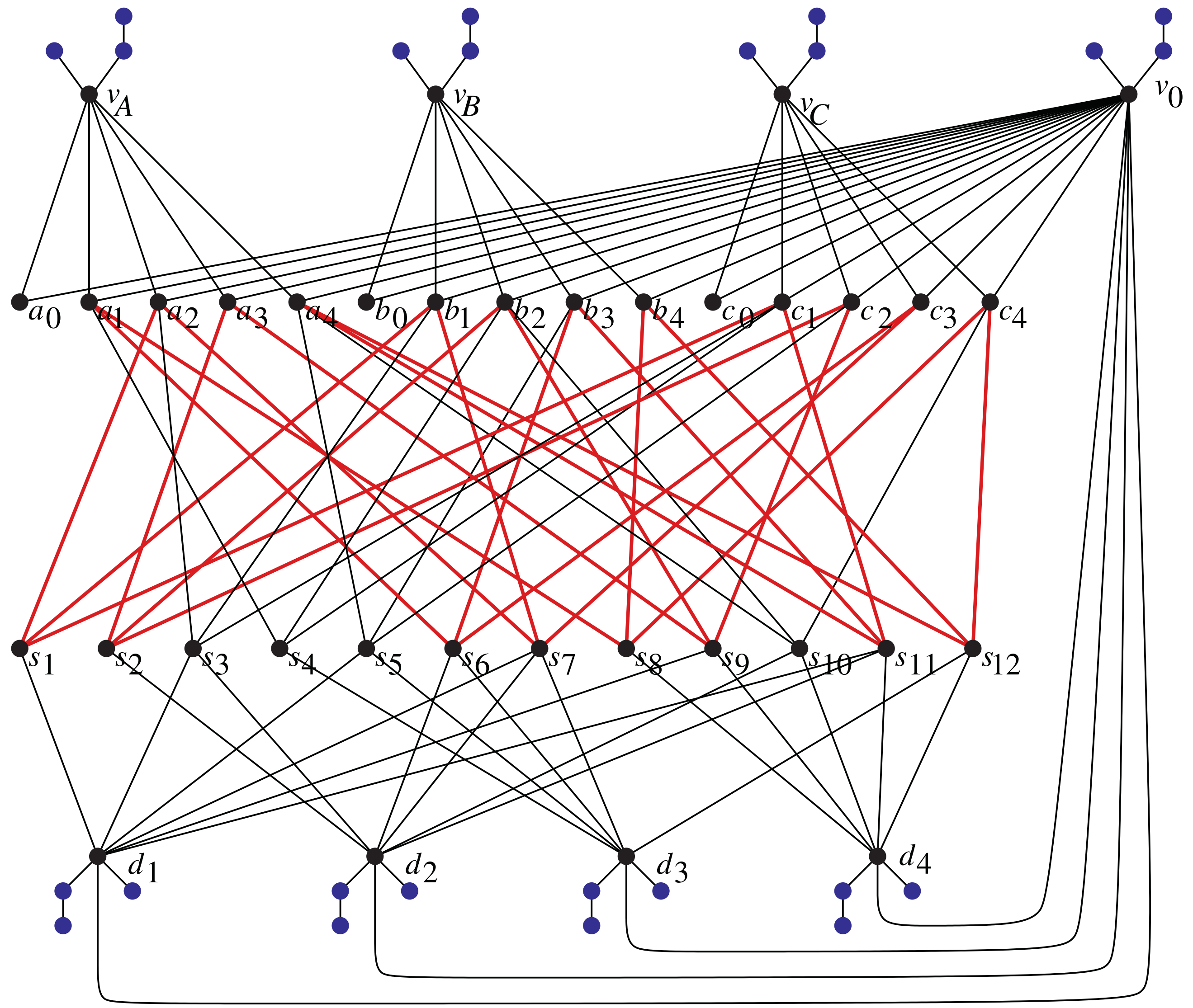


Fig. 2. This graphic illustrates the transformation from 3D2M to 3-MD. The Instance I consisting of $A = \{a_1, \dots, a_4\}$, $B = \{b_1, \dots, b_4\}$, $C = \{c_1, \dots, c_4\}$, $S = \{s_1, \dots, s_{12}\}$ with $s_1 = (a_2, b_1, c_1)$, $s_2 = (a_3, b_2, c_2)$, $s_3 = (a_2, b_1, c_1)$, $s_4 = (a_1, b_2, c_1)$, $s_5 = (a_4, b_3, c_2)$, $s_6 = (a_1, b_3, c_3)$, $s_7 = (a_2, b_1, c_3)$, $s_8 = (a_1, b_4, c_4)$, $s_9 = (a_3, b_2, c_2)$, $s_{10} = (a_4, b_2, c_4)$, $s_{11} = (a_4, b_3, c_1)$, $s_{12} = (a_4, b_4, c_4)$ for 3D2M is transformed into the graph G and $x = (4+4)3 + 3 + (3-1)n = 35$. The set of triples $M = \{s_1, s_2, s_6, s_7, s_8, s_9, s_{11}, s_{12}\}$, indicated in the figure by the red lines, is a 2-matching for instance I, where $L \cup \{a_0, b_0, c_0\} \cup M$ is a 3-resolving set for G of size x. Set L is the set of vertices of the legs attached at the vertices $v_A, v_B, v_C, v_0, d_1, d_2, d_3, d_4$. In the figure, the vertices of L are colored blue. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

First of all, let us note some properties of G.

- P1: G is bipartite.
- P2: The distance between
  - (a) two vertices of $\{v_B, v_B, v_C\}$ is 4,
  - (b) two vertices of $\{d_1, \dots, d_{m'}\}$ is 2,
  - (c) a vertex of $\{v_B, v_B, v_C\}$ and a vertex of $\{d_1, \dots, d_{m'}\}$ is 3,
  - (d) vertex v_0 and a vertex of $\{v_B, v_B, v_C\}$ is 2, and
  - (e) vertex v_0 and a vertex of $\{d_1, \dots, d_{m'}\}$ is 1.
- P3: Every k-resolving set for G contains all vertices of L. This follows from the observation that for each vertex $v \in L_{\text{root}}$ the two vertices of L_v adjacent to v are only resolved by the k vertices of L_v.

Now we will prove that S has a (k-1)-matching for instance I if and only if G has a resolving set of size

$$x = (4 + m')k + 3 + (k - 1)n.$$

"$\Rightarrow$:" Let $M \subseteq S$ be a (k-1)-matching for instance I. The aim is to show that

$$R = L \cup \{a_0, b_0, c_0\} \cup M$$

is a k-resolving set for G of size

$$x = (4 + m')k + 3 + (k - 1)n.$$

That is, each pair of two distinct vertices u_1, u_2 of G is resolved by at least k distinct vertices of U. Here the triple s_j of M are considered as vertices of G.

Consider the following case distinctions for two vertices u_1 and u_2.

1. $u_1, u_2 \in L_v,\ v \in L_{\mathrm{root}}$.
   (a) $d(u_1, v) = d(u_2, v)$. Each of the k vertices of L_v resolves u_1 and u_2.
   (b) $d(u_1, v) \neq d(u_2, v)$. Each of the k vertices of $L_{v'},\ v' \in L_{\mathrm{root}} \setminus \{v\}$, resolves u_1 and u_2.
2. $u_1 \in L_{v_1},\ u_2 \in L_{v_2},\ v_1, v_2 \in L_{\mathrm{root}},\ v_1 \neq v_2$, and $d(u_1, v_1) \leq d(u_2, v_2)$. Each of the k vertices of L_{v_1} resolves u_1 and u_2.

Up to this point all pairs of vertices u_1, u_2 are considered of which both are in L.

3. $u_1 \in L_{v_A} \cup L_{v_B} \cup L_{v_C}$ and $u_2 \notin L$. Each of the k vertices of L_{v_0} resolves u_1 and u_2.
4. $u_1 \in L_{d_1} \cup \cdots \cup L_{d_{m'}}$ and $u_2 \notin L$.
   (a) $u_2 \notin \{v_B, v_C\}$. Each of the k vertices of L_{v_A} resolves u_1 and u_2.
   (b) $u_2 \notin \{v_A, v_C\}$. Each of the k vertices of L_{v_B} resolves u_1 and u_2.
   (c) $u_2 \notin \{v_A, v_B\}$. Each of the k vertices of L_{v_C} resolves u_1 and u_2.
5. $u_1 \in L_{v_0}$ and $u_2 \notin L$.
   (a) $u_2 \in \{v_A, a_0, \ldots, a_n\}$. Each of the k vertices of L_{v_A} resolves u_1 and u_2.
   (b) $u_2 \in \{v_B, b_0, \ldots, b_n\}$. Each of the k vertices of L_{v_B} resolves u_1 and u_2.
   (c) $u_2 \in \{v_C, c_0, \ldots, c_n\}$. Each of the k vertices of L_{v_C} resolves u_1 and u_2.
   (d) $u_2 \in \{d_i\} \cup \{s_j \mid \text{the } i\text{-th bit in the binary representation of } j \text{ is } 1\}$. Each of the k vertices of L_{d_i} resolves u_1 and u_2.

Up to this point all pairs of vertices u_1, u_2 are considered of which at least one of them is in L.

6. $u_1 \in L_{\mathrm{root}}$ and $u_2 \notin L$. Each of the k vertices of L_{u_1} resolves u_1 and u_2.

   Up to this point all pairs of vertices u_1, u_2 are considered of which at least one of them is in $L \cup L_{\mathrm{root}}$.

7. $u_1 = s_{i_1} \in \{s_1, \ldots, s_{m'}\}$ and $u_2 \notin L \cup L_{\mathrm{root}}$.
   (a) $u_2 = s_{i_2} \in \{s_1, \ldots, s_{m'}\}$. Each of the k vertices of L_{d_j} resolves u_1 and u_2, if the binary representation of i_1 and i_2 differs in position j.
   (b) $u_2 \in \{a_0, \ldots, a_n\}$, $u_2 \in \{b_0, \ldots, b_n\}$, or $u_2 \in \{c_0, \ldots, c_n\}$. Each of the k vertices of L_{v_A}, L_{v_B}, or L_{v_C}, respectively, resolves u_1 and u_2.

Up to this point all pairs of vertices u_1, u_2 are considered of which at least one of them is in $L \cup L_{\mathrm{root}} \cup \{s_1, \ldots, s_{m'}\}$.

8. $u_1 \in \{a_1, \ldots, a_n\}$ and $u_2 \notin L \cup L_{\mathrm{root}} \cup \{s_1, \ldots, s_{m'}\}$.
   (a) $u_2 \in \{b_0, \ldots, b_n, c_0, \ldots, c_n\}$. Each of the k vertices of L_{v_A} resolves u_1 and u_2.
   (b) $u_2 \in \{a_1, \ldots, a_n\}$. Each vertex s_i for which triple s_i contains u_1 or u_2 resolves u_1 and u_2. There are $2(k-1) \geq k$ such vertices for $k \geq 2$.
   (c) $u_2 = a_0$. Each vertex s_i for which triple s_i contains u_1 resolves u_1 and u_2, and vertex a_0 resolves u_1 and u_2. Altogether these are exactly $(k-1) + 1 = k$ vertices.
9. $u_1 \in \{b_1, \ldots, b_n\}$ and $u_2 \notin L \cup L_{\mathrm{root}} \cup \{s_1, \ldots, s_{m'}\}$ (analogous to case 8).
10. $u_1 \in \{c_1, \ldots, c_n\}$ and $u_2 \notin L \cup L_{\mathrm{root}} \cup \{s_1, \ldots, s_{m'}\}$ (analogous to case 8).
11. $u_1, u_2 \in \{a_0, b_0, c_0\}$. Each of the k vertices of L_{v_A}, L_{v_B} or L_{v_C} resolves u_1 and u_2.

Now all pairs of vertices u_1, u_2 of G are considered and it is shown that all of them are resolved by at least k vertices from R. Note that only the vertex pairs $u_1, u_2 \in \{a_0, \ldots, a_n\}$, $u_1, u_2 \in \{b_0, \ldots, b_n\}$ and $u_1, u_2 \in \{c_0, \ldots, c_n\}$ are not already resolved by k vertices of L. Strictly speaking, not a single vertex from $L \cup \{v_A, v_B, v_C, v_0, d_1, \ldots, d_{m'}\}$ resolves such a pair of vertices.

"$\Leftarrow$:" Let $R \subseteq V$ be a k-resolving set for G with $x = (4 + m')k + 3 + (k-1)n$ vertices. By Property P3, R contains all the $(4+m)'k$ vertices of L. This leaves $3 + (k-1)n$ vertices of R that are not in L. Let us now consider the vertex pairs a_0, a_i, and b_0, b_i, and c_0, c_i for $i = 1, \ldots, n$. The vertices of L and the vertices of $\{v_A, v_B, v_C, v_0, d_1, \ldots, d_{m'}\}$ do not resolve these vertex pairs. The only way to resolve these $3n$ vertex pairs at least k times with $3 + (k-1)n$ vertices for $n > k \geq 2$, is to use $k-1$ vertices from $\{s_1, \ldots, s_m\}$ that form a $k-1$ matching and the three vertices a_0, b_0, c_0. This is the point where it is necessary that n is greater than k. □

In the introduction of this paper, we mentioned that k-Metric Dimension and (k,t)-Metric Dimension of [21] are equivalent if t is greater than or equal to the diameter of G. Since the constructed graph in Theorem 2 has diameter $2 \cdot \lceil k/2 \rceil + 3$, Theorem 2 also proves the NP-completeness of (k,t)-Metric Dimension for each $k \geq 2$ and $t \geq 2 \cdot \lceil k/2 \rceil + 3$, even for bipartite graphs.

Data availability

No data was used for the research described in the article.

References

[1] I.G. Yero, A. Estrada-Moreno, J.A. Rodríguez-Velázquez, Computing the k-metric dimension of graphs, Appl. Math. Comput. 300 (2017) 60–69, doi:10.1016/j.amc.2016.12.005.
[2] P.J. Slater, Leaves of trees, Congressum Numerant. 14 (1975) 549–559.
[3] F. Harary, R.A. Melter, On the metric dimension of a graph, Ars Combinatoria 2 (1976) 191–195.
[4] Z. Beerliova, F. Eberhard, T. Erlebach, A. Hall, M. Hoffmann, M. Mihaľák, L.S. Ram, Network discovery and verification, in: D. Kratsch (Ed.), Graph-Theoretic Concepts in Computer Science, Springer, Berlin Heidelberg, 2005, pp. 127–138, doi:10.1007/11604686_12.
[5] K. Liu, N.B. Abu-Ghazaleh, Virtual coordinates with backtracking for void traversal in geographic routing, in: Ad-Hoc, Mobile, and Wireless Networks, 5th International Conference, ADHOC-NOW 2006, 2006, pp. 46–59, doi:10.1007/11814764_6. Ottawa, Canada, August 17-19, 2006, Proceedings
[6] A. Sebö, E. Tannier, On metric generators of graphs, Math. Oper. Res. 29 (2) (2004) 383–393, doi:10.1287/moor.1030.0070.
[7] S. Hoffmann, E. Wanke, Metric dimension for gabriel unit disk graphs is np-complete, in: Algorithms for Sensor Systems, 8th International Symposium on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile Entities, ALGOSENSORS 2012, 2012, pp. 90–92, doi:10.1007/978-3-642-36092-3_10. Ljubljana, Slovenia, September 13-14, 2012. Revised Selected Papers
[8] S. Khuller, B. Raghavachari, A. Rosenfeld, Landmarks in graphs, Discrete Appl. Math. 7 (3) (1996) 217–229, doi:10.1016/0166-218X(95)00106-2.
[9] G. Chartrand, L. Eroh, M.A. Johnson, O. Oellermann, Resolvability in graphs and the metric dimension of a graph, Discrete Appl. Math. 105 (1–3) (2000) 99–113, doi:10.1016/S0166-218X(00)00198-0.
[10] S. Hayat, Computing distance-based topological descriptors of complex chemical networks: new theoretical techniques, Chem. Phys. Lett. 688 (1) (1977) 51–58, doi:10.1016/j.cplett.2017.09.055.
[11] M.C. Hernando, M. Mora, I.M. Pelayo, C. Seara, J. Cáceres, M.L. Puertas, On the metric dimension of some families of graphs, Electron. Note. Discrete Math. 22 (2005) 129–133, doi:10.1016/j.endm.2005.06.023.
[12] R.A. Melter, I. Tomescu, Metric bases in digital geometry, Comput. Vis. Graphic. Image Process. 25 (1) (1984) 113–121, doi:10.1016/0734-189X(84)90051-3.
[13] M. Bača, E.T. Baskoro, A.N.M. Salman, S.W. Saputro, D. Suprijanto, The metric dimension of regular bipartite graphs, Bulletin mathématiques de la Société des sciences mathématiques de Roumanie 54 (1) (2011) 15–28.
[14] H. Iswadi, E.T. Baskoro, A. Salman, R. Simanjuntak, The metric dimension of amalgamation of cycles, Far East J. Math. Sci. (FJMS) 41 (1) (2010) 19–31.
[15] J. Díaz, O. Pottonen, M.J. Serna, E.J. van Leeuwen, On the complexity of metric dimension, in: Algorithms - ESA 2012 - 20th Annual European Symposium, 2012, pp. 419–430, doi:10.1007/978-3-642-33090-2_37. Ljubljana, Slovenia, September 10-12, 2012. Proceedings
[16] M. Hauptmann, R. Schmied, C. Viehmann, Approximation complexity of metric dimension problem, J. Discrete Algor. 14 (2012) 214–222, doi:10.1016/j.jda.2011.12.010.
[17] G.G. Chappell, J.G. Gimbel, C. Hartman, Bounds on the metric and partition dimensions of a graph, Ars Combinatoria 88 (2008).
[18] G. Chartrand, C. Poisson, P. Zhang, Resolvability and the upper dimension of graphs, Comput. Math. Appl. 39 (12) (2000) 19–28.
[19] A. Estrada-Moreno, J.A. Rodríguez-Velázquez, I.G. Yero, The k-metric dimension of a graph, Appl. Math. Inf. Sci. 9 (6) (2013) 2829–2840, doi:10.12785/amis/090609.
[20] M.C. Hernando, P.J.S. M. Mora, D.R. Wood, Fault-tolerant metric dimension of graphs, Convex. Discrete Struct. 5 (2008) 81–85.
[21] A. Estrada-Moreno, I. Yero, J. Rodríguez-Velázquez, On the (k, t)-metric dimension of graphs, Comput. J. (2016).
[22] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, 1979.

9.2 The directed metric dimension of directed co-graphs

The directed metric dimension of directed co-graphs
Yannick Schmitz, Egon Wanke
Eingereicht in: Journal of Graph Algorithms and Applications

Diese Arbeit enthält einen Algorithmus zur Berechnung der gerichteten Metrischen Dimension gerichteter Co-Graphen in linearer Zeit und einen NP-Vollständigkeitsbeweis für die gerichtete Metrische Dimension auf DAGs. Die Ergebnisse sind in Kapitel 5 zu finden. Alle Autoren haben gleichermaßen zur Erarbeitung und zur Formulierung der Ergebnisse beigetragen. Den Hauptteil meines Beitrags bilden die Ergebnisse zur Bestimmung der minimalen trennenden Mengen aus den Teillösungen sowie der NP-Vollständikeitsbeweis für DAGs.

The directed metric dimension of directed co-graphs

Yannick Schmitz and Egon Wanke

Heinrich-Heine-Universität Düsseldorf, Germany
yannick.schmitz@hhu.de, egon.wanke@hhu.de

Abstract

A vertex w *resolves* two vertices u and v in a directed graph G if the distance from w to u is different to the distance from w to v. A set of vertices R is a *resolving set* for a directed graph G if for every pair of vertices u, v which are not in R there is at least one vertex in R that resolves u and v in G. The *directed metric dimension* of a directed graph G is the size of a minimum resolving set for G. The decision problem Directed Metric Dimension for a given directed graph G and a given number k is the question whether G has a resolving set of size at most k. In this paper, we study directed co-graphs. We introduce a linear time algorithm for computing a minimum resolving set for directed co-graphs and show that Directed Metric Dimension already is NP-complete for directed acyclic graphs.

1 Introduction

The metric dimension of graphs is originally defined for undirected graphs. In an undirected graph G, two vertices u and v are *resolved* by a vertex w if the distance between w and u differs from the distance between w and v. In directed graphs, two vertices u and v are *resolved* by a vertex w if the distance from w to u differs from the distance from w to v. In the undirected case as well as in the directed case, a set of vertices R is called a *resolving set* for G if for every pair of vertices u, v which are not in R there is at least one vertex w in R resolving u and v. The *metric dimension* of G is the size of a smallest resolving set for G.

The metric dimension of undirected graphs has been introduced in the 1970s independently by Slater [Sla75] and by Harary and Melter [HM76]. It finds applications in various areas, including network discovery and verification [BEE+05], geographical routing protocols [LA06], combinatorial optimisation [ST04], sensor networks [HW12], robot navigation [KRR96], and chemistry [CEJO00, Hay77].

Deciding whether a given graph G has metric dimension $\leq k$ is NP-complete for undirected and directed graphs, see [GJ79, KRR96]. There are several algorithms for computing a minimum resolving set in polynomial time for special classes of undirected graphs, as for example for trees [CEJO00, KRR96], wheels [HMP+05], grid graphs [MT84], k-regular bipartite graphs [BBS+11], amalgamation of cycles [IBSS10], and outerplanar graphs [DPSL12]. The approximability of the metric dimension has been studied for bounded degree, dense, and general graphs in [HSV12]. Upper and lower bounds on the metric dimension are considered in [CGH08, CPZ00] for further classes of undirected graphs. In the undirected case, several variants of the metric dimension have been studied, which are usually NP-complete for general graphs as well [OP07, FR18, SVW21].

A natural way of generalising graph theoretical problems is to consider their directed counterparts. In the context of the metric dimension of graphs, this was first considered by Chartrand, Rains, and Zhang in [CRZK00], before receiving further consideration in several other papers, see [FGO06, FXW13, Loz13, RRCM14].

In this paper, we study directed co-graphs [BDGR97] and introduce a linear time algorithm for computing minimum resolving sets for directed co-graphs in linear time. We also show that Directed Metric Dimension already is NP-complete for directed acyclic graphs.

1.1 The directed metric dimension

All graphs in this paper are finite and simple. For a graph $G = (V, E)$ with vertex set V and edge set E, we write $V(G)$ for vertex set V and $E(G)$ for edge set E to reduce the number of variable names and indices when using several graphs. For two vertices $u, v \in V(G)$ the distance $d_G(u, v)$ from u to v is the length (number of edges) of a shortest path from u to v. If there is no such path from u to v, then $d_G(u, v)$ is not defined. A directed graph G is *strongly connected* if for each pair of vertices u and v, there is a path from u to v and a path from v to u.

The *metric dimension* of a directed graph can be defined in the same way as for undirected graphs, see Figure 1 for an example.

Definition 1 (Directed metric dimension)**.** *Two distinct vertices u and v of a directed graph G are* resolved *by a vertex w if*

1. *$w = u$,*
2. *$w = v$, or*
3. *there is a path from w to u and a path from w to v such that $d_G(w, u) \neq d_G(w, v)$.*

A set of vertices $R \subseteq V(G)$ is called a resolving set *of G if for every pair of vertices $u, v \in V(G)$ there is at least one vertex w in R resolving u and v. The* directed metric dimension *of G is the size of a minimum resolving set for G.*

Note that it is also possible to consider the distances $d_G(u, w)$ from each vertex to the vertices in R instead of the distances $d_G(w, u)$, but both definitions are equivalent if every edge (u, v) in G is replaced by an edge (v, u). Also note that if $d_G(w, u)$ is undefined, it can not be compared to any other distance $d_G(w, v)$.

The decision problem Directed Metric Dimension it defined as follows.

Directed Metric Dimension	
Instance:	A directed graph G and a number k.
Question:	Is there a resolving set $R \subseteq V(G)$ for G of size at most k?

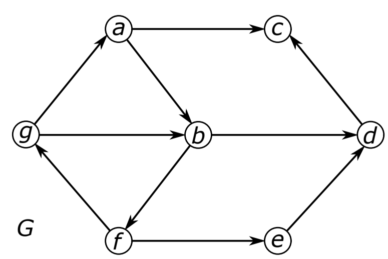


Figure 1: Vertex set $\{a, f, g\}$ is a minimum resolving set for directed graph G. Vertex a resolves the vertex pairs a, b, a, c, a, d, a, e, a, f, a, g, b, d, b, e, b, f, b, g, c, d, c, e, c, f, c, g, d, e, d, g, f, e, and f, g in G, vertex f resolves the vertex pairs a, c, a, e, a, f, a, g, b, c, b, e, b, f, b, g, c, d, c, e, c, f, c, g, d, e, d, f, d, g, e, f, and f, g in G, and vertex g resolves the vertex pairs a, c, a, d, a, e, a, f, a, g, b, c, b, d, b, e, b, f, b, g, c, e, c, g, d, e, d, g, e, f, e, g, and f, g in G.

The following notion of a vertex set R that resolves only the vertex pairs of a vertex set $U \subseteq V(G)$ is frequently used in the next section.

Notation 1. *Let $U \subseteq V(G)$ be a set of vertices. A set of vertices $R \subseteq U$* resolves a vertex set U in G*, or in other words,* R is a resolving set for U in G*, if for each pair of vertices $u, v \in U$ there is at least one vertex w in R that resolves u and v in G.*

If R resolves the vertices of U in G, then the necessary shortest paths with different distances $d_G(w,u)$ and $d_G(w,v)$ are paths in graph G and do not need to be paths in the subgraph of G induced by U. For example, in Figure 1 vertex set $R = \{\mathsf{a}\}$ is a resolving set for $U = \{\mathsf{c}, \mathsf{d}, \mathsf{g}\}$ in G.

2 Directed metric dimension of directed co-graphs

In this section, we show how to compute a minimum resolving set for directed co-graphs in linear time. In Definition 2, co-graphs and co-trees are defined step by step simultaneously. We use variable names with a hat symbol for nodes in trees to distinguish them more clearly from vertices in graphs.

Definition 2 (**Directed co-graphs and co-trees**).

- *A directed graph G that consists of a single vertex u is a directed co-graph.*

  The co-tree T of G consists of a single node $\hat{u}$ associated with vertex u of G. Node $\hat{u}$ is the root *of T. Let* $\text{vertex}(\hat{u}) := u$ *and* $\text{node}(u) := \hat{u}$*. The notation* $\text{vertex}(\hat{u})$ *is only defined for leaves $\hat{u}$ of T.*

- *If G_1 and G_2 are two directed co-graphs, then the* disjoint union*,* join*, or* directed join *of G_1 and G_2, denoted by $G_1 \cup G_2$, $G_1 \times G_2$, and $G_1 \gg G_2$, respectively, is a directed co-graph G with vertex set $V(G_1) \cup V(G_2)$ and edge set*

  $$\begin{array}{l} E(G_1) \cup E(G_2), \\ E(G_1) \cup E(G_2) \cup \{(u,v),(v,u) \mid u \in V(G_1),\ v \in V(G_2)\}, \textit{and} \\ E(G_1) \cup E(G_2) \cup \{(u,v) \mid u \in V(G_1),\ v \in V(G_2)\},\ \textit{respectively.} \end{array}$$

  Let T_1 and T_2 be the co-trees of G_1 and G_2 with root $\hat{l}$ and $\hat{r}$, respectively. The co-tree T of G is the disjoint union of T_1 and T_2 with an additional node $\hat{u}$ and two additional edges $\{\hat{u}, \hat{l}\}$ and $\{\hat{u}, \hat{r}\}$. Node $\hat{u}$ is the root *of T labelled by $\cup$, $\times$, and $\gg$, respectively. Node $\hat{l}$ is the* left successor node *of $\hat{u}$, also denoted by* $\text{left}(\hat{u})$*, and node $\hat{r}$ is the* right successor node *of $\hat{u}$, also denoted by* $\text{right}(\hat{u})$*. Node $\hat{u}$ is the* predecessor node *of $\hat{l}$ and $\hat{r}$.*

The nodes of T that are not leaves are called inner nodes *of T. If $\hat{u}$ is an inner node of T labelled by $\cup$ or $\times$ and one of its two successor nodes* $\text{left}(\hat{u})$ *and* $\text{right}(\hat{u})$ *is a leaf and the other one is not a leaf, then without loss of generality, we assume that successor node* $\text{right}(\hat{u})$ *is the leaf.*

Figure 2 shows an example of a directed co-graph and its co-tree. Directed co-graphs can be recognised in linear time [CP06]. This includes the computation of the co-tree. In a strongly connected directed co-graph for each pair of vertices $u, v \in V(G)$ the distance $d_G(u,v)$ is at most 2.

For a directed co-graph G with co-tree T and a node $\hat{w} \in V(T)$ let $T_{\hat{w}}$ be the subtree of T rooted at $\hat{w}$ and let $G_{\hat{w}}$ be the subgraph of G induced by the vertices $\text{vertex}(\hat{u})$ of the leaves $\hat{u}$ of $T_{\hat{w}}$.

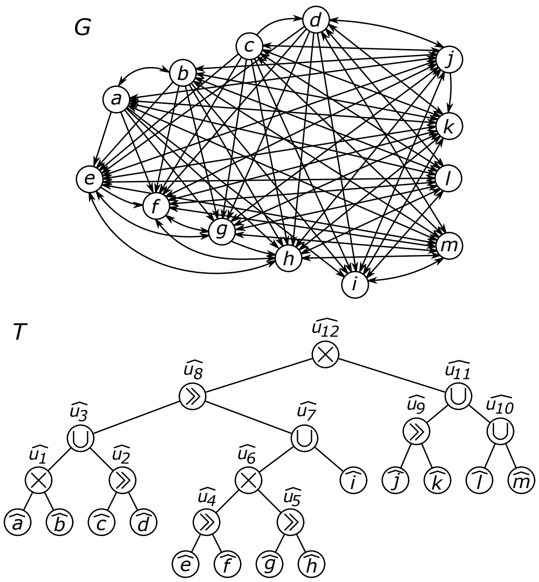


Figure 2: A directed co-graph G and its co-tree T.

Lemma 1. *Let G be a strongly connected directed co-graph with co-tree T and $\hat{w} \in V(T)$ be an inner node of T. There is no vertex $w \in V(G) \setminus V(G_{\hat{w}})$ that resolves two vertices of $V(G_{\hat{w}})$ in G.*

Proof. Since in each further composition of $G_{\hat{w}}$ all vertices of $V(G_{\hat{w}})$ get the same additional connections to vertices of $V(G) \setminus V(G_{\hat{w}})$, it follows that for each vertex $w \in V(G) \setminus V(G_{\hat{w}})$ and each vertex pair $v_1, v_2 \in V(G_{\hat{w}})$ either $d_G(w, v_1) = d_G(w, v_2) = 1$ or $d_G(w, v_1) = d_G(w, v_2) = 2$. Note that G is strongly connected and thus $d_G(u, v) \leq 2$ for each vertex pair $u, v \in V(G)$. $\square$

If R is a resolving set for G, then by Lemma 1 $R \cap V(G_{\hat{w}})$ is a resolving set for $V(G_{\hat{w}})$ in G. Note that $R \cap V(G_{\hat{w}})$ does not need to be a resolving set for $G_{\hat{w}}$, because $G_{\hat{w}}$ does not need to be strongly connected. For example, suppose G_1, G_2, G_3, and G_4 are four graphs with exactly one vertex u_1, u_2, u_3, and u_4, respectively. Then $\{u_1\}$ is a resolving set for $\{u_1, u_2, u_3\}$ in $G = ((G_1 \gg G_2) \cup G_3) \times G_4$, but not a resolving set for $G' = (G_1 \gg G_2) \cup G_3$, because there is no path from u_1 to u_3 in G'.

Our analysis of directed co-graphs requires the distinction between the following two different vertex types.

Definition 3. *Let G be a directed graph, $u, v \in V(G)$ be two distinct vertices of G, and $R \subseteq V(G)$ be a non-empty set of vertices. A vertex $u \in V(G) \setminus \{R\}$ is called a* distance 1 vertex *(or 1-vertex for short) w.r.t. R if $\forall w \in R : (w, u) \in E(G)$. It is called a* distance 2 vertex *(or 2-vertex for short) w.r.t. R if $\forall w \in R : (w, u) \notin E(G)$.*

Figure 3 shows an example of a 1-vertex and a 2-vertex w.r.t. a vertex set R.

If a vertex $u \in V(G) \setminus R$ is a 1-vertex or 2-vertex w.r.t. vertex set $R \subseteq V(G)$, then u obviously is a 1-vertex or 2-vertex, respectively, w.r.t. each non-empty subset R' of R. If R is a resolving set for G, then there is at most one 1-vertex w.r.t. R and at most one 2-vertex w.r.t. R, because two 1-vertices and two 2-vertices u, v w.r.t. R have the same distance $d_G(w, u_1) = d_G(w, u_2) = 1$ and $d_G(w, u_1) = d_G(w, u_2) = 2$, respectively, from each vertex $w \in R$.

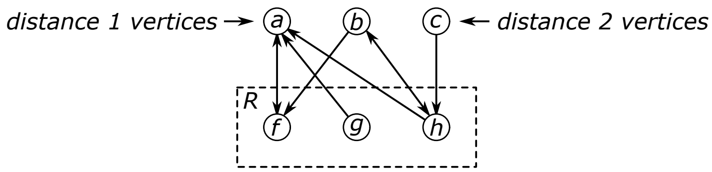


Figure 3: Vertex a is a 1-vertices w.r.t. $R = \{f, g, h\}$, vertex c is a 2-vertex w.r.t. R, and vertex b is neither a 1-vertex nor a 2-vertex w.r.t. R.

The algorithm for computing a minimum resolving set for a strongly connected directed co-graph G analyses the co-tree T for G bottom-up, see procedure **BottomUp**($\hat{w}$). For each inner node $\hat{w}$ of T, it computes one or two minimum resolving sets for $V(G_{\hat{w}})$ in G. The resolving sets for node $\hat{w}$ are build by the union of a resolving set $R_{\hat{l}}$ for $V(G_{\hat{l}})$ in G and a resolving set $R_{\hat{r}}$ for $V(G_{\hat{r}})$ in G, where $\hat{l}$ and $\hat{r}$ are the left and right successor nodes of $\hat{w}$ in T.

BottomUp($\hat{w}$)
  if ($\hat{w}$ *is an inner node of* T) **then**
    $\hat{l} \leftarrow$ left successor node of $\hat{w}$;
    BottomUp($\hat{l}$);
    $\hat{r} \leftarrow$ right successor node of $\hat{w}$;
    BottomUp($\hat{r}$);
    Merge($\hat{w}, \hat{l}, \hat{r}$);
  end

The next lemma specifies the conditions under which the minimum resolving sets for $V(G_{\hat{l}})$ and $V(G_{\hat{r}})$ in G only need to be joined to get a resolving set for $V(G_{\hat{w}})$ in G.

Lemma 2. *Let G be a strongly connected directed co-graph with co-tree T, $\hat{w} \in V(T)$ be an inner node of T, $\hat{l} = \mathrm{left}(\hat{w})$ be the left successor of $\hat{w}$ and $\hat{r} = \mathrm{right}(\hat{w})$ be the right successor of $\hat{w}$. If $\hat{l}$ is an inner node of T, then let $R_{\hat{l}} \subseteq V(G_{\hat{l}})$ be a resolving set for $V(G_{\hat{l}})$ in G. If $\hat{r}$ is an inner node of T, then let $R_{\hat{r}} \subseteq V(G_{\hat{r}})$ be a resolving set for $V(G_{\hat{r}})$ in G.*

We distinguish between the three labels $\cup$, $\times$ and $\gg$ for node $\hat{w}$ and the cases where the successor nodes of $\hat{w}$ are inner nodes or leaves.

1. *Let $\hat{w}$ be labelled by $\cup$.*

   (a) *Let $\hat{l}$ and $\hat{r}$ be inner nodes of T.*
   $R_{\hat{l}} \cup R_{\hat{r}}$ is a minimum resolving set for $V(G_{\hat{w}})$ in G iff $R_{\hat{l}}$ is a minimum resolving set for $V(G_{\hat{l}})$ in G, $R_{\hat{r}}$ is a minimum resolving set for $V(G_{\hat{r}})$ in G, and $G_{\hat{l}}$ has no 2-vertex w.r.t. $R_{\hat{l}}$ or $G_{\hat{r}}$ has no 2-vertex w.r.t. $R_{\hat{r}}$.

   (b) *Let $\hat{l}$ be an inner node of T and $\hat{r}$ be a leaf of T.*
   $R_{\hat{l}}$ is a minimum resolving set for $V(G_{\hat{w}})$ in G iff $R_{\hat{l}}$ is a minimum resolving set for $V(G_{\hat{l}})$ in G and $G_{\hat{l}}$ has no 2-vertex w.r.t. $R_{\hat{l}}$.

2. *Let $\hat{w}$ be labelled by $\times$.*

   (a) *Let $\hat{l}$ and $\hat{r}$ be inner nodes of T.*

$R_{\hat{l}} \cup R_{\hat{r}}$ is a minimum resolving set for $V(G_{\hat{w}})$ in G iff $R_{\hat{l}}$ is a minimum resolving set for $V(G_{\hat{l}})$ in G, $R_{\hat{r}}$ is a minimum resolving set for $V(G_{\hat{r}})$ in G, and $G_{\hat{l}}$ has no 1-vertex w.r.t. $R_{\hat{l}}$ or $G_{\hat{r}}$ has no 1-vertex w.r.t. $R_{\hat{r}}$.

(b) *Let $\hat{l}$ be an inner node of T and $\hat{r}$ be a leaf of T.*

$R_{\hat{l}}$ is a minimum resolving set for $V(G_{\hat{w}})$ in G iff $R_{\hat{l}}$ is a minimum resolving set for $V(G_{\hat{l}})$ in G and $G_{\hat{l}}$ has no 1-vertex w.r.t. $R_{\hat{l}}$.

3. *Let $\hat{w}$ be labelled by $\gg$.*

(a) *Let $\hat{l}$ and $\hat{r}$ be inner nodes of T.*

$R_{\hat{l}} \cup R_{\hat{r}}$ is a minimum resolving set for $V(G_{\hat{w}})$ in G iff $R_{\hat{l}}$ is a minimum resolving set for $V(G_{\hat{l}})$ in G, $R_{\hat{r}}$ is a minimum resolving set for $V(G_{\hat{r}})$ in G, and $G_{\hat{l}}$ has no 1-vertex w.r.t. $R_{\hat{l}}$ or $G_{\hat{r}}$ has no 2-vertex w.r.t. $R_{\hat{r}}$.

(b) *Let $\hat{l}$ be an inner node of T and $\hat{r}$ be a leaf of T.*

$R_{\hat{l}}$ is a minimum resolving set for $V(G_{\hat{w}})$ in G iff $R_{\hat{l}}$ is a minimum resolving set for $V(G_{\hat{l}})$ in G, and $G_{\hat{l}}$ has no 1-vertex w.r.t. $R_{\hat{l}}$.

(c) *Let $\hat{l}$ be a leaf of T and $\hat{r}$ be an inner node of T.*

$R_{\hat{r}}$ is a minimum resolving set for $V(G_{\hat{w}})$ in G iff $R_{\hat{r}}$ is a minimum resolving set for $V(G_{\hat{r}})$ in G, and $G_{\hat{r}}$ has no 2-vertex w.r.t. $R_{\hat{r}}$.

Proof. Since two distinct vertices from $V(G_{\hat{l}})\setminus R_{\hat{l}}$ as well as two distinct vertices from $V(G_{\hat{r}})\setminus R_{\hat{r}}$ are resolved in G by a vertex of $R_{\hat{r}}$ and a vertex of $R_{\hat{l}}$, respectively, we only need to consider the case in which one vertex is from $V(G_{\hat{l}}) \setminus R_{\hat{l}}$ and the other vertex is from $V(G_{\hat{r}}) \setminus R_{\hat{r}}$.

1. Let $\hat{w}$ be labelled by $\cup$.

In this case, for each vertex $u_l \in V(G_{\hat{l}}) \setminus R_{\hat{l}}$ and for each vertex $u_r \in V(G_{\hat{r}}) \setminus R_{\hat{r}}$, or $u_r = \text{vertex}(\hat{r})$ if $\hat{r}$ is a leaf in T, we have $d_G(u_l, u_r) = d_G(u_r, u_l) = 2$.

(a) Let $\hat{l}$ and $\hat{r}$ be inner nodes of T.

If u_l is not a 2-vertex w.r.t. $R_{\hat{l}}$, then there is vertex $v_l \in R_{\hat{l}}$ such that $d_G(v_l, u_l) = 1$. If u_r is not a 2-vertex w.r.t. $R_{\hat{r}}$, then there is a vertex $v_r \in R_{\hat{r}}$ such that $d_G(v_r, u_r) = 1$. Since $d_G(v_l, u_r) = 2$ and $d_G(v_r, u_l) = 2$, vertex u_l and vertex u_r are resolved in G by v_l or v_r. Thus $R_{\hat{l}} \cup R_{\hat{r}}$ is a resolving set for $V(G_{\hat{w}})$ in G. By Lemma 1, it is a minimum resolving set for $V(G_{\hat{w}})$ in G if $R_{\hat{l}}$ is a minimum resolving set for $V(G_{\hat{l}})$ in G and $R_{\hat{r}}$ is a minimum resolving set for $V(G_{\hat{l}})$ in G.

If u_l is a 2-vertex w.r.t. $R_{\hat{l}}$ and u_r is a 2-vertex w.r.t. $R_{\hat{r}}$, then for each vertex $v \in R_{\hat{l}} \cup R_{\hat{r}}$, $d_G(v, u_l) = d_G(v, u_r) = 2$. In this case, u_l and u_r are not resolved by any vertex of $R_{\hat{l}} \cup R_{\hat{r}}$ in G and thus $R_{\hat{l}} \cup R_{\hat{r}}$ is not a resolving set for $V(G_{\hat{w}})$ in G.

(b) Let $\hat{l}$ be an inner node of T and $\hat{r}$ be a leaf of T.

If u_l is not a 2-vertex w.r.t. $R_{\hat{l}}$, then there is a vertex $v_l \in R_{\hat{l}}$ such that $d_G(v_l, u_l) = 1$. Since $d_G(v_l, u_r) = 2$, vertex u_l and vertex u_r are resolved in G by v_l. Thus $R_{\hat{l}}$ is a resolving set for $V(G_{\hat{w}})$ in G. By Lemma 1, it is a minimum resolving set for $V(G_{\hat{w}})$ in G if $R_{\hat{l}}$ is a minimum resolving set for $V(G_{\hat{l}})$ in G.

If u_l is a 2-vertex w.r.t. $R_{\hat{l}}$, then for each vertex $v_l \in R_{\hat{l}}$, $d_G(v_l, u_l) = 2$. In this case, vertex u_l and vertex u_r are not resolved by any vertex of $R_{\hat{l}}$ and thus $R_{\hat{l}}$ is not a resolving set for $V(G_{\hat{w}})$ in G.

2. Let $\hat{w}$ be labelled by $\times$.

   The proof is running analogously to the proof for label $\cup$.

3. Let $\hat{w}$ be labelled by $\gg$.

   In this case, for each vertex $u_l \in V(G_{\hat{l}})$, or $u_l = \text{vertex}(\hat{l})$ if $\hat{l}$ is a leaf in T, and for each vertex $u_r \in V(G_{\hat{r}})$, or $u_r = \text{vertex}(\hat{r})$ if $\hat{r}$ is a leaf in T, we have $d_G(u_l, u_r) = 1$ and $d_G(u_r, u_l) = 2$.

   (a) Let $\hat{l}$ and $\hat{r}$ be inner nodes of T.

   If u_l is not a 1-vertex w.r.t. $R_{\hat{l}}$, then there is a vertex $v_l \in R_{\hat{l}}$ such that $d_G(v_l, u_l) = 2$. If u_r is not a 2-vertex w.r.t. $R_{\hat{r}}$, then there is a vertex $v_r \in R_{\hat{r}}$ such that $d_G(v_r, u_r) = 1$. Since $d_G(v_l, u_r) = 1$ and $d_G(v_r, u_l) = 2$, vertex u_l and vertex u_r are resolved in G by v_l or v_r. Thus $R_{\hat{l}} \cup R_{\hat{r}}$ is a resolving set for $V(G_{\hat{w}})$ in G. By Lemma 1, it is a minimum resolving set for $V(G_{\hat{w}})$ in G if $R_{\hat{l}}$ is a minimum resolving set for $V(G_{\hat{l}})$ in G and $R_{\hat{r}}$ is a minimum resolving set for $V(G_{\hat{l}})$ in G.

   If u_l is a 1-vertex w.r.t. $R_{\hat{l}}$ and u_r is a 2-vertex w.r.t. $R_{\hat{r}}$, then for every $v \in R_{\hat{l}}$, $d_G(v, u_l) = d_G(v, u_r) = 1$ and for every $v \in R_{\hat{r}}$, $d_G(v, u_l) = d_G(v, u_r) = 2$. In this case u_l and u_r are not resolved by any vertex of $R_{\hat{l}} \cup R_{\hat{r}}$ and thus $R_{\hat{l}} \cup R_{\hat{r}}$ is not a resolving set for $V(G_{\hat{w}})$ in G.

   (b) Let $\hat{l}$ be an inner node of T and $\hat{r}$ be a leaf of T.

   If u_l is not a 1-vertex w.r.t. $R_{\hat{l}}$, then there is a vertex $v_l \in R_{\hat{l}}$ such that $d_G(v_l, u_l) = 2$. Since $d_G(v_l, u_r) = 1$, vertex u_l and vertex u_r are resolved in G by v_l. Thus $R_{\hat{l}}$ is a resolving set for $V(G_{\hat{w}})$ in G. By Lemma 1, it is a minimum resolving set for $V(G_{\hat{w}})$ in G if $R_{\hat{l}}$ is a minimum resolving set for $V(G_{\hat{l}})$ in G.

   If u_l is a 1-vertex w.r.t. $R_{\hat{l}}$, then for each vertex $v_l \in R_{\hat{l}}$, $d_G(v_l, u_l) = 1$. In this case, vertex u_l and vertex u_r are not resolved by any vertex of $R_{\hat{l}}$ and thus $R_{\hat{l}}$ is not a resolving set for $V(G_{\hat{w}})$ in G.

   (c) Let $\hat{l}$ be a leaf of T and $\hat{r}$ be an inner node of T.

   If u_r is not a 2-vertex w.r.t. $R_{\hat{r}}$, then there is a vertex $v_r \in R_{\hat{r}}$ such that $d_G(v_r, u_r) = 1$. Since $d_G(v_r, u_l) = 2$, vertex u_l and vertex u_r are resolved in G by v_r. Thus $R_{\hat{r}}$ is a resolving set for $V(G_{\hat{w}})$ in G. By Lemma 1, it is a minimum resolving set for $V(G_{\hat{w}})$ in G if $R_{\hat{r}}$ is a minimum resolving set for $V(G_{\hat{r}})$ in G.

   If u_r is a 2-vertex w.r.t. $R_{\hat{r}}$, then for each vertex $v_r \in R_{\hat{r}}$, $d_G(v_r, u_r) = 2$. In this case, vertex u_r and vertex u_l are not resolved by any vertex of $R_{\hat{r}}$ and thus $R_{\hat{r}}$ is not a resolving set for $V(G_{\hat{w}})$ in G.

□

The proof of Lemma 2 shows the following observation.

1. Suppose $\hat{l}$ and $\hat{r}$ are inner node of T. If $R_{\hat{l}} \cup R_{\hat{r}}$ is not a resolving set for $V(G_{\hat{w}})$ in G, then it is sufficient to add one additional vertex u of $G_{\hat{w}}$ to $R_{\hat{l}} \cup R_{\hat{r}}$ to get a resolving set for $V(G_{\hat{w}})$ in G. This is possible with a vertex of $G_{\hat{l}}$ and with a vertex from $G_{\hat{r}}$. If $\hat{w}$ is labelled by $\cup$ ($\times$, $\gg$), then we can use the 2-vertex (1-vertex, 1-vertex) of $G_{\hat{l}}$ and the (2-vertex, 1-vertex, 2-vertex) of $G_{\hat{r}}$.

2. Suppose $\hat{l}$ is an inner node and $\hat{r}$ is a leaf of T. If $R_{\hat{l}}$ is not a resolving set for $V(G_{\hat{w}})$ in G, then it is also sufficient to add one additional vertex u of $G_{\hat{w}}$ to $R_{\hat{l}}$ to get a resolving set for $V(G_{\hat{w}})$ in G. If $\hat{w}$ is labelled by $\cup$ ($\times$, $\gg$), then we can use the 2-vertex (1-vertex, 1-vertex) of $G_{\hat{l}}$ or vertex vertex($\hat{r}$).

3. Suppose $\hat{l}$ is a leaf and $\hat{r}$ is an inner node of T. If $R_{\hat{r}}$ is not a resolving set for $V(G_{\hat{w}})$ in G, then again it is also sufficient to add one additional vertex u of $G_{\hat{w}}$ to $R_{\hat{r}}$ to get a resolving set for $V(G_{\hat{w}})$ in G. Since $\hat{w}$ is labelled by $\gg$, we can use vertex vertex($\hat{l}$) and the 2-vertex of $G_{\hat{r}}$.

If a strongly connected graph G has a 1-vertex u_1 (a 2-vertex u_2) w.r.t. a resolving set R, then, obviously, G has no 1-vertex w.r.t. $R \cup \{u_1\}$ (no 2-vertex w.r.t. $R \cup \{u_2\}$, respectively). Suppose G has both a 1-vertex u_1 and a 2-vertex u_2 w.r.t. a resolving set R. If there is an edge from u_1 to u_2 (no edge from u_2 to u_1), then G neither has a 1-vertex nor a 2-vertex w.r.t. $R \cup \{u_1\}$ (w.r.t. $R \cup \{u_2\}$, respectively). In this case, we call vertex u_1 (vertex u_2, respectively) a *double remover*.

Notation 2. *Let u_1 be a 1-vertex and u_2 be a 2-vertex of a strongly connected graph G w.r.t. a vertex set R. A vertex v is a* double remover *if u_1 is not a 1-vertex and u_2 is not a 2-vertex of G w.r.t. $R \cup \{v\}$.*

Figure 4 (1.), (2.), and (3.) show the necessary and forbidden edges for the case that a 1-vertex u_1 is a double remover, a 2-vertex u_2 is a double remover, and a 1-vertex u_1 and a 2-vertex u_2 are both double removers. An induced subgraph with all necessary edges and without all of the forbidden edges of the graph in the third case is not a directed co-graph.

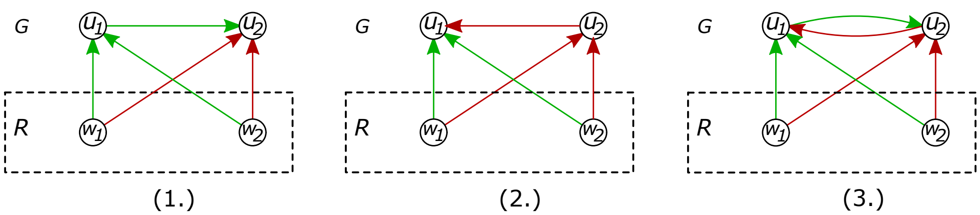


Figure 4: Let R be a resolving set for $V(G)$ in G, vertex u_1 be a 1-vertex w.r.t. R, and vertex u_2 be a 2-vertex w.r.t. R. In graph (1.), vertex u_2 is not a 2-vertex w.r.t. $R \cup \{u_1\}$. In graph (2.), vertex u_1 is not a 1-vertex w.r.t. $R \cup \{u_2\}$. In graph (3.), vertex u_1 is not a 1-vertex w.r.t. $R \cup \{u_2\}$ and vertex u_2 is not a 2-vertex w.r.t. $R \cup \{u_1\}$. If these conditions are met, the green edges must be edges of $E(G)$ and the red edges must not be edges of $E(G)$. There is no directed co-graph that has an induced subgraph with vertex set $\{u_1, u_2, w_1, w_2\}$ and all the necessary green edges but none of the forbidden red edges of graph (3.).

The next lemma shows that there is no vertex $v \in V(G_{\hat{w}}) \setminus \{u_1, u_2\}$ such that $G_{\hat{w}}$ neither has a 1-vertex nor a 2-vertex w.r.t. $R_{\hat{w}} \cup \{v\}$. That is, u_1 and u_2 are the only possible double removers. This shows that we only need to consider a 1-vertex or a 2-vertex to remove a 1-vertex or 2-vertex of $G_{\hat{w}}$ w.r.t. a resolving set $R_{\hat{w}}$, see Figure 5.

Lemma 3. *Let G be a strongly connected directed co-graph with co-tree T, $\hat{w} \in V(T)$ be an inner node of T, and $R_{\hat{w}} \subseteq V(G_{\hat{w}})$ be a resolving sets for $V(G_{\hat{w}})$ in G, such that $G_{\hat{w}}$ has a 1-vertex u_1 and a 2-vertex u_2 w.r.t. $R_{\hat{w}}$. Then there is no vertex $v \in V(G_{\hat{w}}) \setminus \{u_1, u_2\}$ such that $R_{\hat{w}} \cup \{v\}$ is a resolving set for $V(G_{\hat{w}})$ in G and $G_{\hat{w}}$ has neither a 1-vertex nor a 2-vertex w.r.t. $R_{\hat{w}} \cup \{v\}$.*

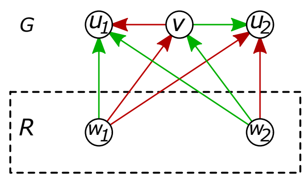


Figure 5: Let R be a resolving set for $V(G)$ in G, vertex u_1 be a 1-vertex w.r.t. R, vertex u_2 be a 2-vertex w.r.t. R, and vertex v be a vertex such that u_1 is not a 1-vertex w.r.t. $R \cup \{v\}$ and u_2 is not a 2-vertex w.r.t. $R \cup \{v\}$. Then the green edges must be edges of $E(G)$ and the red edges must not be edges of $E(G)$. There is no directed co-graph that has an induced subgraph with vertex set $\{u_1, v, u_2, w_1, w_2\}$ and all the necessary green edges but none of the forbidden red edges.

Proof by contradiction. Suppose there is a vertex $v \in V(G_{\hat{w}}) \setminus \{u_1, u_2\}$ such that $R_{\hat{w}} \cup \{v\}$ is a resolving set for $V(G_{\hat{w}})$ in G and $G_{\hat{w}}$ neither has a 1-vertex nor a 2-vertex w.r.t. $R_{\hat{w}} \cup \{v\}$. Since $R_{\hat{w}}$ is a resolving set for $V(G_{\hat{w}})$ in G the three vertex pairs u_1, u_2, u_1, v and u_2, v have to be resolved in G by vertices of $R_{\hat{w}}$. Since each vertex of $R_{\hat{w}}$ resolves the vertex pair u_1, u_2 in G, set $R_{\hat{w}}$ has to contain at least two vertices w_1, w_2, where w_1 has no edge to v and w_2 has an edge to v. That is, vertex w_1 resolves vertex pair u_1, v in G and vertex w_2 resolves vertex pair u_2, v in G. Up to now, we have $(w_1, u_1), (w_2, u_1), (w_2, v) \in E(G_{\hat{w}})$ and $(w_1, u_2), (w_1, v), (w_2, u_2) \notin E(G_{\hat{w}})$. Since u_1 is not a 1-vertex w.r.t. $R_{\hat{w}} \cup \{v\}$, we have $(v, u_1) \notin E(G_{\hat{w}})$. Since u_2 is not a 2-vertex w.r.t. $R_{\hat{w}} \cup \{v\}$, we have $(v, u_2) \in E(G_{\hat{w}})$. Figure 5 illustrates this situation.

Now it is easy to see that there is no partition of the vertex set $\{u_1, u_2, v, w_1, w_2\}$ into two non-empty sets U_1 and U_2 such that there are graphs G_1 and G_2 with vertex sets U_1 and U_2, respectively, and $G_1 \cup G_2$, $G_1 \times G_2$, or $G_1 \gg G_2$ contains the required edges and does not contain the forbidden edges. □

For each inner node $\hat{w}$ of T, the procedure **Merge**$(\hat{w}, \hat{l}, \hat{r})$ computes at most two of the following 4 minimum resolving sets for $V(G_{\hat{w}})$ in G. The 4 sets are denoted by $R_{\hat{w}, t_{\hat{w}}}$, where $t_{\hat{w}} \in \{0, 1, 2, 12\}$ and

1. $R_{\hat{w},0}$ is a minimum resolving set for $V(G_{\hat{w}})$ in G such that $G_{\hat{w}}$ neither has a 1-vertex nor a 2-vertex w.r.t. $R_{\hat{w},0}$,

2. $R_{\hat{w},1}$ is a minimum resolving set for $V(G_{\hat{w}})$ in G such that $G_{\hat{w}}$ has a 1-vertex but no 2-vertex w.r.t. $R_{\hat{w},1}$,

3. $R_{\hat{w},2}$ is a minimum resolving set for $V(G_{\hat{w}})$ in G such that $G_{\hat{w}}$ has no 1-vertex but a 2-vertex w.r.t. $R_{\hat{w},2}$, and

4. $R_{\hat{w},12}$ is a minimum resolving set for $V(G_{\hat{w}})$ in G such that $G_{\hat{w}}$ has a 1-vertex and a 2-vertex w.r.t. $R_{\hat{w},12}$.

Additionally, we store for each defined set $R_{\hat{w},1}$ and $R_{\hat{w},2}$, the 1-vertex and 2-vertex of $G_{\hat{w}}$ w.r.t. $R_{\hat{w},1}$ and $R_{\hat{w},2}$, respectively, as well as for each defined set $R_{\hat{w},12}$, the 1-vertex and 2-vertex of $G_{\hat{w}}$ w.r.t. $R_{\hat{w},12}$ and the information whether one of these two vertices, and if so, which one of them, is a double remover.

The minimum resolving sets for $V(G_{\hat{w}})$ in G are computed from the previously computed minimum resolving sets for $V(G_{\hat{l}})$ and $V(G_{\hat{r}})$ in G. In the simplest case, a minimum resolving set for $V(G_{\hat{w}})$ in G can be defined by the union of a minimum resolving set for $V(G_{\hat{l}})$ in G and a minimum resolving set for $V(G_{\hat{r}})$ in G. In the worst case, at most one additional vertex from $V(G_{\hat{w}})$ has to be added. However, this additional vertex v is always a 1-vertex or 2-vertex of $G_{\hat{l}}$ w.r.t. the corresponding minimum resolving set for $V(G_{\hat{l}})$ in G, or a 1-vertex or 2-vertex of $G_{\hat{r}}$ w.r.t. the corresponding minimum resolving set for $V(G_{\hat{r}})$ in G. Existing double removers are of course preferred here. If $\hat{l}$ or $\hat{r}$ or both are leaves, then vertex vertex($\hat{l}$) and vertex($\hat{r}$) are also a possible choice for v.

To explain how the procedure **Merge**$(\hat{w}, \hat{l}, \hat{r})$ works in detail, we again distinguish between the three labels $\cup$, $\times$ and $\gg$ for node $\hat{w}$ and the cases where the successor nodes of $\hat{w}$ are inner nodes or leaves. Let $R_{\hat{l},t_{\hat{l}}}$ be an already computed resolving set for $V(G_{\hat{l}})$ in G and $R_{\hat{r},t_{\hat{r}}}$ be an already computed resolving set for $V(G_{\hat{r}})$ in G, where $t_{\hat{l}}, t_{\hat{r}} \in \{0, 1, 2, 12\}$. To update a resolving set $R_{\hat{w},t_{\hat{w}}}$ for some $t_{\hat{w}} \in \{0, 1, 2, 12\}$ by a minimum resolving set R means that $R_{\hat{w},t_{\hat{w}}}$ is set to R, if $R_{\hat{w},t_{\hat{w}}}$ is not already defined, or the size of R is less than the size of $R_{\hat{w},t_{\hat{w}}}$, if $R_{\hat{w},t_{\hat{w}}}$ is already defined.

Suppose $R_{\hat{w},0}$, $R_{\hat{w},1}$, $R_{\hat{w},2}$, and $R_{\hat{w},12}$ are defined. The following simplifications for updating the quantities result from the fact that we can prefer smaller quantities or quantities of the same size with fewer restrictions.

- If $|R_{\hat{w},0}| \leq |R_{\hat{w},1}|$ or $|R_{\hat{w},0}| \leq |R_{\hat{w},2}|$, then we do not need to update $R_{\hat{w},1}$ or $R_{\hat{w},2}$.
- If $|R_{\hat{w},1}| < |R_{\hat{w},0}|$ or $|R_{\hat{w},2}| < |R_{\hat{w},0}|$, then we do not need to update $R_{\hat{w},0}$.
- If $|R_{\hat{w},1}| \leq |R_{\hat{w},12}|$ or $|R_{\hat{w},2}| \leq |R_{\hat{w},12}|$, then we do not need to update $R_{\hat{w},12}$.
- If $|R_{\hat{w},1}| < |R_{\hat{w},12}|$ or $|R_{\hat{w},2}| < |R_{\hat{w},12}|$, then we do not need to update $R_{\hat{w},1}$ or $R_{\hat{w},2}$.

That is, we either consider the resolving set $R_{\hat{w},0}$, one or both of the resolving sets $R_{\hat{w},1}$ and $R_{\hat{w},2}$, or the resolving set $R_{\hat{w},12}$ for $V(G_{\hat{w}})$ in G.

Initially, all resolving sets $R_{\hat{w},t_{\hat{w}}}$, $\hat{w} \in V(T)$, $t_{\hat{w}} \in \{0, 1, 2, 12\}$, are undefined. The rules for updating the sets are summarised in Tables 1 to 3. The rows $(t_{\hat{l}}, t_{\hat{r}}, v, t_{\hat{w}}, u_1, u_2)$ of Tables 1 and 2 show how the resolving sets $R_{\hat{w},t_{\hat{w}}}$ for $V(G_{\hat{w}})$ in G are updated.

- The types $t_{\hat{l}}$, $t_{\hat{r}}$, and $t_{\hat{w}}$ are the types of the resolving sets for $V(G_{\hat{l}})$, $V(G_{\hat{r}})$, and $V(G_{\hat{w}})$ in G, respectively. If $t_{\hat{l}} = " - "$ or $t_{\hat{r}} = " - "$, then $\hat{l}$ or $\hat{r}$ is a leaf of T, respectively.
- Vertex v is the vertex that can be added to create a resolving set for $V(G_{\hat{w}})$ in G from the union of a minimum resolving set for $V(G_{\hat{l}})$ in G and a minimum resolving set for $V(G_{\hat{r}})$ in G. If $v = " - "$, then $R_{\hat{l},t_{\hat{l}}} \cup R_{\hat{r},t_{\hat{r}}}$ is already a resolving set for $V(G_{\hat{w}})$ in G. In this case, $R_{\hat{w},t_{\hat{w}}}$ is updated by $R_{\hat{l},t_{\hat{l}}} \cup R_{\hat{r},t_{\hat{r}}}$. Otherwise, $R_{\hat{l},t_{\hat{l}}} \cup R_{\hat{r},t_{\hat{r}}} \cup \{v\}$ is a resolving set for $V(G_{\hat{w}})$ in G and $R_{\hat{w},t_{\hat{w}}}$ is updated by $R_{\hat{l},t_{\hat{l}}} \cup R_{\hat{r},t_{\hat{r}}} \cup \{v\}$. Vertex v is either the 1-vertex, 2-vertex, or double remover of $V(G_{\hat{l}})$ w.r.t. $R_{\hat{l},t_{\hat{l}}}$ denoted by $u_{l,1}$, $u_{l,2}$, and $u_{l,\star}$, respectively, or the 1-vertex, 2-vertex, or double remover of $V(G_{\hat{r}})$ w.r.t. $R_{\hat{r},t_{\hat{r}}}$ denoted by $u_{r,1}$, $u_{r,2}$, and $u_{r,\star}$, respectively. If $\hat{l}$ or $\hat{r}$ is a leaf, then $u_l = \text{vertex}(\hat{l})$ and $u_r = \text{vertex}(\hat{r})$, respectively.
- Vertex u_1 and u_2 are the 1-vertex and 2-vertex of $G_{\hat{w}}$ w.r.t. the resulting resolving set for $V(G_{\hat{w}})$ in G. A created double remover gets a superscript asterisk. If $u_1 = " - "$ or $u_2 = " - "$, then $G_{\hat{w}}$ has no 1-vertex or 2-vertex, respectively, w.r.t. the resulting resolving set for $V(G_{\hat{w}})$ in G.

Table 3 describes the minimum resolving sets for $V(G_{\hat{w}})$ in G for the case that $\hat{l}$ and $\hat{r}$ are leaves in T.

$t_{\hat{l}}$	$t_{\hat{r}}$	v	$t_{\hat{w}}$	u_1	u_2
0	0	–	0	–	–
0	1	–	0	–	–
0	2	–	2	–	$u_{r,2}$
0	12	–	2	–	$u_{r,2}$
1	0	–	0	–	–
1	1	–	0	–	–
1	2	–	2	–	$u_{r,2}$
1	12	–	2	–	$u_{r,2}$
2	0	–	2	–	$u_{l,2}$
2	1	–	2	–	$u_{l,2}$
2	2	$u_{l,2}$	2	–	$u_{r,2}$
		$u_{r,2}$	2	–	$u_{l,2}$
2	12	$u_{l,2}$	2	–	$u_{r,2}$
		$u_{r,2}$	2	–	$u_{l,2}$
		$u_{r,\star}$	2	–	$u_{l,2}$
12	0	–	2	–	$u_{l,2}$
12	1	–	2	–	$u_{l,2}$
12	2	$u_{l,2}$	2	–	$u_{r,2}$
		$u_{l,\star}$	2	–	$u_{r,2}$
		$u_{r,2}$	2	–	$u_{l,2}$
12	12	$u_{l,2}$	2	–	$u_{r,2}$
		$u_{l,\star}$	2	–	$u_{r,2}$
		$u_{r,2}$	2	–	$u_{l,2}$
		$u_{r,\star}$	2	–	$u_{l,2}$

Label $\cup$

$t_{\hat{l}}$	$t_{\hat{r}}$	v	$t_{\hat{w}}$	u_1	u_2
0	0	–	0	–	–
0	1	–	1	$u_{r,1}$	–
0	2	–	0	–	–
0	12	–	1	$u_{r,1}$	–
1	0	–	1	$u_{l,1}$	–
1	1	$u_{l,1}$	1	$u_{r,1}$	–
		$u_{r,1}$	1	$u_{l,1}$	–
1	2	–	1	$u_{l,1}$	–
1	12	$u_{l,1}$	1	$u_{r,1}$	–
		$u_{r,1}$	1	$u_{l,1}$	–
		$u_{r,\star}$	1	$u_{l,1}$	–
2	0	–	0	–	–
2	1	–	1	$u_{r,1}$	–
2	2	–	0	–	–
2	12	–	1	$u_{r,1}$	–
12	0	–	1	$u_{l,1}$	–
12	1	$u_{l,1}$	1	$u_{r,1}$	–
		$u_{l,\star}$	1	$u_{r,1}$	–
		$u_{r,1}$	1	$u_{l,1}$	–
12	2	–	1	$u_{l,1}$	–
12	12	$u_{l,1}$	1	$u_{r,1}$	–
		$u_{l,\star}$	1	$u_{r,1}$	–
		$u_{r,1}$	1	$u_{l,1}$	–
		$u_{r,\star}$	1	$u_{l,1}$	–

Label $\times$

$t_{\hat{l}}$	$t_{\hat{r}}$	v	$t_{\hat{w}}$	u_1	u_2
0	0	–	0	–	–
0	1	–	1	$u_{r,1}$	–
0	2	–	0	–	–
0	12	–	1	$u_{r,1}$	–
1	0	–	0	–	–
1	1	–	1	$u_{r,1}$	–
1	2	$u_{l,1}$	0	–	–
		$u_{r,2}$	0	–	–
1	12	$u_{l,1}$	1	$u_{r,1}$	–
		$u_{r,2}$	1	$u_{r,1}$	–
		$u_{r,\star}$	0	–	–
2	0	–	2	–	$u_{l,2}$
2	1	–	12	$u_{r,1}$	$u_{l,2}$
2	2	–	2	–	$u_{l,2}$
2	12	–	12	$u_{r,1}$	$u_{l,2}$
12	0	–	2	–	$u_{l,2}$
12	1	–	12	$u_{r,1}$	$u_{l,2}$
12	2	$u_{l,1}$	2	–	$u_{l,2}$
		$u_{l,\star}$	0	–	–
		$u_{r,2}$	2	–	$u_{l,2}$
12	12	$u_{l,1}$	12	$u_{r,1}$	$u_{l,2}$
		$u_{l,\star}$	1	$u_{r,1}$	–
		$u_{r,2}$	12	$u_{r,1}$	$u_{l,2}$
		$u_{r,\star}$	2	–	$u_{l,2}$

Label $\gg$

Table 1: The tables for the case that $\hat{l}$ and $\hat{r}$ are inner nodes of T.

The rows with a blue background do not need to be taken into account, because in these cases there is another selection that provides a solution that is at least as good as the blue one.

The running time of computing a minimum resolving set for G is therefore linear in the size of the given co-graph G.

Theorem 1. *A minimum resolving set for a strongly connected directed co-graph G is computable in linear time.*

Example 1. *The table below shows the results computed by the* **BottomUp**() *procedure for the co-graph G defined by the co-tree T below. There is a column for each possible minimum resolving set $R_{\hat{w},t_{\hat{w}}}$, $t_{\hat{w}} \in \{0,1,2,12\}$. The vertices u_1, u_2 next to the minimum resolving sets $R_{\hat{w},t_{\hat{w}}}$ are the 1-vertex and 2-vertices of $G_{\hat{w}}$ w.r.t. $R_{\hat{w},t_{\hat{w}}}$. The double remover is marked by an asterisk.*

$t_{\hat{l}}$	$t_{\hat{r}}$	v	$t_{\hat{w}}$	u_1	u_2
0	–	–	2	–	u_r
1	–	–	12	$u_{l,1}$	$u_r^\star$
2	–	$u_{l,2}$	2	–	u_r
		u_r	2	–	$u_{l,2}$
12	–	$u_{l,2}$	12	$u_{l,1}$	$u_r^\star$
		$u_{l,\star}$	2	–	u_r
		u_r	2	–	$u_{l,2}$

Label $\cup$, $\hat{r}$ is a leaf

$t_{\hat{l}}$	$t_{\hat{r}}$	v	$t_{\hat{w}}$	u_1	u_2
0	–	–	1	u_r	–
1	–	$u_{l,1}$	1	u_r	–
		u_r	1	$u_{l,1}$	–
2	–	–	12	$u_r^\star$	$u_{l,2}$
12	–	$u_{l,1}$	12	$u_r^\star$	$u_{l,2}$
		$u_{l,\star}$	1	u_r	–
		u_r	1	$u_{l,1}$	–

Label $\times$, $\hat{r}$ is a leaf

$t_{\hat{l}}$	$t_{\hat{r}}$	v	$t_{\hat{w}}$	u_1	u_2
0	–	–	1	u_r	–
1	–	$u_{l,1}$	1	u_r	–
		u_r	0	–	–
2	–	–	12	u_r	$u_{l,2}$
12	–	$u_{l,1}$	12	u_r	$u_{l,2}$
		$u_{l,\star}$	1	u_r	–
		u_r	2	–	$u_{l,2}$

Label $\gg$, $\hat{r}$ is a leaf

$t_{\hat{l}}$	$t_{\hat{r}}$	v	$t_{\hat{w}}$	u_1	u_2
–	0	–	2	–	u_l
–	1	–	12	$u_{r,1}$	u_l
–	2	u_l	0	–	–
		$u_{r,2}$	2	–	u_l
–	12	u_l	1	$u_{r,1}$	–
		$u_{r,2}$	12	$u_{r,1}$	u_l
		$u_{r,\star}$	2	–	u_l

Label $\gg$, $\hat{l}$ is a leaf

Table 2: The tables for the case that either $\hat{l}$ or $\hat{r}$ is a leaf of T.

$R_{\hat{w},t_{\hat{w}}}$	$t_{\hat{w}}$	u_1	u_2
$\{u_l\}$	2	–	u_r
$\{u_r\}$	2	–	u_l

Label $\cup$, $\hat{l}$ and $\hat{r}$ are leaves

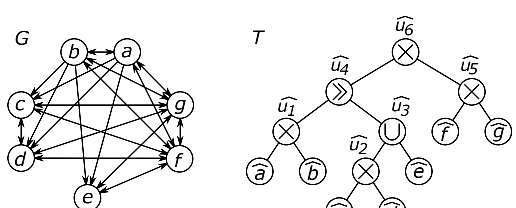
$R_{\hat{w},t_{\hat{w}}}$	$t_{\hat{w}}$	u_1	u_2
$\{u_l\}$	1	u_r	–
$\{u_r\}$	1	u_l	–

Label $\times$, $\hat{l}$ and $\hat{r}$ are leaves

$R_{\hat{w},t_{\hat{w}}}$	$t_{\hat{w}}$	u_1	u_2
$\{u_l\}$	1	u_r	–
$\{u_r\}$	2	–	u_l

Label $\gg$, $\hat{l}$ and $\hat{r}$ are leaves

Table 3: The tables for the case that $\hat{l}$ and $\hat{r}$ are leaves of T.



$G_{\hat{w}}$	$R_{\hat{w},0}$	$R_{\hat{w},1}/u_1$	$R_{\hat{w},2}/u_2$	$R_{\hat{w},12}/u_1,u_2$
$G_{\hat{u}_1} = G_{\hat{a}} \times G_{\hat{b}}$	–	$\{a\}/b$	–	–
$G_{\hat{u}_2} = G_{\hat{c}} \times G_{\hat{d}}$	–	$\{c\}/d$	–	–
$G_{\hat{u}_3} = G_{\hat{u}_2} \cup \hat{e}$	–	–	–	$\{c\}/d, e^\star$
$G_{\hat{u}_4} = G_{\hat{u}_1} \gg G_{\hat{u}_3}$	$\{a, c, e\}$	–	–	–
$G_{\hat{u}_5} = G_{\hat{f}} \times G_{\hat{g}}$	–	–	$\{f\}/g$	–
$G_{\hat{u}_6} = G_{\hat{u}_4} \times G_{\hat{u}_5}$	$\{a, c, e, f\}$	–	–	–

Vertex set $\{a, c, e, f\}$ *is a minimum resolving set for the strongly connected directed co-graph* G *defined by co-tree* T *of the figure above.*

3 Directed acyclic graphs

An undirected graph can easily be transformed into a directed graph by replacing each undirected edge $\{u, v\}$ by two directed edges (u, v) and (v, u). Thus, Directed Metric Dimension is NP-complete, because Metric Dimension is NP-complete, see [KRR96], but it is also NP-complete for oriented graphs, see [RRCM14]. However, the following theorem shows that Directed Metric Dimension is also NP-complete for directed acyclic graphs, i.e., for DAGs.

Theorem 2. Directed Metric Dimension *is NP-complete for DAGs.*

Proof. The problem is obviously in NP. The NP-hardness is shown by a polynomial time reduction from Hitting Set. Let $\mathcal{C} = \{C_1, \dots, C_m\}$ be a set of subsets of a set $X = \{x_1, \dots, x_n\}$. We define a directed acyclic graph G such that G has a resolving set of size at most $3 + k$ if and only if there is a subset $X' \subset X$ of size at most k such that $X' \cap C_j \neq \emptyset$ for $j = 1, \dots, m$. W.l.o.g., we assume that $m > n$, otherwise duplicate some sets of $\mathcal{C}$, and that for each subset C_j there is at least one x_i such that $x_i \notin C_j$.

The graph G defined for $X, \mathcal{C}$ has

1. three vertices u_a, u_b and u_c,
2. n vertices u_{x_i} for $i = 1, \dots, n$,
3. $2m$ vertices $u_{C_j}, u_{C'_j}$ for $j = 1, \dots, m$,

and

1. n edges (u_a, u_{x_i}) for $i = 1, \dots, n$,
2. one edge (u_b, u_{x_1}),
3. $n - 1$ edges $(u_{x_i}, u_{x_{i+1}})$ for $i = 1, \dots, n - 1$,
4. two edges $(u_c, u_{C_1}), (u_c, u_{C'_1})$,
5. m edges $(u_{C_j}, u_{C'_j})$, for $j = 1, \dots, m$,
6. $4(m-1)$ edges $(u_{C_j}, u_{C_{j+1}}), (u_{C_j}, u_{C'_{j+1}}), (u_{C'_j}, u_{C_{j+1}}), (u_{C'_j}, u_{C'_{j+1}})$, for $j = 1, \dots, m - 1$,
7. nm edges (u_{x_i}, u_{C_j}) for $i = 1, \dots, n$ for $j = 1, \dots, m$, and
8. an edge $(u_{x_i}, u_{C'_j})$, $1 \le i \le n$, $1 \le j \le m$, if and only if $x_i \notin C_j$.

Figure 6 shows an example.

Each minimum resolving set for G contains the three vertices u_a, u_b, u_c, because these vertices have no incoming edges. Vertex u_b resolves every vertex pair u_{x_i}, u_{x_j}, $1 \le i < j \le n$, because $d_G(u_b, u_{x_i}) = i$. Vertex u_c resolves every vertex pair (u_{C_i}, u_{C_j}), $(u_{C_i}, u_{C'_j})$, $(u_{C'_i}, u_{C_j})$ and $(u_{C'_i}, u_{C'_j})$, $1 \le i < j \le m$, because $d_G(u_c, u_{C_i}) = d_G(u_c, u_{C'_i}) = i$. Vertex u_a resolves every vertex pair u_{x_i}, u_{C_j} and every vertex pair $u_{x_i}, u_{C'_j}$, $1 \le i \le n$, $1 \le j \le m$, because $d_G(u_a, u_{x_i}) = 1$ and $d_G(u_a, u_{C_j}) = 2$. Note that we assume that for each subset C_j there is at least one x_i such that $x_i \notin C_j$, therefore $d_G(u_a, u_{C'_j}) = 2$. The only vertex pairs that have not

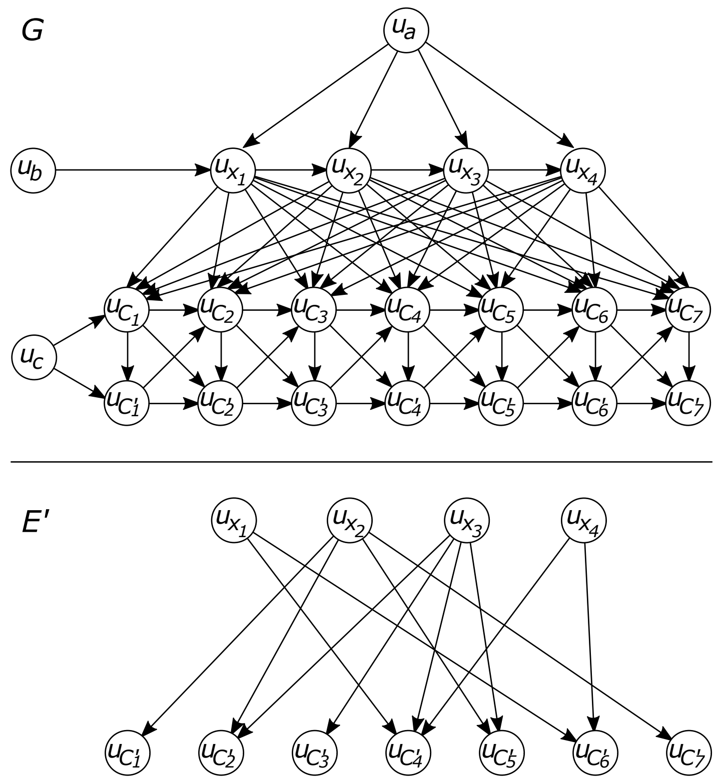


Figure 6: Graph G obtained by instance $X, \mathcal{C}$ with $X = \{x_1, x_2, x_3, x_4\}$, $\mathcal{C} = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7\}$, $C_1 = \{x_1, x_3, x_4\}$, $C_2 = \{x_1, x_4\}$, $C_3 = \{x_1, x_2, x_4\}$, $C_4 = \{x_2\}$, $C_5 = \{x_1, x_4\}$, $C_6 = \{x_2, x_3\}$, $C_7 = \{x_1, x_3, x_4\}$ for Hitting Set in the proof of Theorem 2. The edges which result from the membership of the elements x_i to the subset C_j are drawn separately below for reasons of clarity. These are the edges in edge set E'. Set $X' = \{x_2, x_4\}$ is a minimum Hitting set for $X, \mathcal{C}$, where $\{u_a, u_b, u_c, u_{x_2}, u_{x_4}\}$ is a minimum resolving set for G.

been considered yet are the vertex pairs $u_{C_j}, u_{C'_j}$ for $j = 1, \ldots, m$. A vertex pair $u_{C_j}, u_{C'_j}$ can only be resolved by vertex u_{C_j}, by vertex $u_{C'_j}$ or by a vertex u_{x_i} such that $x_i \in C_j$, because $x_i \in C_j$ if and only if $d_G(u_{x_i}, u_{C'_j}) = 2$.

Let $X' \subseteq X$ be a hitting set for $\mathcal{C}$ of size at most k, i.e., $X' \cap C_j \neq \emptyset$ for $j = 1, \ldots, m$. Then the vertices u_{x_i} with $x_i \in X'$ resolve all vertex pairs u_{C_j}, $u_{C'_j}$, $1 \leq j \leq m$. Thus

$$R = \{u_a, u_b, u_c\} \cup \bigcup_{x_i \in X'} u_{x_i}$$

is a resolving set for G of size at most $3 + k$. Let R be a resolving set for G of size at most $3 + k$. If R contains a vertex $u_{C_j} \in R$ or $u_{C'_j} \in R$, then replace it by a vertex u_{x_i} where $x_i \in C_j$. The resulting set R' is also a resolving set for G of size at most $3 + k$ and

$$X' = \{x_i | u_{x_i} \in R'\}$$

is a hitting set for $\mathcal{C}$. □

4 Conclusions

In this paper we have shown that Directed Metric Dimension is decidable in linear time for directed co-graphs and we also presented an algorithm to compute minimum resolving sets for directed co-graphs in linear time. Additionally, we have shown that Directed Metric Dimension is NP-complete for DAGs, extending the existing results for general directed and oriented graphs.

The metric dimension as well as its variants have rarely been studied for directed graphs [SW21]. Developing efficient algorithms to compute the metric dimension for specific graph classes is one of the most interesting challenges to us.

References

[BBS+11] Bača, Martin ; Baskoro, Edy T. ; Salman, A. N. M. ; Saputro, Suhadi W. ; Suprijanto, Djoko: The Metric Dimension of Regular Bipartite Graphs. In: *Bulletin mathématiques de la Société des sciences mathématiques de Roumanie* 54 (2011), Nr. 1, S. 15–28

[BDGR97] Bechet, Denis ; De Groote, Philippe ; Retoré, Christian: A complete axiomatisation for the inclusion of series-parallel partial orders. In: *International Conference on Rewriting Techniques and Applications* Springer, 1997, S. 230–240

[BEE+05] Beerliova, Zuzana ; Eberhard, Felix ; Erlebach, Thomas ; Hall, Alexander ; Hoffmann, Michael ; Mihaľák, Matúš ; Ram, L. S.: Network Discovery and Verification. In: Kratsch, Dieter (Hrsg.): *Graph-Theoretic Concepts in Computer Science*, Springer Berlin Heidelberg, 2005, 127–138

[CEJO00] Chartrand, Gary ; Eroh, Linda ; Johnson, Mark A. ; Oellermann, Ortrud: Resolvability in graphs and the metric dimension of a graph. In: *Discrete Applied Mathematics* 105 (2000), Nr. 1-3, 99–113. http://dx.doi.org/10.1016/S0166-218X(00)00198-0. – DOI 10.1016/S0166–218X(00)00198–0

[CGH08] Chappell, Glenn G. ; Gimbel, John G. ; Hartman, Chris: Bounds on the metric and partition dimensions of a graph. In: *Ars Combinatoria* 88 (2008)

[CP06] Crespelle, Christophe ; Paul, Christophe: Fully dynamic recognition algorithm and certificate for directed cographs. In: *Discrete Applied Mathematics* 154 (2006), Nr. 12, S. 1722–1741

[CPZ00] Chartrand, Gary ; Poisson, Christopher ; Zhang, Ping: Resolvability and the upper dimension of graphs. In: *Computers and Mathematics with Applications* 39 (2000), Nr. 12, S. 19–28

[CRZK00] Chartrand, Gary ; Raines, Michael ; Zhang, Ping ; Kalamazoo: The directed distance dimension of oriented graphs. In: *Mathematica Bohemica* 125 (2000), Nr. 02, 155–168. https://www.emis.de/journals/MB/125.2/mb125_2_5.pdf

[DPSL12] Díaz, Josep ; Pottonen, Olli ; Serna, Maria J. ; Leeuwen, Erik J.: On the Complexity of Metric Dimension. In: *Algorithms - ESA 2012 - 20th Annual European Symposium, Ljubljana, Slovenia, September 10-12, 2012. Proceedings*, 2012, 419–430

[FGO06] Fehr, Melodie ; Gosselin, Shonda ; Oellermann, Ortrud R.: The metric dimension of Cayley digraphs. In: *Discrete Mathematics* 306 (2006), Nr. 1, 31–41. http://dx.doi.org/10.1016/j.disc.2005.09.015. – DOI 10.1016/j.disc.2005.09.015

[FR18] Fernau, Henning ; Rodríguez-Velázquez, Juan A.: On the (adjacency) metric dimension of corona and strong product graphs and their local variants: Combinatorial and computational results. In: *Discrete Applied Mathematics* 236 (2018), 183–202. http://dx.doi.org/10.1016/j.dam.2017.11.019. – DOI 10.1016/j.dam.2017.11.019

[FXW13] Feng, Min ; Xu, Min ; Wang, Kaishun: On the metric dimension of line graphs. In: *Discret. Appl. Math.* 161 (2013), Nr. 6, 802–805. http://dx.doi.org/10.1016/j.dam.2012.10.018. – DOI 10.1016/j.dam.2012.10.018

[GJ79] Garey, Michael R. ; Johnson, David S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979. – ISBN 0–7167–1044–7

[Hay77] Hayat, Sakander: Computing distance-based topological descriptors of complex chemical networks: New theoretical techniques. In: *Chemical Physics Letters* 688 (1977), Nr. 1, 51–58. http://dx.doi.org/10.1016/j.cplett.2017.09.055. – DOI 10.1016/j.cplett.2017.09.055

[HM76] Harary, Frank ; Melter, Robert A.: On the metric dimension of a graph. In: *Ars Combinatoria* 2 (1976), S. 191–195

[HMP+05] Hernando, M. C. ; Mora, Mercè ; Pelayo, Ignacio M. ; Seara, Carlos ; Cáceres, José ; Puertas, María Luz: On the metric dimension of some families of graphs. In: *Electronic Notes in Discrete Mathematics* 22 (2005), 129–133. http://dx.doi.org/10.1016/j.endm.2005.06.023. – DOI 10.1016/j.endm.2005.06.023

[HSV12] Hauptmann, Mathias ; Schmied, Richard ; Viehmann, Claus: Approximation complexity of Metric Dimension problem. In: *Journal of Discrete Algorithms* 14 (2012), 214–222. http://dx.doi.org/10.1016/j.jda.2011.12.010. – DOI 10.1016/j.jda.2011.12.010

[HW12] Hoffmann, Stefan ; Wanke, Egon: Metric Dimension for Gabriel Unit Disk Graphs Is NP-Complete. In: *Algorithms for Sensor Systems, 8th International Symposium on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile Entities, ALGOSENSORS 2012, Ljubljana, Slovenia, September 13-14, 2012. Revised Selected Papers*, 2012, 90–92

[IBSS10] Iswadi, H. ; Baskoro, Edy T. ; Salman, A.N.M. ; Simanjuntak, Rinovia: The metric dimension of amalgamation of cycles. In: *Far East Journal of Mathematical Sciences (FJMS)* 41 (2010), Nr. 1, S. 19–31

[KRR96] Khuller, Samir ; Raghavachari, Balaji ; Rosenfeld, Azriel: Landmarks in Graphs. In: *Discrete Applied Mathematics* 70 (1996), Nr. 3, 217–229. http://dx.doi.org/10.1016/0166-218X(95)00106-2. – DOI 10.1016/0166–218X(95)00106–2

[LA06] Liu, Ke ; Abu-Ghazaleh, Nael B.: Virtual Coordinates with Backtracking for Void Traversal in Geographic Routing. In: *Ad-Hoc, Mobile, and Wireless Networks, 5th International Conference, ADHOC-NOW 2006, Ottawa, Canada, August 17-19, 2006, Proceedings*, 2006, 46–59

[Loz13] Lozano, Antoni: Symmetry Breaking in Tournaments. In: *Electron. J. Comb.* 20 (2013), Nr. 1, 69. http://dx.doi.org/10.37236/3182. – DOI 10.37236/3182

[MT84] Melter, Robert A. ; Tomescu, Ioan: Metric bases in digital geometry. In: *Computer Vision, Graphics, and Image Processing* 25 (1984), Nr. 1, 113–121. http://dx.doi.org/10.1016/0734-189X(84)90051-3. – DOI 10.1016/0734–189X(84)90051–3

[OP07] Oellermann, Ortrud R. ; Peters-Fransen, Joel: The strong metric dimension of graphs and digraphs. In: *Discrete Applied Mathematics* 155 (2007), Nr. 3, 356–364. http://dx.doi.org/10.1016/j.dam.2006.06.009. – DOI 10.1016/j.dam.2006.06.009

[RRCM14] Rajan, Bharati ; Rajasingh, Indra ; Cynthia, Jude A. ; Manuel, Paul D.: Metric dimension of directed graphs. In: *International Journal of Computer Mathematics* 91 (2014), Nr. 7, 1397–1406. http://dx.doi.org/10.1080/00207160.2013.844335. – DOI 10.1080/00207160.2013.844335

[Sla75] Slater, Peter J.: Leaves of trees. In: *Congressum Numerantium* 14 (1975), S. 549–559

[ST04] Sebö, András ; Tannier, Eric: On Metric Generators of Graphs. In: *Mathematics of Operations Research* 29 (2004), Nr. 2, 383–393. http://dx.doi.org/10.1287/moor.1030.0070. – DOI 10.1287/moor.1030.0070

[SVW21] Schmitz, Yannick ; Vietz, Duygu ; Wanke, Egon: A note on the complexity of k-Metric

Dimension. In: *CoRR* abs/2101.12018 (2021). https://arxiv.org/abs/2101.12018

[SW21] Schmitz, Yannick ; Wanke, Egon: On the Strong Metric Dimension of directed co-graphs. In: *CoRR* abs/2111.13054 (2021). https://arxiv.org/abs/2111.13054

9.3 On the strong metric dimension of composed graphs

On the strong metric dimension of composed graphs
Marcel Wagner, Yannick Schmitz, Egon Wanke
Eingereicht in: Theory of Computing Systems

Diese Arbeit enthält einen Algorithmus zur Berechnung der Starken Metrischen Dimension, basierend auf einer Zerlegung eines Graphen in seine zweifachen Zusammenhangskomponenten. Die Ergebnisse sind in Kapitel 6 zu finden. Alle Autoren haben gleichermaßen zur Erarbeitung und zur Formulierung der Ergebnisse beigetragen. Den Hauptteil meines Beitrags bilden die Algorithmen zu den Co-Graphen.

On the Strong Metric Dimension of composed graphs

Marcel Wagner[a], Yannick Schmitz[a,*], Egon Wanke[a]

[a]*Heinrich-Heine-Universität Düsseldorf, Universitätsstraße 1, Düsseldorf, 40225, Germany*

Abstract

Two vertices u and v of an undirected graph G are strongly resolved by a vertex w if there is a shortest path between w and u containing v or a shortest path between w and v containing u. A vertex set R is a strong resolving set for G if for each pair of vertices there is a vertex in R that strongly resolves them. The strong metric dimension of G is the size of a minimum strong resolving set for G. We show that a minimum strong resolving set for an undirected graph G can be computed efficiently if and only if a minimum strong resolving set for each biconnected component of G can be computed efficiently.

Keywords: strong metric dimension, composed graphs, linear time, metric dimension

1. Introduction

In this paper we consider the *strong metric dimension* introduced by Sebö and Tannier in [1]. The strong metric dimension is a variant of the original *metric dimension* (which we simply call metric dimension) which is the smallest number k of vertices so that the vector of distances to every vertex in the graph is unique. Here the distance between two vertices is the number of edges on a shortest path. The k-dimensional distance vectors of the vertices can be viewed as their positions in a k-dimensional space whose structure is defined by the graph.

The metric dimension has been introduced by Slater in [2] and [3] and independently by Harary and Melter in [4]. There are numerous research reports on the analysis of the metric dimension of graphs. Determining whether the metric dimension of a given graph is less than a given integer has been shown to be NP-complete by a reduction from 3-SAT [5] and 3-Dimensional Matching [6]. It is NP-complete for general graphs, planar graphs [7], even for those with maximum degree 6, and Gabriel unit disk graphs [8]. There are several algorithms for computing the metric dimension in polynomial time for special classes of graphs, as for example for trees [9, 5], wheels [10], grid graphs [11], k-regular bipartite graphs [12], amalgamation of cycles [13], outerplanar graphs [7], cactus block graphs [14], chain graphs [15], and graphs with bounded extended biconnected components [16].

The strong metric dimension of a graph G, in contrast to the metric dimension, is the size of a smallest set R of vertices with the following property. For each pair of two

*Corresponding author. E-mail: yannick.schmitz@hhu.de

Declarations of interest: none

distinct vertices u and v in G, there is a vertex $w \in R$ such that there is a shortest path between w and u that contains v or a shortest path between w and v that contains u. Such a set R is called a *strong resolving set* for G. Since in both cases the distance between w and u is different from the distance between w and v, a strong resolving set for G is always a resolving set for G and thus the strong metric dimension is always greater than or equal to the metric dimension. However, if we again calculate the distance vectors $\vec{u} = (d_G(u, w_1), \ldots, d_G(u, w_k))$ for vertices u to the k vertices $w_1, \ldots, w_k$ of a strong resolving set R, then there are significant advantages in contrast to the metric dimension when navigating through the graph. The distance between two vertices u and v is the maximum difference between $d_G(u, w_i)$ and $d_G(v, w_i)$ for $i = 1, \ldots, k$, that is,

$$d_G(u, v) \;=\; \max_{i=1}^{k} |d_G(u, w_i) - d_G(v, w_i)|.$$

To navigate from a vertex u to a vertex v in graph G, we can now simply determine a neighbour u' of u on a shortest path between u and v. This is a neighbour u' of u with $d_G(u', v) = d_G(u, v) - 1$.

Determining whether the strong metric dimension of a given graph is less than a given integer k is NP-complete [17], like it is the case for the metric dimension. Computing the strong metric dimension also has been extensively studied for different graph classes, see for example [18], [19], [20], [21], [22] and [23].

In this paper we show that an efficient computation of the strong metric dimension for a graph G can be reduced to an efficient computation of the biconnected components of G. That is, we consider a composition mechanism that connects two graphs G_1 and G_2 by identifying a vertex from G_1 with a vertex from G_2 in the disjoint union of G_1 and G_2. With this composition mechanism, a graph can be assembled from its biconnected components. Computing the strong metric dimension for graphs obtained by join operations, like the Cartesian product, the strong product and the corona product, has also been studied by other authors, see for example [24] and [25]. We demonstrate the power of our approach by three examples. We show that the strong metric dimension for graphs in which the biconnected components are grids, cycles or co-graphs can be computed in linear time.

2. Strong metric dimension

We consider undirected, connected, and finite graphs $G = (V, E)$, where V is the set of vertices and $E \subseteq \{\{u, v\} \mid u, v \in V, u \neq v\}$ is the set of edges. Two distinct vertices $u, v \in V$ of G are *strongly resolved* by a vertex $w \in V$ if there is a shortest path between w and u that contains v or a shortest path between w and v that contains u. The length of a path is the number of edges. A vertex set $R \subseteq V$ is a *strong resolving set* for G if for each pair of vertices $u, v \in V \setminus R, u \neq v$ there is a vertex $w \in R$ such that u and v are strongly resolved by w. The *strong metric dimension* of graph G is the size of a smallest strong resolving set for G.

Oellermann and Peters-Fransen showed in [17] that finding a strong resolving set of G is equivalent to finding a vertex cover of the so-called *strong resolving graph* G_{SR} of G, defined as follows. For a vertex $u \in V$ let $N_G(u) = \{v \mid \{u, v\} \in E\}$ and $N_G[u] = N_G(u) \cup \{u\}$ be the open and closed neighbourhoods of u, respectively. For two vertices $u, v \in V$ let $d_G(u, v)$ be the distance between u and v in G, that is, the number of edges of a shortest path between u and v. We say a vertex $u \in V$ is *maximally distant* from a vertex w if there is no vertex $v \in N_G(u)$ in the neighbourhood of u with $d_G(v, w) > d_G(u, w)$.

The vertices of the strong resolving graph G_{SR} are the vertices of G. There is an edge between two vertices u and v in G_{SR} if and only if u is maximally distant from v and v is maximally distant from u. In this case we also say that u and v are *mutually maximally distant*. It is easy to see that each strong resolving set for G must contain at least one of two vertices that are mutually maximally distant. Also each set of vertices that contains at least one of two vertices that are mutually maximally distant is a strong resolving set for G. It follows that a strong resolving set for G is a vertex cover for G_{SR} and vice versa. See Figure 1 for an example.

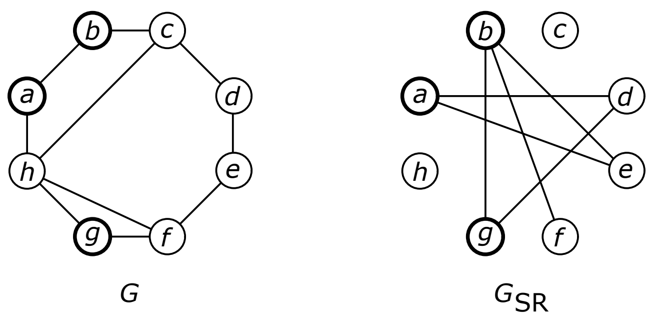


Figure 1: A Graph G and its strong resolving graph G_{SR}. The vertex set $\{a,b,g\}$ is a minimum vertex cover for G_{SR} and a minimum strong resolving set for G.

3. Composing graphs

The graphs we consider arise from attaching child graphs $G_1, \ldots, G_k$ to a parent graph H. This attachment is performed by merging vertices $u_1, \ldots, u_k$ of $G_1, \ldots, G_k$ with vertices $v_1, \ldots, v_k$ from H in the disjoint union of the graphs $G_1, \ldots, G_k$, and H.

Definition 1. *Let $G_i = (V_i, E_i)$, $1 \le i \le k$, and $H = (V_H, E_H)$ be $k+1$ graphs, $u_i \in V_i$ for $i = 1, \ldots, k$, and $v_1, \ldots, v_k \in V_H$. Let $G_{1,\ldots,k}$ be the disjoint union of $G_1, \ldots, G_k$. That is, $G_{1,\ldots,k}$ has vertex set $\cup_{i=1}^k V_i$ and edge set $\cup_{i=1}^k E_i$.*

Then graph

$$G_{1,\ldots,k} \circ_{(u_1,\ldots,u_k \to v_1,\ldots,v_k)} H$$

is defined by vertex set

$$(V_1 \cup \cdots \cup V_k \cup V_H) \setminus \{u_1, \ldots, u_k\}$$

and edge set

$$\begin{array}{ll} (& E_1 \cup \cdots \cup E_k \cup E_H \\ & \cup \{\{w, v_i\} \mid w \in N_{G_i}(u_i), 1 \le i \le k\} \quad) \\ \setminus & \{\{w, u_i\} \mid w \in N_{G_i}(u_i), 1 \le i \le k\}. \end{array}$$

The graph $G_{1,\dots,k} \circ_{(u_1,\dots,u_k \to v_1,\dots,v_k)} H$ is formed by the disjoint union of the k graphs $G_1, \dots, G_k$ and graph H without the vertices $u_1, \dots, u_k$ and their incident edges, in which for $i = 1, \dots, k$ the neighbours of vertex u_i in G_i are connected to vertex v_i. Figure 2 shows an example of the $\circ_{(u_1,\dots,u_k \to v_1,\dots,v_k)}$ operation. For all further discussions, we only consider the case in that all graphs $G_1, \dots, G_k, H$ are vertex disjoint, connected, and have at least two vertices. The vertices $v_1, \dots, v_k$ of H do not need to be distinct.

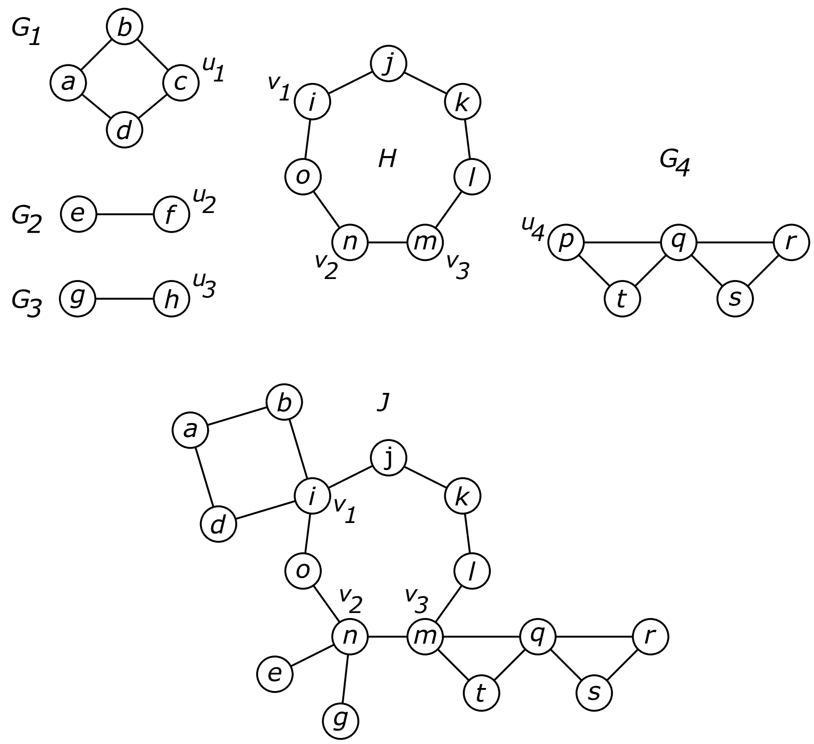


Figure 2: Five graphs G_1, G_2, G_3, G_4, H, and the graph $J = G_{1,2,3,4} \circ_{(u_1,u_2,u_3,u_4 \to v_1,v_2,v_2,v_3)} H$ created by the composition as defined in Definition 1.

4. The strong resolving graph

To compute a strong resolving set for a composed graph $J = G_{1,\dots,k} \circ_{(u_1,\dots,u_k \to v_1,\dots,v_k)} H$, we first determine the edge set of the strong resolving graph J_{SR}.

Definition 2. *For a connected graph $G = (V, E)$ and a vertex $u \in V$, let $MD(G, u)$ be the set of vertices that are maximally distant from vertex u in G.*

The following prerequisite is used in each of the following lemmas and theorems.

Prerequisite 1. *Let $G_i = (V_i, E_i)$, $1 \leq i \leq k$, and $H = (V_H, E_H)$ be $k+1$ vertex disjoint, connected graphs, $|V_i| \geq 2$, $u_i \in V_i$ for $i = 1, \dots, k$, $|V_H| \geq 2$, $v_1, \dots, v_k \in V_H$, let $G_{1,\dots,k}$ be the disjoint union of $G_1, \dots, G_k$, and let*

$$J = (V_J, E_J) = G_{1,\dots,k} \circ_{(u_1,\dots,u_k \to v_1,\dots,v_k)} H.$$

The vertices $v_1, \dots, v_k$ of H do not need to be distinct.

Lemma 1. *Let Prerequisite 1 be given. Then the vertices* $v_1, \ldots, v_k$ *have no incident edges in* J_{SR}.

Proof. The vertices $v_1, \ldots, v_k$ are separation vertices in J, because all graphs G_i, $1 \le i \le k$, and H are connected and have at least two vertices. Separation vertices are not maximally distant from any vertex. $\square$

Lemma 2. *Let Prerequisite 1 be given. Let* $w \in V_H$*, then*

$$MD(\,J,\,w\,) = \left(\begin{array}{cc} & MD(\,H,\,w\,) \setminus \{v_1, \ldots, v_k\} \\ \cup & \bigcup_{1 \le i \le k} MD(\,G_i,\,u_i\,) \end{array} \right).$$

Proof. This follows from the fact that each shortest path in J between w and a vertex of G_i for $1 \le i \le k$ passes vertex v_i and each shortest path in J between w and a vertex of H does not pass a vertex outside of H. Separation vertices are not maximally distant from any vertex, see also Lemma 1. $\square$

In Figure 2 the vertices $\{\mathsf{a,o,e,g,t,s,r}\}$ are maximally distant from vertex l in J.

Lemma 3. *Let Prerequisite 1 be given. Let* $u'_j \in V_j \setminus \{u_j\}$ *for some* j, $1 \le j \le k$. *Then*

$$MD(\,J,\,u'_j\,) = \left(\begin{array}{cc} & MD(\,G_j,\,u'_j\,) \setminus \{u_j\} \\ \cup & MD(\,H,\,v_j\,) \setminus \{v_1, \ldots, v_k\} \\ \cup & \bigcup_{1 \le i \le k, i \ne j} MD(\,G_i,\,u_i\,) \end{array} \right).$$

Proof. This follows from the fact that each shortest path in J between u'_j and a vertex of H or G_i for $i \ne j$ passes vertex v_j and each shortest path in J between u'_j and a vertex of $(V_j \setminus \{u_j\}) \cup \{v_j\}$ do not pass a vertex outside of $(V_j \setminus \{u_j\}) \cup \{v_j\}$. Separation vertices are not maximally distant from any vertex, see also Lemma 1. Note that vertex u_j can be maximally distant from u'_j in G_j, but is not a vertex of J. This is why we need to remove u_j as well from $\mathrm{MD}(\,G_j,\,u'_j\,)$. $\square$

In Figure 2 the vertices $\{\mathsf{l,e,g,t,s,r}\}$ are maximally distant from vertex a in J. Note that vertex c is also maximally distant from vertex a in G_1, but is excluded from $\mathrm{MD}(\,G_1,\,\mathsf{a}\,)$.

The next lemmas characterise the edges of J_{SR} between vertices of $V_i \setminus \{u_i\}$ and vertices of $V_j \setminus \{u_j\}$ for $i \ne j$, and the edges between vertices of $V_i \setminus \{u_i\}$ and vertices of V_H.

Lemma 4. *Let Prerequisite 1 be given. Then for each vertex* $u'_i \in V_i \setminus \{u_i\}$ *for some* i, $1 \le i \le k$, *and each vertex* $v' \in V_J \setminus V_i$*, the following statements hold true.*

1. *If* u'_i *is maximally distant from* u_i *in* G_i*, or equivalently maximally distant from* v_i *in* J*, then* u'_i *is maximally distant from* v' *in* J.
2. *If* v' *is maximally distant from* v_i *in* J*, then* v' *is maximally distant from* u'_i *in* J.

Proof. This follows again from the fact that each shortest path in J between u'_i and v' passes vertex v_i, and thus $d_J(u'_i, v') = d_J(u'_i, v_i) + d_J(v_i, v')$. $\square$

In Figure 2 vertex a is maximally distant from vertex c in G_1, thus vertex a is maximally distant from all vertices except for b and d in J. Also, vertex e is maximally distant from vertex f in G_2, thus the vertices a and e are mutually maximally distant in J.

Lemma 5. *Let Prerequisite 1 be given. Two vertices $u'_i \in V_i \setminus \{u_i\}$ and $u'_j \in V_j \setminus \{u_j\}$ for $i \neq j$ are mutually maximally distant in J if and only if u'_i is maximally distant from u_i in G_i and u'_j is maximally distant from u_j in G_j.*

Proof.
"$\Rightarrow$" Let u'_i be maximally distant from u_i in G_i and u'_j be maximally distant from u_j in G_j. By Lemma 4, u'_i is maximally distant from each vertex of $V_J \setminus V_i$ and u'_j is maximally distant to each vertex of $V_J \setminus V_j$. Thus u'_i and u'_j are mutually maximally distant in J and $\{u'_i, u'_j\}$ is an edge in J_{SR}.

"$\Leftarrow$" Since each path between u'_i and u'_j in J passes vertex u_i (and vertex u_j), the following statement holds true. If u'_i and u'_j are mutually maximally distant in J, then u'_i is maximally distant from u_i in G_i and u'_j is maximally distant from u_j in G_j. □

Lemma 5 identifies the edges of J_{SR} between $u'_i \in V_i$ and $u'_j \in V_j$ for $i \neq j$.

Next we identify the edges of J_{SR} between two vertices of $V_i \setminus \{u_i\}$ and between two vertices of $V_H \setminus \{v_1, \ldots, v_k\}$.

Lemma 6. *Let Prerequisite 1 be given.*

1. *Two vertices $u'_i, u''_i \in V_i \setminus \{u_i\}$ are mutually maximally distant in J if and only if they are mutually maximally distant in G_i.*
2. *Two vertices $v', v'' \in V_H \setminus \{v_1, \ldots, v_k\}$ are mutually maximally distant in J if and only if they are mutually maximally distant in H.*

Proof. The statements follow from the facts that each shortest path between u'_i and u''_i in J does not pass a vertex of $V_J \setminus (V_i \cup \{v_i\})$ and each shortest path between v' and v'' in J does not pass a vertex of $V_1 \cup \ldots \cup V_k$. □

The following theorem follows from Lemma 5 and Lemma 6 and characterises all edges of J_{SR}. Figure 3 shows the strong resolving graph J_{SR} of J from Figure 2.

Theorem 1. *Let Prerequisite 1 be given. The strong resolving graph J_{SR} has an edge $\{w_1, w_2\}$ if and only if $w_1, w_2 \notin \{v_1, \ldots, v_k\}$ and*

1. *$w_1, w_2 \in V_i$ for some i, $1 \leq i \leq k$, and w_1 and w_2 are mutually maximally distant in G_i,*
2. *$w_1, w_2 \in V_H$ and w_1 and w_2 are mutually maximally distant in H,*
3. *$w_1 \in V_i$ and $w_2 \in V_j$ for some i, j, $1 \leq i < j \leq k$, and w_1 is maximally distant from u_i in G_i and w_2 is maximally distant from u_j in G_j, or*
4. *if $w_1 \in V_i$ for some i, $1 \leq i \leq k$, $w_2 \in V_H$, w_1 is maximally distant from u_i in G_i and w_2 is maximally distant from v_i in H.*

5. A minimum vertex cover

Theorem 1 characterises the edges in the strong resolving graph J_{SR} as follows, see Figure 3.

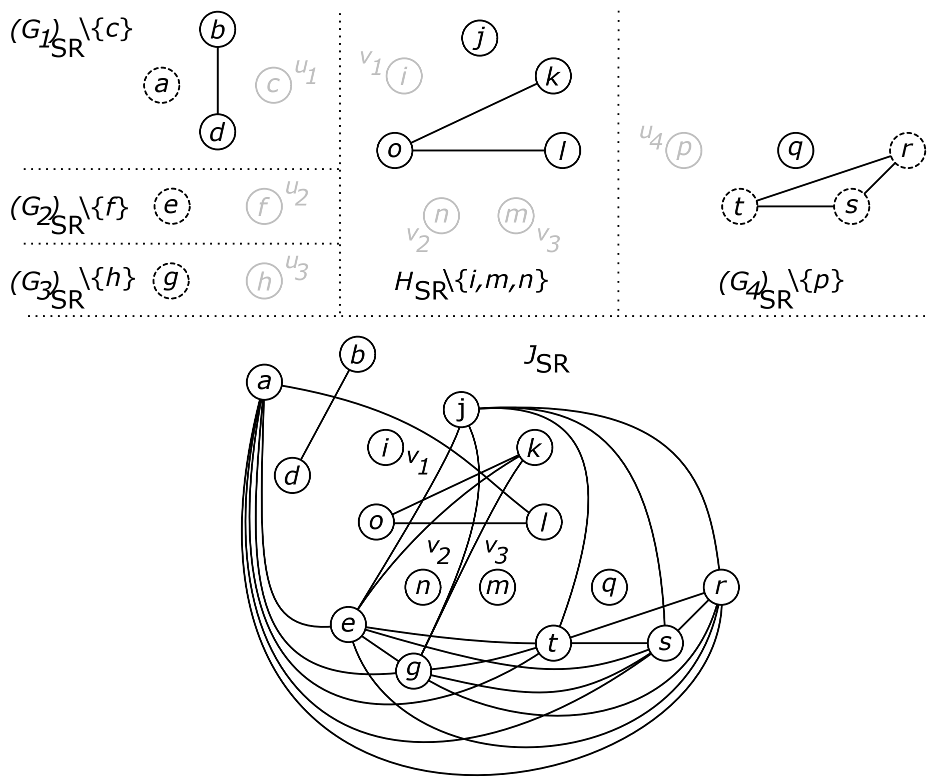


Figure 3: The graphs $(G_1)_{\mathrm{SR}} \setminus \{c\}$, $(G_2)_{\mathrm{SR}} \setminus \{f\}$, $(G_3)_{\mathrm{SR}} \setminus \{h\}$, $(G_4)_{\mathrm{SR}} \setminus \{p\}$, $H_{\mathrm{SR}} \setminus \{i,m,n\}$, and J_{SR}, where G_1, G_2, G_3, G_4, H, and J are from Figure 2. The dashed vertices in $(G_1)_{\mathrm{SR}}$, $(G_2)_{\mathrm{SR}}$, $(G_3)_{\mathrm{SR}}$, and $(G_4)_{\mathrm{SR}}$ are the vertices which are maximally distant from u_1, u_2, u_3, and u_4 in G_1, G_2, G_3, and G_4, respectively. The grey vertices are the vertices that are merged to separation vertices of J. These vertices are not taken into account when computing minimum strong resolving sets for J.

- The edges considered in Case 1 are the edges of $(G_i)_{\mathrm{SR}} \setminus \{u_i\}$.
- The edges considered in Case 2 are the edges of $H_{\mathrm{SR}} \setminus \{v_1, \ldots, v_k\}$.
- The edges considered in Case 3 are the edges of a complete k-partite graph $K_{n_1,\ldots,n_k}$, $n_i = |\mathrm{MD}(\,G_i,\, u_i\,)|$, with vertex set

$$\bigcup_{1 \le i \le k} \mathrm{MD}(\,G_i,\, u_i\,)$$

and edge set

$$\{\{w_1, w_2\} \mid w_1, \in \mathrm{MD}(\,G_i,\, u_i\,), w_2, \in \mathrm{MD}(\,G_j,\, u_j\,), 1 \le i < j \le k\},$$

.

- The edges considered in Case 4 are the edges of k complete bipartite graphs K_{n_i,m_i}, $1 \le i \le k$, $n_i = |\mathrm{MD}(G_i, u_i)|$, $m_i = |\mathrm{MD}(H, v_i) \setminus \{v_1, \ldots, v_k\}|$ with vertex set

$$\mathrm{MD}(G_i, u_i) \cup (\mathrm{MD}(H, v_i) \setminus \{v_1, \ldots, v_k\})$$

and edge set

$$\{\{w_1, w_2\} \mid w_1, \in \mathrm{MD}(G_i, u_i), w_2, \in \mathrm{MD}(H, v_i) \setminus \{v_1, \ldots, v_k\}\}.$$

As mentioned in Section 2, a vertex set is a strong resolving set for J if and only if it is a vertex cover for J_{SR}. The strong resolving graph J_{SR} contains the k-partite graph $K_{n_1,\ldots,n_k}$ and the k bipartite graphs K_{n_i,m_i}, $1 \le i \le k$, as subgraphs. That is, each vertex cover U for J_{SR} contains for some j, $1 \le j \le k$, all vertices of the vertex sets $\mathrm{MD}(G_i, u_i)$, $1 \le i \le k$, $i \ne j$, and additionally either all vertices of $\mathrm{MD}(G_j, u_j)$ or all vertices of $\mathrm{MD}(H, v_j) \setminus \{v_1, \ldots, v_k\}$.

To compute the size of a minimum vertex cover of J_{SR}, we use the following notation of a restricted vertex cover.

Notation 1. *For a graph $G = (V, E)$ let*

$$VC(G)$$

be a minimum vertex cover for G. For a graph $G = (V, E)$ and a vertex set $M \subseteq V$ let

$$\overline{VC}(G, M)$$

be a vertex set of minimum size that contains all vertices of M and which is a vertex cover for G.

With the help of Notation 1, a minimum vertex cover for J_{SR} can now easily be specified.

Lemma 7. *Let Prerequisite 1 be given. Let*

$$U_0 = \left(\begin{array}{ll} & VC(H_{\mathrm{SR}} \setminus \{v_1, \ldots, v_k\}) \\ \cup & \bigcup_{1 \le i \le k} \overline{VC}((G_i)_{\mathrm{SR}} \setminus \{u_i\}, MD(G_i, u_i)) \end{array} \right)$$

and

$$U_j = \left(\begin{array}{ll} & \overline{VC}(H_{\mathrm{SR}} \setminus \{v_1, \ldots, v_k\}, MD(H, v_j) \setminus \{v_1, \ldots, v_k\}) \\ \cup & VC((G_j)_{\mathrm{SR}} \setminus \{u_j\}) \\ \cup & \bigcup_{1 \le i \le k, i \ne j} \overline{VC}((G_i)_{\mathrm{SR}} \setminus \{u_i\}, MD(G_i, u_i)) \end{array} \right)$$

for $j = 1, \ldots, k$.

Each vertex sets $U_0, U_1, \ldots, U_k$ is a vertex cover for J_{SR}, and at least one of them is a minimum vertex cover for J_{SR}.

Proof. As mentioned above, J_{SR} contains a complete k-partite subgraph with vertex set

$$\mathrm{MD}(\,G_1,\,u_1\,) \cup \cdots \cup \mathrm{MD}(\,G_k,\,u_k\,)$$

and k bipartite subgraphs with vertex sets

$$\mathrm{MD}(\,G_i,\,u_i\,) \cup (\mathrm{MD}(\,H,\,u_i\,) \setminus \{u_1,\dots,u_k\})$$

for $i = 1,\dots,k$. Each vertex cover of J_{SR} must therefore contain all vertices from all sets $\mathrm{MD}(\,G_1,\,u_1\,),\dots,\mathrm{MD}(\,G_k,\,u_k\,)$ except for one of these sets $\mathrm{MD}(\,G_j,\,u_j\,)$, $1 \le j \le k$, and must additionally contain either all vertices of $\mathrm{MD}(\,G_j,\,u_j\,)$ or all vertices of $\mathrm{MD}(\,H,\,v_j\,) \setminus \{v_1,\dots,v_k\}$. The edges in J_{SR} which are not incident with the vertices of the selected sets must of course also be covered by a vertex cover. These $k+1$ cases are treated by considering the sets $U_0, U_1, \dots, U_k$. $\square$

The next two lemmas consider the case that graph H has an additional vertex w that can be used to attach J to further graphs, see Figure 4. In this case we compute a minimum vertex cover for $J_{\mathrm{SR}}\setminus\{w\}$ and a minimum vertex cover for $J_{\mathrm{SR}}\setminus\{w\}$ that contains all vertices of $\mathrm{MD}(\,J,\,w\,)$.

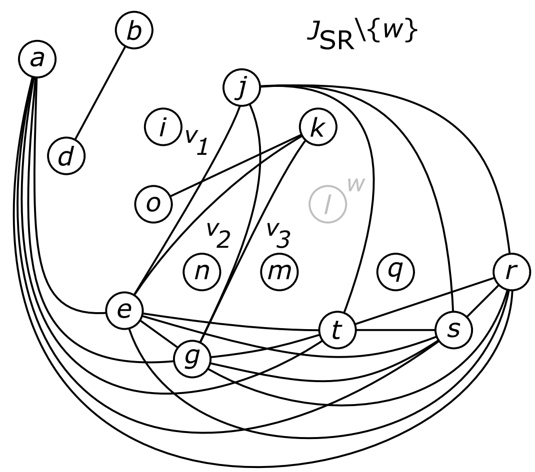


Figure 4: The strong resolving graph J_{SR} without vertex w, where G_1, G_2, G_3, G_4, and H are from Figure 2. The four vertex sets $\{a\}$, $\{e\}$, $\{g\}$, $\{r,s,t\}$ form a complete 4-partite subgraph of J_{SR}. Vertex a is the only vertex that is maximally distant from u_1 in G_1, vertex e is the only vertex that is maximally distant from u_2 in G_2, vertex g is the only vertex that is maximally distant from u_3 in G_3, and the vertices r,s,t are the only vertices that are maximally distant from u_4 in G_4. Also, the vertices l and m are maximally distant from v_1, the vertices j and k are maximally distant from v_2, and the vertices i and j are maximally distant from v_3 in H. Therefore, the vertex sets $\{e\}$ and $\{j,k\}$ form a complete bipartite subgraph in J_{SR}, the vertex sets $\{g\}$ and $\{j,k\}$ form a second complete bipartite subgraph and the vertex sets $\{r,s,t\}$ and $\{j\}$ form a third one. Since the vertices i,m and n are separation vertices in J and the vertex l will be a separation vertex later on (see Lemma 8 and Lemma 9), they have no incident edges in J_{SR}.

Lemma 8. *Let Prerequisite 1 be given and* $w \in V_H$. *Let*

$$U_0 = \left(\begin{array}{ll} & VC(\, H_{\mathrm{SR}} \setminus \{v_1, \ldots, v_k, w\}\,) \\ \cup & \bigcup_{1 \le i \le k} \overline{VC}(\,(G_i)_{\mathrm{SR}} \setminus \{u_i\},\, MD(\, G_i,\, u_i\,)\,) \end{array} \right)$$

and

$$U_j = \left(\begin{array}{ll} & \overline{VC}(\, H_{\mathrm{SR}} \setminus \{v_1, \ldots, v_k, w\},\, MD(\, H,\, v_j\,) \setminus \{v_1, \ldots, v_k, w\}\,) \\ \cup & VC(\,(G_j)_{\mathrm{SR}} \setminus \{u_j\}\,) \\ \cup & \bigcup_{1 \le i \le k, i \ne j} \overline{VC}(\,(G_i)_{\mathrm{SR}} \setminus \{u_i\},\, MD(\, G_i,\, u_i\,)\,) \end{array} \right)$$

for $j = 1, \ldots, k$.

Each vertex sets $U_0, U_1, \ldots, U_k$ *is a vertex cover for* $J_{\mathrm{SR}} \setminus \{w\}$, *and at least one of them is a minimum vertex cover for* $J_{\mathrm{SR}} \setminus \{w\}$.

Proof. The only difference between Lemma 7 and Lemma 8 is the additional vertex w in H, which is removed from the computations of the vertex covers, since it becomes a separation vertex in all further compositions. The correctness follows from the reasoning applied in Lemma 7. □

Lemma 9. *Let Prerequisite 1 be given and* $w \in V_H$. *Then*

$$\left(\begin{array}{ll} & \overline{VC}(\, H_{\mathrm{SR}} \setminus \{v_1, \ldots, v_k, w\},\, MD(\, H,\, w\,) \setminus \{v_1, \ldots, v_k, w\}\,) \\ \cup & \bigcup_{1 \le i \le k} \overline{VC}(\,(G_i)_{\mathrm{SR}} \setminus \{u_i\},\, MD(\, G_i,\, u_i\,)\,). \end{array} \right)$$

is a minimum vertex cover for $J_{\mathrm{SR}} \setminus \{w\}$ *that contains all vertices of* $MD(\, J,\, w\,)$.

Proof. Recall the definition of $\mathrm{MD}(\, J,\, w\,)$ from *Lemma* 2. A vertex set of J_{SR} that contains all vertices from the sets $\mathrm{MD}(\, G_i,\, u_i\,)$, $1 \le i \le k$, already covers all edges of the k-partite graph and the k bipartite graphs as defined in the proof of Lemma 7. Therefore, no further case distinctions are necessary for the computation of $\overline{\mathrm{VC}}(\, J_{\mathrm{SR}} \setminus \{w\},\, \mathrm{MD}(\, J,\, w\,)\,)$. □

The results from Lemmas 7 to 9 are summarized by the following theorem.

Theorem 2. *Let Prerequisite 1 be given and* $w \in V_H$. *Then*

$$VC(\, J_{\mathrm{SR}}\,),$$

$$VC(\, J_{\mathrm{SR}} \setminus \{w\}\,),$$

and

$$\overline{VC}(\, J_{\mathrm{SR}} \setminus \{w\},\, MD(\, J,\, w\,)\,)$$

are computable from $G_1, \ldots, G_k, H$, $u_1, \ldots, u_k$, $v_1, \ldots, v_k$, w, *and the following vertex sets.*

1. $VC(\,(G_i)_{\mathrm{SR}} \setminus \{u_i\}\,)$, *for* $i = 1, \ldots, k$,
   (used by Lemma 7: U_j, *Lemma 8:* U_j,*),*
2. $\overline{VC}(\,(G_i)_{\mathrm{SR}} \setminus \{u_i\},\, MD(\, G_i,\, \{u_i\}\,)\,)$ *for* $i = 1, \ldots, k$,
   (used by Lemma 7: U_0, U_j, *Lemma 8:* U_0, U_j, *Lemma 9),*

3. (a) $VC(\, H_{\mathrm{SR}} \setminus \{v_1, \ldots, v_k\}\,)$,
   (used by Lemma 7: U_0),
   (b) $VC(\, H_{\mathrm{SR}} \setminus \{v_1, \ldots, v_k, w\}\,)$,
   (used by Lemma 8: U_0),
4. (a) $\overline{VC}(\, H_{\mathrm{SR}} \setminus \{v_1, \ldots, v_k\},\ MD(\, H,\ \{v_i\}\,) \setminus \{v_1, \ldots, v_k\}\,)$ *for* $i = 1, \ldots, k$,
   (used by Lemma 7: U_j),
   (b) $\overline{VC}(\, H_{\mathrm{SR}} \setminus \{v_1, \ldots, v_k, w\},\ MD(\, H,\ \{v_i\}\,) \setminus \{v_1, \ldots, v_k, w\}\,)$ *for* $i = 1, \ldots, k$,
   (used by Lemma 8: U_j), and
5. $\overline{VC}(\, H_{\mathrm{SR}} \setminus \{v_1, \ldots, v_k, w\},\ MD(\, H,\ \{w\}\,) \setminus \{v_1, \ldots, v_k, w\}\,)$,
   (used by Lemma 9).

In the next section we show how a minimum strong resolving set for a graph G can be efficiently computed using Lemmas 7 to 9, provided that a minimum strong resolving set for the biconnected components can be efficiently computed.

6. The algorithmic frame

The given graph G is first decomposed into its biconnected components. Edges, whose end vertices are both separation vertices or vertices of degree one, are also regarded as biconnected components. Then the *decomposition tree* T for G is built. Tree T contains a so-called *b-node* for each biconnected component of G and a so-called *s-node* for each separation vertex of G. We use variable names with a hat symbol for nodes in trees to distinguish them more clearly from the vertices in graphs. A b-node $\hat{u}$ and an s-node $\hat{v}$ are connected by an edge $\{\hat{u}, \hat{v}\}$ in T if and only if the separation vertex for $\hat{v}$ is part of the biconnected component for $\hat{u}$. This preprocessing to compute T can be done in linear time. Figure 5 shows an example of such a decomposition.

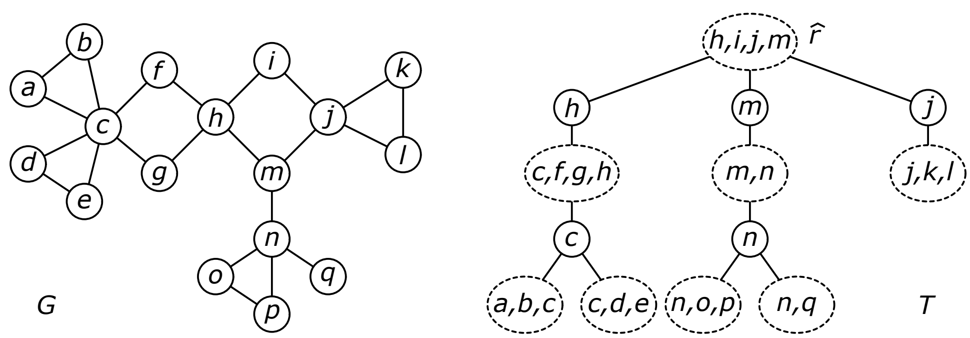


Figure 5: A graph G to the left and its decomposition tree T to the right. The dashed circles are the b-nodes of T for the biconnected components of G. The other nodes are the s-nodes of T for the separation vertices of G. The b-node for the biconnected component of G induced by the vertices *h,i,j,m* has been selected as the root $\hat{r}$ of T.

Computing a minimum strong resolving set for G, or equivalently, computing a minimum vertex cover for G_{SR}, can now be done via a bottom-up processing of G according to the decomposition tree T. Tree T is first oriented by choosing any b-node $\hat{r}$ as the root of T. All leaves of T are b-nodes. The predecessor nodes of the leaves are s-nodes for separation vertices of G that connect the biconnected components of the leaves to the rest of G. The predecessor nodes of the predecessor nodes of the leaves of T are again b-nodes for biconnected components of G at which the biconnected components for the leaves are linked via the separation vertices, and so on.

For a b-node $\hat{u}$ of T let $G(\hat{u})$ be the biconnected component for $\hat{u}$ and $\widetilde{G}(\hat{u})$ be the subgraph of G induced by the vertices of all biconnected components of the b-nodes in the subtree of T with root $\hat{u}$. For an s-node $\hat{v}$ let $\mathrm{s}(\hat{v})$ be the separation vertex for $\hat{v}$.

The bottom-up processing computes for each b-node $\hat{u} \neq \hat{r}$ with predecessor s-node $\hat{w}$ two sets

$$\mathrm{VC}(\,\widetilde{G}(\hat{u})_{\mathrm{SR}} \setminus \{\mathrm{s}(\hat{w})\}\,) \quad \text{and} \quad \overline{\mathrm{VC}}(\,\widetilde{G}(\hat{u})_{\mathrm{SR}} \setminus \{\mathrm{s}(\hat{w})\},\, \mathrm{MD}(\,\widetilde{G}(\hat{u}),\, \mathrm{s}(\hat{w})\,)\,)$$

according to Lemmas 8 and 9, and for the root $\hat{r}$ a set

$$\mathrm{VC}(\,\widetilde{G}(\hat{r})_{\mathrm{SR}}\,)$$

according to Lemma 7. This is done as follows.

1. Suppose $\hat{u}$ is a leaf.
   (a) Suppose $\hat{u} \neq \hat{r}$. Then let $\hat{w}$ be the predecessor s-node of $\hat{u}$ in T. Figure 6 shows an example of this case on the left-hand side.
   Compute the two sets

   $$\mathrm{VC}(\,\widetilde{G}(\hat{u})_{\mathrm{SR}} \setminus \{\mathrm{s}(\hat{w})\}\,) \quad \text{and} \quad \overline{\mathrm{VC}}(\,\widetilde{G}(\hat{u})_{\mathrm{SR}} \setminus \{\mathrm{s}(\hat{w})\},\, \mathrm{MD}(\,\widetilde{G}(\hat{u}),\, \mathrm{s}(\hat{w})\,)\,)$$

   from the biconnected subgraph $G(\hat{u}) = \widetilde{G}(\hat{u})$ of G.
   (b) Suppose $\hat{u} = \hat{r}$.
   Compute the set $\mathrm{VC}(\,G(\hat{u})_{\mathrm{SR}}\,)$ from the biconnected graph $G(\hat{u}) = \widetilde{G}(\hat{u}) = G$.
2. Suppose $\hat{u}$ is not a leaf. Then let $\hat{u}_1, \cdots, \hat{u}_k$ be the successor b-nodes of the successor s-nodes of $\hat{u}$, and $\hat{v}_1, \ldots, \hat{v}_k$ be the predecessor s-nodes of $\hat{u}_1, \cdots, \hat{u}_k$. Note that $\hat{v}_1, \ldots, \hat{v}_k$ do not need to be distinct, because several of the b-nodes $\hat{u}_1, \cdots, \hat{u}_k$ may have the same predecessor s-node. If we replace the separation vertex $\mathrm{s}(\hat{v}_i)$ in subgraph $\widetilde{G}(\hat{u}_i)$ by a new vertex w_i such that all k graphs $\widetilde{G}(\hat{u}_1), \ldots, \widetilde{G}(\hat{u}_k)$, and the biconnected component $G(\hat{u})$ are vertex-disjoint, then

   $$\widetilde{G}(\hat{u}) = (\widetilde{G}(\hat{u}_1) \cup \ldots \cup \widetilde{G}(\hat{u}_k)) \circ_{(w_1,\ldots,w_k \to \mathrm{s}(\hat{v}_1),\ldots,\mathrm{s}(\hat{v}_k))} G(\hat{u}).$$

   (a) Suppose $\hat{u} \neq \hat{r}$. Then let $\hat{w}$ be the predecessor s-node of $\hat{u}$ in T. Figure 6 shows an example of this case on the right-hand side.
   Compute the two sets

   $$\mathrm{VC}(\,\widetilde{G}(\hat{u})_{\mathrm{SR}} \setminus \{\mathrm{s}(\hat{w})\}\,) \quad \text{and} \quad \overline{\mathrm{VC}}(\,\widetilde{G}(\hat{u})_{\mathrm{SR}} \setminus \{\mathrm{s}(\hat{w})\},\, \mathrm{MD}(\,\widetilde{G}(\hat{u}),\, \mathrm{s}(\hat{w})\,)\,)$$

   from the biconnected subgraph $G(\hat{u})$ of G and

   $$\overline{\mathrm{VC}}(\,\widetilde{G}(\hat{u}_i)_{\mathrm{SR}} \setminus \{\mathrm{s}(\hat{v}_i)\},\, \mathrm{MD}(\,\widetilde{G}(\hat{u}_i),\, \mathrm{s}(\hat{v}_i)\,)\,)$$

   for $i = 1, \ldots, k$ according to Lemmas 8 and 9, respectively.

(b) Suppose $\hat{u} = \hat{r}$.
Compute the set $\mathrm{VC}(\,\widetilde{G}(\hat{u})_{\mathrm{SR}}\,)$ from the biconnected subgraph $G(\hat{u})$ and

$$\overline{\mathrm{VC}}(\,\widetilde{G}(\hat{u}_i)_{\mathrm{SR}} \setminus \{\mathrm{s}(\hat{v}_i)\},\, \mathrm{MD}(\,\widetilde{G}(\hat{u}_i),\, \mathrm{s}(\hat{v}_i)\,)\,)$$

for $i = 1, \ldots, k$ according to Lemma 7.

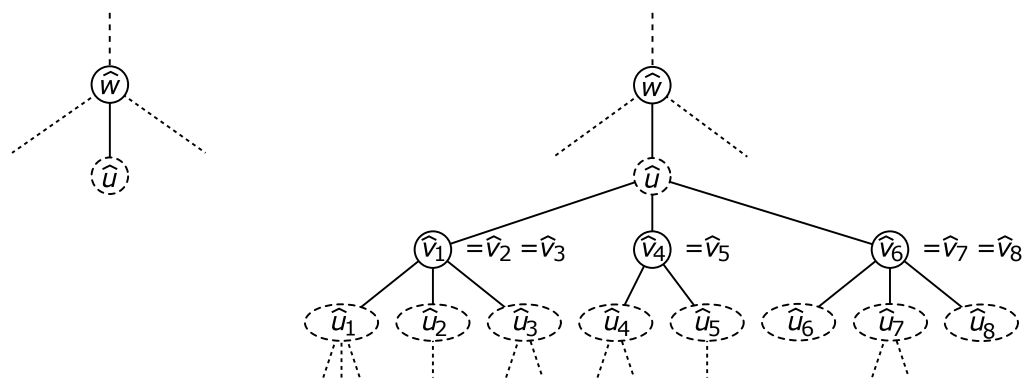


Figure 6: The left-hand side shows the first case where the b-node $\hat{u}$ is a leaf of T. The right-hand side shows the second case in which the b-node $\hat{u}$ is an inner node of T.

Lemmas 7 and 8 consider the k bipartite subgraphs of $\widetilde{G}(\hat{u})_{\mathrm{SR}}$ (graph J_{SR} in Lemmas 7 and 8), between vertices of $\widetilde{G}(\hat{u}_i)$ (graph G_i in Lemmas 7 and 8) and vertices of $G(\hat{u})$ (graph H in Lemmas 7 and 8). To determine which one of these k bipartite subgraphs, if any, needs to be covered by vertices of $G(\hat{u})$, we have defined the sets U_j for $0 \leq j \leq k$ in the proofs of Lemmas 7 and 8. The sizes of these sets can be calculated as follow. Let

$$h_i = |\overline{\mathrm{VC}}(\,\widetilde{G}(\hat{u}_i)_{\mathrm{SR}} \setminus \{\mathrm{s}(\hat{v}_i)\},\, \mathrm{MD}(\,\widetilde{G}(\hat{u}_i),\, \mathrm{s}(\hat{v}_i)\,)\,)|, \text{ and}$$

$$h_i' = |\overline{\mathrm{VC}}(\,G(\hat{u})_{\mathrm{SR}} \setminus \{\mathrm{s}(\hat{v}_1), \ldots, \mathrm{s}(\hat{v}_k)\},\, \mathrm{MD}(\,G(\hat{u}),\, \mathrm{s}(\hat{v}_i)\,) \setminus \{\mathrm{s}(\hat{v}_1), \ldots, \mathrm{s}(\hat{v}_k)\}\,)|,$$

for $1 \leq i \leq k$ for Lemma 7, or

$$h_i' = |\overline{\mathrm{VC}}(\,G(\hat{u})_{\mathrm{SR}} \setminus \{\mathrm{s}(\hat{v}_1), \ldots, \mathrm{s}(\hat{v}_k), \mathrm{s}(\hat{w})\},\, \mathrm{MD}(\,G(\hat{u}),\, \mathrm{s}(\hat{v}_i)\,) \setminus \{\mathrm{s}(\hat{v}_1), \ldots, \mathrm{s}(\hat{v}_k), \mathrm{s}(\hat{w})\}\,)|,$$

for $1 \leq i \leq k$ for Lemma 8. Then h_i is the number of vertices needed to cover the i-th bipartite subgraph with vertices of $\widetilde{G}(\hat{u}_i)$ and h_i' is the number of vertices needed to cover it with vertices of $G(\hat{u})$. Therefore, $|U_i| = |U_0| - h_i + h_i'$ for $1 \leq i \leq k$.

7. Three examples

In this section it is shown that a minimum resolving set of a graph H is computable in linear time if each biconnected component of H is a grid, a cycle, or a co-graph. The linear time computation results from the bottom-up processing introduced in Section 6. If for each biconnected grid, cycle, or co-graph G and for each vertex set $W \subseteq V(G)$

1. $\mathrm{VC}(G_{\mathrm{SR}} \setminus W)$ is computable in linear time,
2. $\overline{\mathrm{VC}}(G_{\mathrm{SR}} \setminus W, \mathrm{MD}(G, u) \setminus W)$ for each $u \in V(G)$ is computable in linear time, and
3. the sizes of all vertex sets $\overline{\mathrm{VC}}(G_{\mathrm{SR}} \setminus W, \mathrm{MD}(G, u) \setminus W)$ for all $u \in V(G)$ is computable in total linear time,

then each step in the bottom-up procedure of Section 6 is performed according to Lemmas 7 to 9 in linear time with respect to the size of $G(\hat{u})$. This means, under these prerequisites the total running time for the computation of a minimum resolving set for H is linear in the size of H.

7.1. 2-dimensional Grids

First consider 2-dimensional $n \times m$ grids G with vertex set

$$V(G) = \{x_{1,1}, \ldots, x_{1,m}, \ldots, x_{n,1}, \ldots, x_{n,m}\}, \quad n, m \geq 2,$$

and edge set

$$E(G) = \{\{x_{i,j}, x_{i',j'}\} \mid (i = i' \wedge |j - j'| = 1) \vee (|i - i'| = 1 \wedge j = j')\}.$$

The strong resolving graph G_{SR} of G only contains the two edges $\{x_{1,1}, x_{n,m}\}$ and $\{x_{1,m}, x_{n,1}\}$.

Observation 1. *Let G be an $n \times m$ grid and $x_{i,j} \in V(G)$.*

1. *If $i \in \{1, n\}$ and $j \in \{1, m\}$, then $MD(G, x_{i,j}) = \{x_{n+1-i,m+1-j}\}$. That is, if $x_{i,j}$ is a corner vertex of the grid, then $MD(G, x_{i,j})$ only contains the opposite corner vertex.*
2. *If $i \in \{1, n\}$ and $j \notin \{1, m\}$, then $MD(G, x_{i,j}) = \{x_{n+1-i,1}, x_{n+1-i,m}\}$. Analogously, if $i \notin \{1, n\}$ and $j \in \{1, m\}$, then $MD(G, x_{i,j}) = \{x_{1,m+1-j}, x_{n,m+1-j}\}$. That is, if $x_{i,j}$ is an edge vertex of the grid, then $MD(G, x_{i,j})$ contains the two corner vertices on the opposite edge.*
3. *If $i \notin \{1, n\}$ and $j \notin \{1, m\}$, then $MD(G, x_{i,j}) = \{x_{1,1}, x_{n,1}, x_{1,m}, x_{n,m}\}$. That is, if $x_{i,j}$ is an inner vertex of the grid, then $MD(G, x_{i,j})$ contains all four corner vertices.*

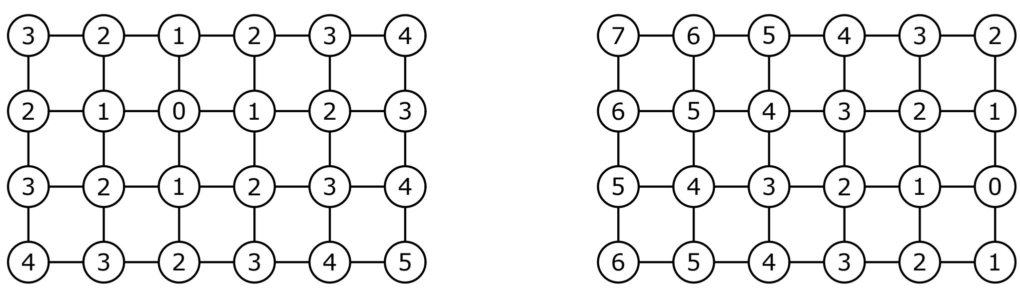


Figure 7: A grid $G_{4,6}$ and the distance of each vertex from the inner vertex $x_{2,3}$ on the left, and from the edge vertex $x_{3,6}$ on the right. This example illustrates the distribution of distances in a grid, that only corner vertices are maximally distant from other vertices, and which corner vertices those are for different types of vertex in the grid.

Figure 7 shows an example of a 4×6 grid and the distances of the vertices to illustrate the previous observation. Let $W \subseteq V(G)$. It is straight forward to see, that a set

$$\mathrm{VC}(\, G_{\mathrm{SR}} \setminus W\,)$$

can be computed in linear time, a set

$$\overline{\mathrm{VC}}(\, G_{\mathrm{SR}} \setminus W,\, \mathrm{MD}(\, G,\, u\,) \setminus W\,)$$

can be computed in linear time for each vertex $u \in V(G)$, and the sizes of all vertex sets

$$\overline{\mathrm{VC}}(\, G_{\mathrm{SR}} \setminus W,\, \mathrm{MD}(\, G,\, u\,) \setminus W\,)$$

for all $u \in V(G)$ can be computed in total linear time. Therefore, the following theorem follows.

Theorem 3. *A minimum strong resolving set for a graph $G = (V, E)$, in that each biconnected component is a grid, can be computed in time $\mathcal{O}(|V| + |E|)$.*

The results can easily be extended to d-dimensional grids for $d > 2$.

7.2. Cycles

Next consider cycles G with vertex set

$$V(G) = x_0, x_1, \ldots, x_{n-1}, \quad n \geq 3,$$

and edge set

$$E(G) = \{\{x_0, x_1\}, \{x_1, x_2\}, \ldots, \{x_{n-1}, x_0\}\}.$$

Let $W \subseteq V(G)$.

If n is even, then G_{SR} has $n/2$ edges, such that no two edges have a vertex in common. In this case, $\mathrm{MD}(\, G,\, x_i\,)$ contains the single vertex x_j with $j = (i + n/2) \bmod n$. If n is odd, then G_{SR} is a cycle and $\mathrm{MD}(\, G,\, x_i\,)$ contains the two vertices x_j and x_{j+1} with $j = (i + \lfloor n/2 \rfloor) \bmod n$, see Figure 8 for an example.

In both cases, $G_{\mathrm{SR}} \setminus W$ is a collection of paths. A minimum vertex cover of $G_{\mathrm{SR}} \setminus W$ has $\lfloor l/2 \rfloor$ vertices from each of those paths that consists of l vertices. However,

$$\overline{\mathrm{VC}}(\, G_{\mathrm{SR}} \setminus W,\, \mathrm{MD}(\, G,\, x_i\,) \setminus W\,)$$

may have some additional vertices depending on which paths the vertices from $\mathrm{MD}(\, G(\hat{u}),\, x_i\,)$ belong to. Since $\mathrm{MD}(\, G,\, x_i\,)$ contains at most two vertices, the size of

$$\overline{\mathrm{VC}}(\, G_{\mathrm{SR}} \setminus W,\, \mathrm{MD}(\, G,\, x_i\,) \setminus W\,)$$

can easily be computed in constant time. Here the length of the paths on which the vertices are located, and in addition, if both vertices are located on the same path p, the distance between them on p must be taken into account. Also observe, that the vertices of

MD(G, x_i) are always at the end of paths if x_i is one of the vertices which are removed from G_{SR}.

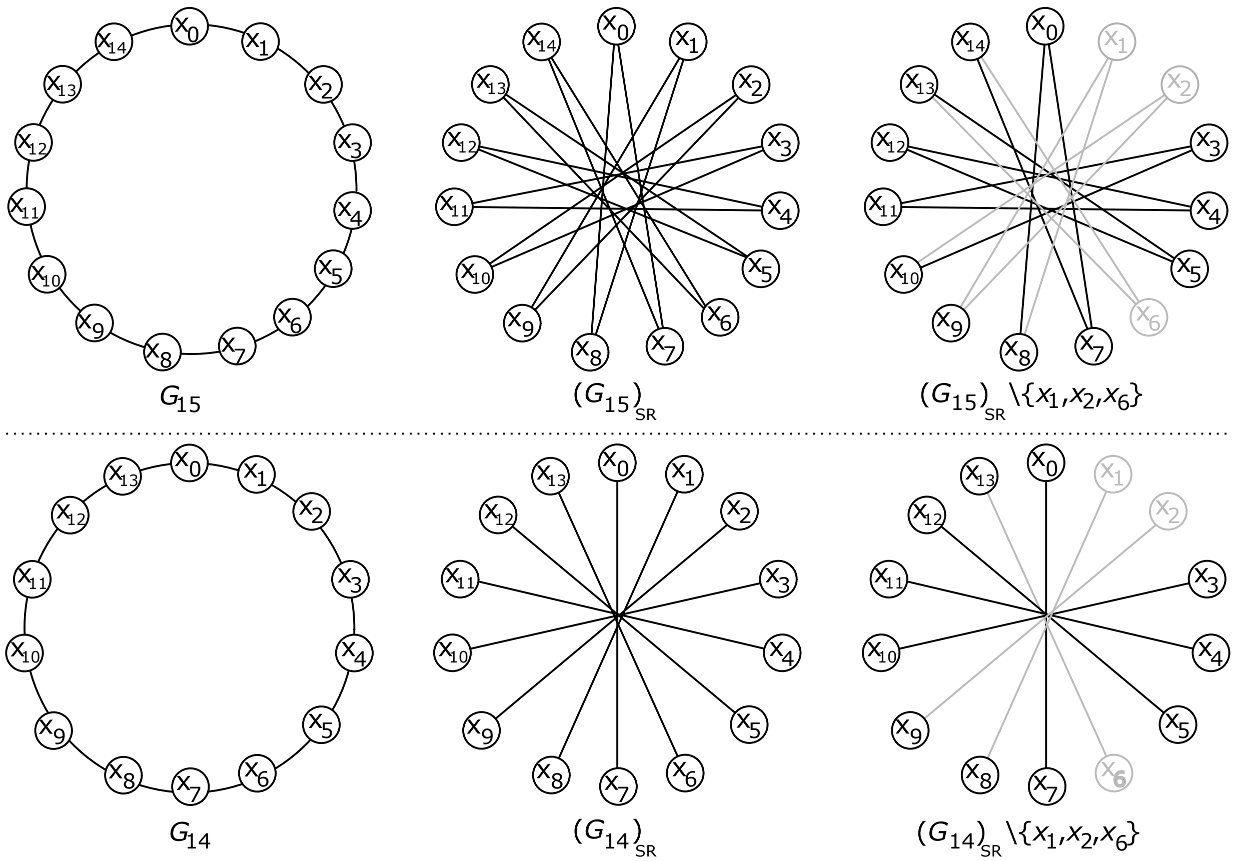


Figure 8: Two cycles of 15 and 14 vertices on the left, the corresponding strong resolving graphs $(G_{15})_{\mathrm{SR}}$ and $(G_{14})_{\mathrm{SR}}$ in the middle, and the strong resolving graphs without the vertices x_1, x_2, x_6 on the right.

Thus a set

$$\mathrm{VC}(\, G_{\mathrm{SR}} \setminus W \,)$$

can be computed in linear time, a set

$$\overline{\mathrm{VC}}(\, G_{\mathrm{SR}} \setminus W,\ \mathrm{MD}(\, G,\ u \,) \setminus W \,)$$

can be computed in linear time for each vertex $u \in V(G)$, and the sizes of all vertex sets

$$\overline{\mathrm{VC}}(\, G_{\mathrm{SR}} \setminus W,\ \mathrm{MD}(\, G,\ u \,) \setminus W \,)$$

for all $u \in V(G)$ can be computed in total linear time. Therefore, the following theorem follows.

Theorem 4. *A minimum strong resolving set for a graph $G = (V, E)$, in that each biconnected component is a cycle, can be computed in time $\mathcal{O}(|V| + |E|)$.*

7.3. Co-Graphs

Finally consider a more complex example in that G is a co-graph. Co-graphs can be defined as follows.

Definition 3 (Co-Graphs and Co-Trees). *[26]*

- *A graph G that consists of a single vertex u is a co-graph. The co-tree T for G consists of a single node $\hat{u}$ associated with vertex u of G. Node $\hat{u}$ is the* root *of T. Let* $\mathrm{vertex}(\hat{u}) = u$ *and* $\mathrm{node}(u) = \hat{u}$. *Note that* $\mathrm{vertex}(\hat{u})$ *is only defined for leaves $\hat{u}$ of T.*

- *If $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are two co-graphs, then the disjoint* union *of G_1 and G_2, denoted by $G_1 \cup G_2$, is a co-graph G with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$. Let T_1 and T_2 be co-trees for G_1 and G_2 with root $\hat{u}_1$ and $\hat{u}_2$, respectively. Then tree T defined by the disjoint union of T_1 and T_2 with an additional node $\hat{u}$ and two additional edges $\{\hat{u}, \hat{u}_1\}$ and $\{\hat{u}, \hat{u}_2\}$ is a co-tree for G. Node $\hat{u}$ is the* root *of T labelled by $\cup$. Node $\hat{u}_1$ and $\hat{u}_2$ are successor nodes of $\hat{u}$. Node $\hat{u}$ is the predecessor node of $\hat{u}_1$ and $\hat{u}_2$.*

- *If $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are two co-graphs, then the* join *of G_1 and G_2, denoted by $G_1 \times G_2$, is a co-graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2 \cup \{\{u, v\} \mid u \in V_1, v \in V_2\}$. Let T_1 and T_2 be co-trees for G_1 and G_2 with root $\hat{u}_1$ and $\hat{u}_2$, respectively. Then tree T defined by the disjoint union of T_1 and T_2 with an additional node $\hat{u}$ and two additional edges $\{\hat{u}, \hat{u}_1\}$ and $\{\hat{u}, \hat{u}_2\}$ is a co-tree for G. Node $\hat{u}$ is the* root *of T labelled by $\times$. Node $\hat{u}_1$ and $\hat{u}_2$ are successor nodes of $\hat{u}$. Node $\hat{u}$ is the predecessor node of $\hat{u}_1$ and $\hat{u}_2$.*

Co-graphs can be recognized in linear time, see [27]. This includes the computation of a co-tree. Co-graphs are the graphs that do not contain an induced P_4, a path with four vertices, or in other words, connected co-graphs are graphs with diameter at most 2.

Two adjacent vertices u and v of a graph G are *true twins* if $N_G[u] = N_G[v]$. In [28] it is shown that the strong resolving graph of a connected co-graph is again a co-graph and that two vertices u and v are mutually maximally distant in G if and only if they are either not adjacent in G or if they are true twins in G.

For the analysis of co-graphs, the so-called *canonical co-tree* is a useful data structure. The canonical co-tree results from combining successive union and join operations into one union and join operation, see also Figure 9. It can also be computed in linear time, see [29]. In a *canonical co-tree* each inner node may have more than two successor nodes. The successor nodes of a union node are join nodes or leaves, the successor nodes of a join node are union nodes or leaves. Two vertices u and v of G are true twins in G if and only if $\mathrm{node}(u)$ and $\mathrm{node}(v)$ are leaves of a common join node in the *canonical co-tree* for G.

A canonical co-tree $T^{\mathrm{c}}_{G_{\mathrm{SR}}}$ for G_{SR} can be easily constructed from a canonical co-tree T^{c}_{G} for G by first transforming the union nodes into join nodes and the join nodes into union nodes. The corresponding graph of the new co-tree has an edge between two vertices u and

v if and only if there was no edge between u and v beforehand. If now a union node $\hat{u}$ of $T^{\mathrm{c}}_{G_{\mathrm{SR}}}$ (that is a join node in T^{c}_{G}) has two or more successor nodes $\hat{u}_1, \ldots, \hat{u}_k$ that are leaves, then $\hat{u}_1, \ldots, \hat{u}_k$ are true twins in G and we detach them from $\hat{u}$ in $T^{\mathrm{c}}_{G_{\mathrm{SR}}}$, insert a new join node $\hat{w}$ in $T^{\mathrm{c}}_{G_{\mathrm{SR}}}$ as a successor node of $\hat{u}$ and append the detached leaves $\hat{u}_1, \ldots, \hat{u}_k$ to the new join node $\hat{w}$. We mark this new join node $\hat{w}$ as a *twin-join node*, see also Figure 9. If all successor nodes of $\hat{u}$ in $T^{\mathrm{c}}_{G_{\mathrm{SR}}}$ were leaves before, then $\hat{u}$ might have only the one successor node $\hat{w}$ after the modification. It is important for our forthcoming processing to preserve this structure and to not clean it up by attaching the leaves of $\hat{w}$ to the predecessor node of $\hat{u}$. The resulting tree $T^{\mathrm{c}}_{G_{\mathrm{SR}}}$ is a canonical co-tree for G_{SR}, because two vertices u and v are adjacent in G_{SR} if and only if they are either not adjacent in G or if they are true twins in G if and only if they are mutually maximally distant in G.

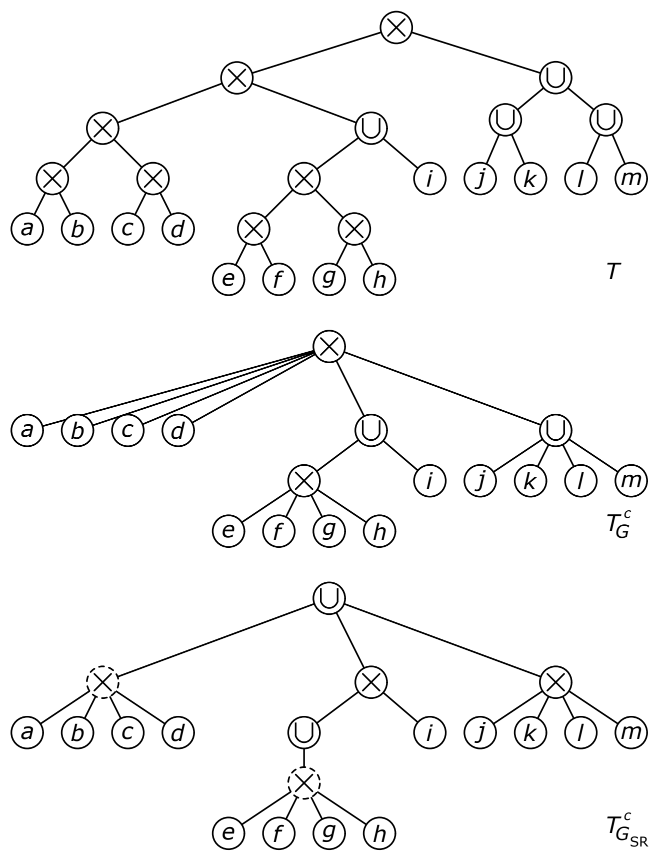


Figure 9: A co-tree T for a co-graph G, a canonical co-tree T^{c}_{G} for G, and a canonical co-tree for $T^{\mathrm{c}}_{G_{\mathrm{SR}}}$ for G_{SR}. The two nodes with the dashed circles are twin-join nodes in $T^{\mathrm{c}}_{G_{\mathrm{SR}}}$. These nodes are not in T^{c}_{G}.

The size of a minimum strong resolving set for a co-graph G can be computed in linear time by computing the size of a minimum vertex cover for co-graph G_{SR}. We use the following notations to describe this well-known computation procedure. Let T be a co-tree for a co-graph G with root $\hat{r}$. For a node $\hat{u}$ of T, let $T(\hat{u})$ be the subtree of T with root $\hat{u}$ and $\mathrm{n}(\hat{u})$ be the number of leaves in $T(\hat{u})$.

The size of a minimum vertex cover for a co-graph G_{SR} with canonical co-tree $T^c_{G_{\mathrm{SR}}}$ and root $\hat{r}$ can be computed in linear time by the bottom-up processing of $T^c_{G_{\mathrm{SR}}}$ with Algorithm 1. The result is $\mathrm{vc}(\hat{r})$, see also Figure 10. Algorithm 1 can easily be modified to compute a set

$$\mathrm{VC}(\,G_{\mathrm{SR}}\,)$$

in linear time. A simple way to reduce the computation to the vertices of $V(G) \setminus W$ is to redefine $\mathrm{n}(\hat{u})$ be the number of leaves in $T(\hat{u})$ that are not in W. Thus a set

$$\mathrm{VC}(\,G_{\mathrm{SR}} \setminus W\,)$$

is also computable in linear time for each set $W \subseteq V(G)$.

Algorithm 1 $(T^c_{G_{\mathrm{SR}}})$

for (*each node* $\hat{u}$ *of* $T^c_{G_{\mathrm{SR}}}$) **do**
  // -1 *means undefined*
  $\mathrm{vc}(\hat{u}) \leftarrow -1;$
end
for (*each leaf* $\hat{u}$ *of* $T^c_{G_{\mathrm{SR}}}$) **do**
  $\mathrm{vc}(\hat{u}) \leftarrow 0;$
end
while $\left(\begin{array}{l} \textit{there is a node } \hat{u} \textit{ of } T^c_{G_{\mathrm{SR}}} \textit{ with successor nodes } \hat{u}_1, \ldots, \hat{u}_k \\ \textit{such that } \mathrm{vc}(\hat{u}) = -1 \textit{ and } \mathrm{vc}(\hat{u}_i) \geq 0, 1 \leq i \leq k \end{array}\right)$ **do**
  if ($\hat{u}$ *is a union node in* $T^c_{G_{\mathrm{SR}}}$) **then**
    $\mathrm{vc}(\hat{u}) \leftarrow \mathrm{vc}(\hat{u}_1) + \cdots + \mathrm{vc}(\hat{u}_k);$
  end
  else
    // $\hat{u}$ *is a join node in* $T^c_{G_{\mathrm{SR}}}$
    $\mathrm{vc}(\hat{u}) \leftarrow \min_{1 \leq i \leq k} (\mathrm{n}(\hat{u}) - \mathrm{n}(\hat{u}_i) + \mathrm{vc}(\hat{u}_i));$
  end
end

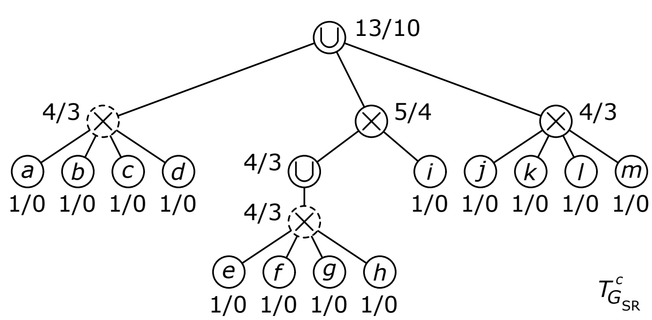


Figure 10: The computation of a minimum vertex cover for co-graph G_{SR} defined by the canonical co-tree $T^c_{G_{\mathrm{SR}}}$ of Figure 9. The nodes $\hat{u}$ of tree $T^c_{G_{\mathrm{SR}}}$ are labeled $\mathrm{n}(\hat{u})/\mathrm{vc}(\hat{u})$.

Next we need to compute the size of a minimum vertex set that is a vertex cover for G_{SR} and which contains all vertices that are maximally distant from a vertex u in co-graph G. This can be done as follows. Let T^c_G be a canonical co-tree for G and $\hat{w}$ be the first common ancestor of $\hat{u} = \text{node}(u)$ and $\hat{v} = \text{node}(v)$ in T^c_G.

1. If $\hat{w}$ is a union node, then u and v are not adjacent in G and thus v is maximally distant from u (and u is maximally distant from v). Note that connected co-graphs have diameter at most 2.
2. If $\hat{w}$ is a join node, then u and v are adjacent in G and v is maximally distant from u if and only if $N_G[v] \subseteq N_G[u]$. This is the case if and only if all nodes on the path between $\hat{u}$ and $\hat{w}$ in T are join nodes. If T^c_G is a canonical co-tree, then this is the case if and only if $\hat{u}$ is a successor node of $\hat{w}$.

For a node $\hat{u}$ of T^c_G let

$$V(\hat{u}) = \{\text{vertex}(\hat{v}) \,|\, \hat{v} \text{ is a leaf of } T^c_G(\hat{u})\}.$$

A vertex v of G which is not in $V(\hat{u})$ is either adjacent to all vertices of $V(\hat{u})$ or to none of them. This depends on whether the first common predecessor of $\hat{u}$ and node(v) in T is a join node or a union node, respectively.

The following algorithm computes the size of a minimum vertex set that is a vertex cover for G_{SR} and which contains all vertices that are maximally distant from a vertex w in G. We assume that vc$(\hat{u})$ is already computed for each node $\hat{u}$ of $T^c_{G_{\text{SR}}}$ by Algorithm 1.

Algorithm 2 $(T^c_G, T^c_{G_{\text{SR}}}, w)$

```
ŵ ← node(w);
if (ŵ = r̂) then
    return 0;
end
Let v̂ be the predecessor node of ŵ in T^c_G;
h(v̂) ← n(v̂) − 1;
while (v̂ ≠ r̂) do
    Let û ← v̂;
    Let v̂ be the predecessor node of û in T^c_{G_SR};
    Let û_1, ..., û_k be the successor nodes of v̂ in T^c_{G_SR} without node û;
    if (v̂ is a join node in T^c_{G_SR}) then
        // v̂ is a union node in T^c_G
        h(v̂) ← h(û) + n(û_1) + ··· + n(û_k);
    end
    else
        // v̂ is a union node in T^c_{G_SR}
        // v̂ is a join node in T^c_G
        h(v̂) ← h(û) + vc(û_1) + ··· + vc(û_k);
    end
end
return h(r̂);
```

Algorithm 2 defines for each node $\hat{u}$ of T^c_G a variable $\mathrm{h}(\hat{u})$ such that $\mathrm{h}(\hat{r})$ is the size of a minimum vertex set that is a vertex cover for $G(\hat{u})_{\mathrm{SR}}$ and which contains all vertices that are maximally distant from a vertex w in $G(\hat{u})_{\mathrm{SR}}$. Algorithm 2 initially sets variable $\mathrm{h}(\hat{v})$ to $\mathrm{n}(\hat{v}) - 1$ for the predecessor node $\hat{v}$ of $\hat{w} = \mathrm{node}(w)$ in T^c_G. To explain the correctness of this instruction, we distinguish between the 4 cases shown in Figure 11. In the cases (a), (b), and (c) node $\hat{v}$ is the predecessor node of $\hat{w}$ in T^c_G. In these cases, all vertices of $V(\hat{v}) \setminus w$ are maximally distant from w in G, based on the second consideration above. One vertex is subtracted here, since vertex w is not maximally distant from w itself if G is connected and has at least two vertices. In case (d), node $\hat{v}$ is a union node in T^c_G and thus all vertices of $V(\hat{v}) \setminus w$ are maximal distant from w based on the first consideration above. If $\hat{v}$ is a node further up on the path to the root $\hat{r}$, we only have to distinguish whether $\hat{v}$ is a join or a union node in $T^c_{G_{\mathrm{SR}}}$. If $\hat{v}$ is a join node in $T^c_{G_{\mathrm{SR}}}$, then $\hat{v}$ is a union node in T^c_G and all vertices of $V(\hat{v}) \setminus V(\hat{u})$ are maximally distant from w. If $\hat{v}$ is a union node in $T^c_{G_{\mathrm{SR}}}$, then the minimum vertex covers of the subgraphs induced by the vertex sets $V(\hat{u}_i)$ of the successor nodes $\hat{u}_i$ of $\hat{v}$ without node $\hat{u}$ have to be merged. In this case, node $\hat{v}$ is a join node in T^c_G and the vertices of $V(\hat{v}) \setminus V(\hat{u})$ are not maximal distant from w.

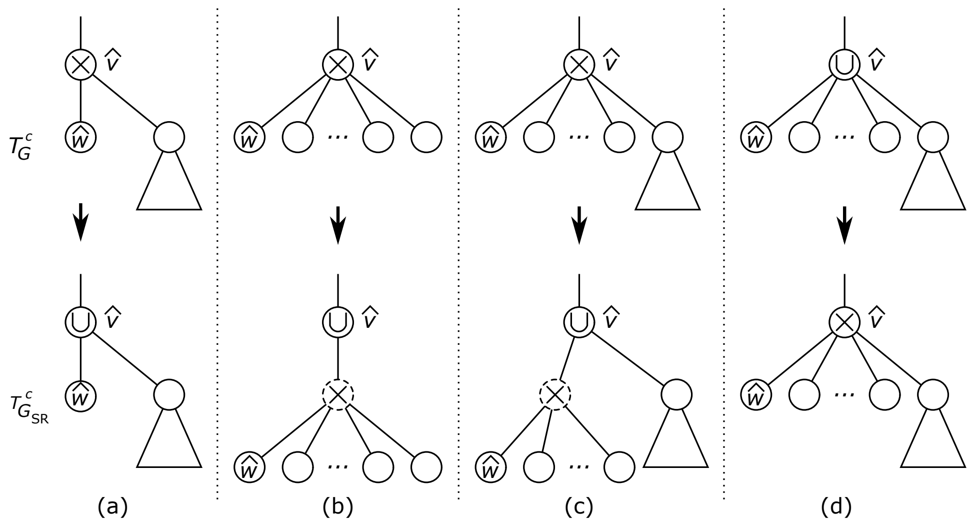


Figure 11: The 4 cases when node $\hat{v}$ is the predecessor node of $\hat{w}$ in T^c_G.

This shows that a vertex set

$$\overline{\mathrm{VC}}(\, G_{\mathrm{SR}},\, \mathrm{MD}(\, G,\, u\,)\,)$$

is computable in linear time for each vertex $u \in V(G)$. By a redefinition of $\mathrm{n}(\hat{u})$ to be the number of leaves in $T(\hat{u})$ that are not in a vertex set W, a vertex set

$$\overline{\mathrm{VC}}(\, G_{\mathrm{SR}} \setminus W,\, \mathrm{MD}(\, G,\, u\,) \setminus W\,)$$

is also computable in linear time for each vertex set $W \subseteq V(G)$.

It remains to show how the sizes of all vertex sets

$$\overline{\mathrm{VC}}(\, G_{\mathrm{SR}} \setminus W,\, \mathrm{MD}(\, G,\, u\,) \setminus W\,)$$

for all $u \in V(G)$ can be computed in total linear time for a vertex set $W \subseteq V(G)$. To achieve this, we calculate the increments of the values $\mathrm{h}(\hat{u})$ at the inner nodes $\hat{u}$ of T^c_G along the path to the root $\hat{r}$ top-down in a preprocessing phase. Algorithm 3 computes all these increments, denoted by $\mathrm{m}(\hat{u})$, in total linear time, see also Figure 12.

Algorithm 3 $(T^c_G, T^c_{G_{\mathrm{SR}}})$

for (*each inner node* $\hat{u}$ *of* T^c_G) **do**
  // -1 *means undefined*
  $\mathrm{m}(\hat{u}) \leftarrow -1$;
end
$\mathrm{m}(\hat{r}) \leftarrow 0$;
while $\left(\begin{array}{l} \textit{there is an inner node } \hat{u} \textit{ in } T^c_G \textit{ with predecessor node } \hat{v} \\ \textit{such that } \mathrm{m}(\hat{u}) = -1 \textit{ and } \mathrm{vc}(\hat{v}) \geq 0 \end{array} \right)$ **do**
  Let $\hat{u}_1, \ldots, \hat{u}_k$ be the successor nodes of $\hat{v}$ in $T^c_{G_{\mathrm{SR}}}$ without node $\hat{u}$;
  if ($\hat{v}$ *is a join node in* $T^c_{G_{\mathrm{SR}}}$) **then**
    // $\hat{v}$ *is a union node in* T^c_G
    $\mathrm{m}(\hat{u}) \leftarrow \mathrm{m}(\hat{v}) + \mathrm{n}(\hat{u}_1) + \cdots + \mathrm{n}(\hat{u}_k)$;
  end
  else
    // $\hat{v}$ *is a union node in* $T^c_{G_{\mathrm{SR}}}$
    // $\hat{v}$ *is a join node in* T^c_G
    $\mathrm{m}(\hat{u}) \leftarrow \mathrm{m}(\hat{v}) + \mathrm{vc}(\hat{u}_1) + \cdots + \mathrm{vc}(\hat{u}_k)$;
  end
end

After the pre-processing by Algorithm 3, the size of each

$$\overline{\mathrm{VC}}(\, G_{\mathrm{SR}},\, \mathrm{MD}(\, G,\, u\,)\,), \quad u \in V(G),$$

is computable in time $\mathcal{O}(1)$ as follows. If $\hat{v}$ is the predecessor node of $\hat{u} = \mathrm{node}(u)$ in T^c_G, then the size of $\overline{\mathrm{VC}}(\, G_{\mathrm{SR}},\, \mathrm{MD}(\, G,\, u\,)\,)$ is $\mathrm{n}(\hat{u}) - 1 + \mathrm{m}(\hat{u})$, see Figure 12 for an example. Note that we have to use the predecessor node $\hat{u} = \mathrm{node}(u)$ in the canonical decomposition tree T^c_G and not the canonical decomposition tree $T^c_{G_{\mathrm{SR}}}$.

Again, by a redefinition of $\mathrm{n}(\hat{u})$ to be the number of leaves in $T(\hat{u})$ that are not in a vertex set $W \subseteq V(G)$, the size of a vertex set

$$\overline{\mathrm{VC}}(\, G_{\mathrm{SR}} \setminus W,\, \mathrm{MD}(\, G,\, u\,) \setminus W\,)$$

is computable in constant time for each vertex $u \in V(G)$ as follows, see also Figure 13 for an example. Two cases must be distinguished. If vertex $u \notin W$, then the calculation of the size is as before $\mathrm{n}(\hat{u}) - 1 + \mathrm{m}(\hat{u})$. However, if $u \in W$, the computation is $\mathrm{n}(\hat{u}) + \mathrm{m}(\hat{u})$. Both cases can be covered by

$$\mathrm{n}(\hat{u}) - \mathrm{n}(\hat{w}) + \mathrm{m}(\hat{u}).$$

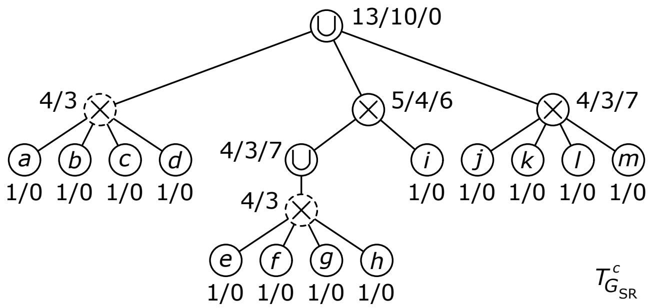


Figure 12: A canonical co-tree $T^c_{G_{\mathrm{SR}}}$ for a co-graph G_{SR}. The inner nodes $\hat{u}$ which also exist in T^c_G are labelled $\mathrm{n}(\hat{u})/\mathrm{vc}(\hat{u})/\mathrm{m}(\hat{u})$ by Algorithm 3. The leaves and the twin-join nodes are labelled only by $\mathrm{n}(\hat{u})/\mathrm{vc}(\hat{u})$. For example,

$$\begin{array}{lclcl}
\overline{\mathrm{VC}}(\,G_{\mathrm{SR}},\,\mathrm{MD}(\,G,\,a\,)\,) & = & 13-1+0 & = & 12, \\
\overline{\mathrm{VC}}(\,G_{\mathrm{SR}},\,\mathrm{MD}(\,G,\,e\,)\,) & = & 4-1+7 & = & 10, \text{ and} \\
\overline{\mathrm{VC}}(\,G_{\mathrm{SR}},\,\mathrm{MD}(\,G,\,i\,)\,) & = & 5-1+6 & = & 10.
\end{array}$$

Note that, the decomposition tree T^c_G does not contain the twin-join nodes. That is the predecessor node of a is the root of the tree.

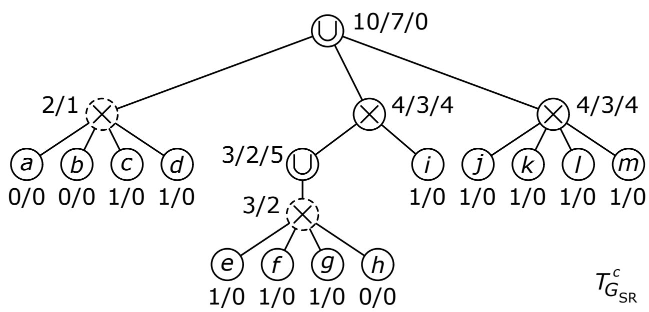


Figure 13: A canonical co-tree $T^c_{G_{\mathrm{SR}}}$ for a co-graph G_{SR}. The inner nodes $\hat{u}$ which also exist in T^c_G are labelled $\mathrm{n}(\hat{u})/\mathrm{vc}(\hat{u})/\mathrm{m}(\hat{u})$ by Algorithm 4. The leaves and the twin-join nodes are labelled only by $\mathrm{n}(\hat{u})/\mathrm{vc}(\hat{u})$. Here, the vertices a, b, and h are left out by setting $\mathrm{n}(a)$, $\mathrm{n}(b)$, and $\mathrm{n}(h)$ to zero. For example,

$$\begin{array}{lclcl}
\overline{\mathrm{VC}}(\,G_{\mathrm{SR}} \setminus \{a,b,h\},\,\mathrm{MD}(\,G,\,a\,) \setminus \{a,b,h\}\,) & = & 10-0+0 & = & 10, \\
\overline{\mathrm{VC}}(\,G_{\mathrm{SR}} \setminus \{a,b,h\},\,\mathrm{MD}(\,G,\,e\,) \setminus \{a,b,h\}\,) & = & 3-1+5 & = & 7, \text{ and} \\
\overline{\mathrm{VC}}(\,G_{\mathrm{SR}} \setminus \{a,b,h\},\,\mathrm{MD}(\,G,\,h\,) \setminus \{a,b,h\}\,) & = & 3-0+5 & = & 8.
\end{array}$$

Theorem 5. *A minimum strong resolving set for a graph $G = (V, E)$, in that each biconnected component is a co-graph, can be computed in time $\mathcal{O}(|V| + |E|)$.*

Ollermann and Peters-Fransen showed in [17] that the size of a strong resolving set can be computed in polynomial time for distance hereditary graphs. Graphs in which the biconnected components are co-graphs are distance hereditary, but our solution presented here runs in linear time.

8. Conclusion

In this paper we have shown that the efficient computation of a strong resolving set for a graph G essentially depends on the efficient computation of strong resolving sets for its biconnected components. If a minimum strong resolving set can be computed for a biconnected graph in polynomial time, it is often also possible to compute minimum strong resolving sets in polynomial time, which additionally contain the vertices that are maximally distant from other vertices.

We have given three examples for which it is possible to compute the required assumptions in linear time. From this it could be concluded that the computation of minimum strong resolving sets for graphs is possible in linear time if the biconnected components are grids, cycles, or co-graphs. It would be interesting to know for which other more complex graph classes this concept is applicable.

A generalization of the procedure for directed graphs and directed strong resolving sets, see for example [28], as well as a generalization of the composition of two graphs over several vertices that are all connected to each other are also interesting challenges.

References

[1] A. Sebö, E. Tannier, On metric generators of graphs, Mathematics of Operations Research 29 (2) (2004) 383–393. doi:10.1287/moor.1030.0070.
URL `https://doi.org/10.1287/moor.1030.0070`

[2] P. J. Slater, Leaves of trees, Congressum Numerantium 14 (1975) 549–559.

[3] P. J. Slater, Dominating and reference sets in a graph, Journal of Mathematical and Physical Sciences 22 (1988) 445 – 455.

[4] F. Harary, R. A. Melter, On the metric dimension of a graph, Ars Combinatoria 2 (1976) 191–195.

[5] S. Khuller, B. Raghavachari, A. Rosenfeld, Landmarks in graphs, Discrete Applied Mathematics 70 (1996) 217–229.

[6] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, 1979.

[7] J. Díaz, O. Pottonen, M. J. Serna, E. J. van Leeuwen, Complexity of metric dimension on planar graphs, J. Comput. Syst. Sci. 83 (1) (2017) 132–158. doi:10.1016/j.jcss.2016.06.006.
URL `https://doi.org/10.1016/j.jcss.2016.06.006`

[8] S. Hoffmann, E. Wanke, Metric dimension for gabriel unit disk graphs is NP-complete, in: A. Bar-Noy, M. M. Halldórsson (Eds.), ALGOSENSORS, Vol. 7718 of Lecture Notes in Computer Science, Springer, 2012, pp. 90–92. doi:10.1007/978-3-642-36092-3.
URL `https://doi.org/10.1007/978-3-642-36092-3`

[9] G. Chartrand, L. Eroh, M. Johnson, O. Oellermann, Resolvability in graphs and the metric dimension of a graph, Discrete Applied Mathematics 105 (1-3) (2000) 99–113.

[10] M. Hernando, M. Mora, I. Pelayo, C. Seara, J. Cáceres, M. Puertas, On the metric dimension of some families of graphs, Electronic Notes in Discrete Mathematics 22 (2005) 129–133.

[11] R. Melter, I. Tomescu, Metric bases in digital geometry, Computer Vision, Graphics, and Image Processing 25 (1) (1984) 113–121.

[12] S. Saputro, E. Baskoro, A. Salman, D. Suprijanto, A. Baca, The metric dimension of regular bipartite graphs, arXiv/1101.3624 (2011). arXiv:arXiv/1101.3624.
URL `http://arxiv.org/abs/1101.3624`

[13] H. Iswadi, E. Baskoro, A. Salman, R. Simanjuntak, The metric dimension of amalgamation of cycles, Far East Journal of Mathematical Sciences (FJMS) 41 (1) (2010) 19–31.

[14] S. Hoffmann, A. Elterman, E. Wanke, A linear time algorithm for metric dimension of cactus block graphs, Theoretical Computer Science 630 (2016) 43–62.

[15] H. Fernau, P. Heggernes, P. van't Hof, D. Meister, R. Saei, Computing the metric dimension for chain graphs, Information Processing Letters 115 (9) (2015) 671–676.

[16] D. Vietz, S. Hoffmann, E. Wanke, Computing the metric dimension by decomposing graphs into extended biconnected components - (extended abstract), in: G. K. Das, P. S. Mandal, K. Mukhopadhyaya, S. Nakano (Eds.), WALCOM: Algorithms and Computation - 13th International Conference, WALCOM 2019, Guwahati, India, February 27 - March 2, 2019, Proceedings, Vol. 11355 of Lecture Notes in Computer Science, Springer, 2019, pp. 175–187.
URL `https://doi.org/10.1007/978-3-030-10564-8_14`

[17] O. R. Oellermann, J. Peters-Fransen, The strong metric dimension of graphs and digraphs, Discrete Applied Mathematics 155 (3) (2007) 356–364. doi:10.1016/j.dam.2006.06.009.
URL `https://doi.org/10.1016/j.dam.2006.06.009`

[18] J.-B. Liu, A. Zafari, H. Zarei, Metric dimension, minimal doubly resolving sets, and the strong metric dimension for jellyfish graph and cocktail party graph, Complexity 2020 (2020). doi:https://doi.org/10.1155/2020/9407456.

[19] D. Kuziak, I. G. Yero, J. A. Rodríguez-Velázquez, Strong metric dimension of rooted product graphs, International Journal of Computer Mathematics 93 (8) (2016) 1265–1280. doi:10.1080/00207160.2015.1061656.

[20] P. Manuel, B. Rajan, C. Grigorious, S. Stephen, On the strong metric dimension of tetrahedral diamond lattice, Mathematics in Computer Science 9 (2) (2015) 201–208. doi:10.1007/s11786-015-0226-0.

[21] D. Kuziak, The strong resolving graph and the strong metric dimension of cactus graphs, Mathematics 8 (8) (2020) 1266. doi:http://doi.org/10.3390/math8081266.

[22] R. Farooq, N. Mehreen, Strong metric dimension of generalized jahangir graph (2019). arXiv:1905.03975.

[23] M. Widyaningrum, T. A. Kusmayadi, On the strong metric dimension of sun graph, windmill graph, and möbius ladder graph, Journal of Physics: Conference Series 1008 (2018) 12–32. doi:10.1088/1742-6596/1008/1/012032.
URL `https://doi.org/10.1088%2F1742-6596%2F1008%2F1%2F012032`

[24] D. Kuziak, I. G. Yero, J. A. Rodríguez-Velázquez, On the strong metric dimension of corona product graphs and join graphs, Discrete Applied Mathematics 161 (7-8) (2013) 1022–1027. doi:10.1016/j.dam.2012.10.009.
URL `https://doi.org/10.1016/j.dam.2012.10.009`

[25] J. A. Rodríguez-Velázquez, I. G. Yero, D. Kuziak, O. R. Oellermann, On the strong metric dimension of cartesian and direct products of graphs, Discrete Mathematics 335 (2014) 8 – 19. doi:https://doi.org/10.1016/j.disc.2014.06.023.
URL `http://www.sciencedirect.com/science/article/pii/S0012365X14002507`

[26] D. G. Corneil, H. Lerchs, L. S. Burlingham, Complement reducible graphs, Discrete Applied Mathematics 3 (3) (1981) 163–174.

[27] B. Jamison, S. Olariu, A linear-time recognition algorithm for p4-reducible graphs, Theor. Comput. Sci. 145 (1&2) (1995) 329–344. doi:10.1016/0304-3975(95)00016-P.
URL `https://doi.org/10.1016/0304-3975(95)00016-P`

[28] Y. Schmitz, E. Wanke, On the strong metric dimension of directed co-graphs, CoRR abs/2111.13054 (2021). arXiv:2111.13054.
URL `https://arxiv.org/abs/2111.13054`

[29] M. Habib, C. Paul, A simple linear time algorithm for cograph recognition, Discrete Applied Mathematics 145 (2) (2005) 183–197.

9.4 On the Strong Metric Dimension of directed co-graphs

On the Strong Metric Dimension of directed co-graphs
Yannick Schmitz, Egon Wanke
Eingereicht in: Theoretical Computer Science

Diese Arbeit enthält Algorithmen zur Berechnung der Starken Metrischen Dimension ungerichteter Co-Graphen und der gerichteten Starken Metrischen Dimension gerichteter Co-Graphen in linearer Zeit. Die Ergebnisse sind in Kapitel 6 und Kapitel 7 zu finden. Alle Autoren haben gleichermaßen zur Erarbeitung und zur Formulierung der Ergebnisse beigetragen. Den Hauptteil meines Beitrags bildet die Bestimmung der strong resolving Graphen und der speziellen Arten von Knoten.

On the Strong Metric Dimension of directed co-graphs

Yannick Schmitz[a,*], Egon Wanke[a]

[a] *Heinrich-Heine-Universität Düsseldorf, Universitätsstraße 1, Düsseldorf, 40225, Germany*

Abstract

Let G be a strongly connected directed graph and $u, v, w \in V(G)$ be three vertices. Then w *strongly resolves* u to v if there is a shortest u-w-path containing v or a shortest w-v-path containing u. A set $R \subseteq V(G)$ of vertices is a *strong resolving set* for a directed graph G if for every pair of vertices $u, v \in V(G)$ there is at least one vertex in R that strongly resolves u to v and at least one vertex in R that strongly resolves v to u. The distances of the vertices of G to and from the vertices of a strong resolving set R uniquely define the connectivity structure of the graph. The *Strong Metric Dimension* of a directed graph G is the size of a smallest strong resolving set for G. The decision problem Strong Metric Dimension is the question whether G has a strong resolving set of size at most r, for a given directed graph G and a given number r. In this paper we study undirected and directed co-graphs and introduce linear time algorithms for Strong Metric Dimension. These algorithms can also compute strong resolving sets for (un)directed co-graphs in linear time.

Keywords: directed strong metric dimension, directed co-graphs, strong metric dimension, co-graphs

1. Introduction

The strong metric dimension is a modified version of the metric dimension. Both terms were first defined and analysed for undirected graphs. In order to embed the results worked out in this paper in the literature, we first need the precise definitions of the metric dimension for undirected and directed graphs in the original and strong version.

A vertex w of an undirected graph G *resolves* two vertices u and v if the distance between w and u is different from the distance between w and v, that is, if $d(w, u) \neq d(w, v)$. A set of vertices $R \subseteq V(G)$ is called a *resolving set* for G if every pair of vertices of G is resolved by at least one vertex of R. The *metric dimension* of G, denoted by $\mathrm{md}(G)$, is the size of a smallest resolving set, which is also called a *metric basis* for G.

The *strong metric dimension* differs from the usual metric dimension only in the concept of resolving two vertices. A vertex w *strongly resolves* u and v if there is either an undirected shortest path between w and u in G containing v or an undirected shortest path between w and v in G containing u. A set of vertices $R \subseteq V(G)$ is called a *strong resolving set* for

*Corresponding author. E-mail: yannick.schmitz@hhu.de

G if every pair of vertices of G is strongly resolved by at least one vertex of R. The *strong metric dimension* of G, denoted by $\mathrm{smd}(G)$, is the size of a smallest strong resolving set, which is also called a *strong metric basis* for G.

Both models can be extended to directed graphs. This leads to the definition of the directed metric dimension and the strong directed metric dimension. In the directed metric dimension, the vertices u and v are resolved by vertex w if the length of a shortest path from w to u and the length of a shortest path from w to v are different. Some paper consider the length of a shortest path from u to w and the length of a shortest path from v to w, which is equivalent if every edge is replaced by an edge with the opposite direction.

For the definition of the strong metric dimension for directed graphs, there are more case distinctions. Let us denote a directed shortest path from a vertex x to a vertex z that contains a vertex y as a shortest $x \to y \to z$-path. Two vertices u and v are strongly resolved from u to v by a vertex w if there is a shortest $w \to u \to v$-path or a shortest $u \to v \to w$-path. The vertices u and v are strongly resolved if they are strongly resolved from u to v and strongly resolved from v to u. Analogously to the undirected case, a set of vertices $R \subseteq V(G)$ is called a *strong resolving set* for G if every pair of vertices of G is strongly resolved by vertices of R. Again, the *strong metric dimension* of G is the size of a smallest strong resolving set for G.

Due to the similarity of both definitions for the directed and undirected version, the decision problem Strong Metric Dimension can be defined as follows for both cases:

	(Directed) Strong Metric Dimension
Instance:	A (directed or undirected) graph G and a number r
Question:	Is there a strong resolving set $R \subseteq V(G)$ for G of size at most r?

The metric dimension of undirected graphs has been introduced in the 1970s independently by Slater [1] and by Harary and Melter [2]. The metric dimension finds applications in various areas, including network discovery and verification [3], geographical routing protocols [4], combinatorial optimization [5], sensor networks [6], robot navigation [7] and chemistry [8, 9]. The strong metric dimension was introduced by Sebö and Tannier [5] who also showed, that a strong metric basis uniquely defines a graph. As both the metric and the strong metric dimension are NP-complete for general graphs [10, 11], special classes of graphs for which those problems can be solved efficiently are of interest. The directed versions of both problems are NP-complete too, as they are equivalent to their undirected version if each undirected edge is replaced by two directed edges.

There are several algorithms for computing a minimum resolving set in polynomial time for certain classes, for example trees [8, 7], wheels [12], grid graphs [13], k-regular bipartite graphs [14], amalgamation of cycles [15], cactus block graphs [16], graphs with size-constrained biconnected components [17] and outerplanar graphs [18]. The approximability of the metric dimension has been studied for bounded degree, dense and general graphs in [19]. Upper and lower bounds on the metric dimension are considered in [20, 21] for further classes of graphs.

The strong metric dimension is mainly studied on different products of graphs [22, 23, 24, 25, 26, 27], but there are also efficient algorithms for other classes of graphs, for example trees [5], distance hereditary graphs [28] and split graphs [29]. However, the strong metric dimension has not been extensively studied for directed graphs. It was first defined by Sebö and Tannier in [5] in order to uniquely define the edges in a graph using the directed shortest path lengths to the vertices of a strong resolving set of vertices.

In this paper we present an algorithm that computes the strong metric dimension for undirected and directed co-graphs in linear time. This is one of the first approaches for the computation of the strong metric dimension of special directed graph classes. In Section 2 we first describe a linear time algorithm for the computation of the strong metric dimension for undirected co-graphs. This also illustrates the algorithmic idea for the linear-time algorithm presented in Section 3 for the computation of the strong metric dimension for directed co-graphs.

1.1. Preliminaries

All (directed) graphs in this paper are finite, simple and if not stated otherwise, (strongly) connected, respectively. For a (directed) graph $G = (V, E)$ with vertex set V and edge set E, we write $V(G)$ for vertex set V and $E(G)$ for edge set E to reduce the number of variable names when using several graphs.

For a (strongly) connected graph G and two vertices $u, v \in V(G)$ the distance $d_G(u, v)$ from u to v in G is the length (number of edges) of a shortest path from u to v in G. If it is clear from the context which graph is meant, we also write $d(u, v)$ instead of $d_G(u, v)$.

1.1.1. Undirected Graphs

The *diameter* of G is the maximum distance between two vertices u and v of G.

Graph G' is a *subgraph* of G, denoted by $G' \subseteq G$, if $V(G') \subseteq V(G)$ and $E(G') \subseteq \{\{u, v\} \mid u, v \in V(G'),\ \{u, v\} \in E(G)\}$. Subgraph G' is an *induced subgraph* of G, if $E(G') = \{\{u, v\} \mid u, v \in V(G'),\ \{u, v\} \in E(G)\}$. We denote the subgraph of G induced by the vertices of $V(G')$ by $G[V(G')]$ or $G|_{V(G')}$. For a vertex $u \in V(G)$ or a vertex set $U \subseteq V(G)$, let $G \setminus u = G[V(G) \setminus \{u\}]$ and $G \setminus U = G[V(G) \setminus U]$ be the subgraphs of G induced by $V(G) \setminus \{u\}$ and $V(G) \setminus U$, respectively.

For a vertex $u \in V(G)$, the *open neighbourhood* of u is defined by $N_G(u) = \{v \mid v \in V(G),\ \{v, u\} \in E(G)\}$ and the *closed neighbourhood* of u is defined by $N_G[u] = N_G(u) \cup \{u\}$. If it is clear from the context which graph is meant, we also write $N(u)$ and $N[u]$ instead of $N_G(u)$ and $N_G[u]$.

Two vertices $u, v \in V(G)$ are called *false twins* if $N(u) = N(v)$ and two vertices $u, v \in V(G)$ are called *true twins* if $N[u] = N[v]$. We call two vertices $u, v \in V(G)$ twins, if they are either false twins or true twins.

The complement graph $\overline{G}$ of G is defined by $V(\overline{G}) = V(G)$ and $E(\overline{G}) = \{\{u, v\} \mid u, v \in V(G),\ u \neq v,\ \{u, v\} \notin E(G)\}$.

The size of a minimum vertex cover, maximum independent set and maximum clique of an undirected graph G is denoted by $\tau(G)$, $\alpha(G)$ and $\omega(G)$, respectively.

1.1.2. Directed Graphs

The *diameter* of G is the maximal distance from a vertex u to a vertex v in G.

Graph G' is a *subgraph* of G, denoted by $G' \subseteq G$, if $V(G') \subseteq V(G)$ and $E(G') \subseteq \{(u,v) \mid u,v \in V(G'),\ (u,v) \in E(G)\}$. Subgraph G' is an *induced subgraph* of G, if $E(G') = \{(u,v) \mid u,v \in V(G'),\ (u,v) \in E(G)\}$. The notation for an induced subgraph is the same as for undirected graphs.

For a vertex $u \in V(G)$, we denote by $N^-(u) = \{v \mid v \in V(G),\ (v,u) \in E(G)\}$ the set of *in-neighbours* of u and by $N^+(u) = \{v \mid v \in V(G),\ (u,v) \in E(G)\}$ the set of *out-neighbours* of u.

For a vertex $u \in V(G)$ or a vertex set $U \subseteq V(G)$, let $G \setminus u = G[V(G) \setminus \{u\}]$ and $G \setminus U = G[V(G) \setminus U]$ be the subgraphs of G induced by $V(G) \setminus \{u\}$ and $V(G) \setminus U$, respectively.

2. The strong metric dimension of undirected co-graphs

In this section, we provide a linear time algorithm to compute the strong metric dimension of undirected co-graphs. May and Oellerman presented in [28] an algorithm that computes the strong metric dimension for distance hereditary graphs in polynomial time. Since co-graphs are distance hereditary, their algorithm computes the strong metric dimension also for co-graphs in polynomial time. However, the complicated data structures used by the algorithm of May and Oellerman are not necessary to compute the strong metric dimension for co-graphs, as our simple algorithm shows. We introduce a different approach by analysing the co-tree, which we will later use to compute the strong metric dimension of directed co-graphs.

Definition 1 (Undirected Co-Graphs). *[30]*

- *An undirected graph G that consists of a single vertex v is an undirected co-graph. The co-tree T of G consists of a single node u associated with vertex v of G. Node u is the* root *of T.*

- *If G_1 and G_2 are two undirected co-graphs, then the* union *of G_1 and G_2, denoted by $G_1 \cup G_2$, is an undirected co-graph G with vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2)$. Let T_1 and T_2 be the co-trees of G_1 and G_2 with root u_1 and u_2, respectively. The co-tree T of G is the disjoint union of T_1 and T_2 with an additional node u and two additional edges $\{u,u_1\}$ and $\{u,u_2\}$. Node u is the* root *of T labelled by $\cup$. Node u_1 and u_2 are* successor *nodes of u.*

- *If G_1 and G_2 are two undirected co-graphs, then the* join *of G_1 and G_2, denoted by $G_1 \times G_2$, is an undirected co-graph with vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2) \cup \{\{u,v\} \mid u \in V(G_1), v \in V(G_2)\}$. Let T_1 and T_2 be the co-trees of G_1 and G_2 with root u_1 and u_2, respectively. The co-tree T of G is the disjoint union of T_1 and T_2 with an additional node u and two additional edges $\{u,u_1\}$ and $\{u,u_2\}$. Node u is the* root *of T labelled by $\times$. Node u_1 and u_2 are* successor *nodes of u.*

Undirected co-graphs are precisely those graphs that do not contain an induced P_4, i.e. a path with four vertices, thus their diameter is at most 2.

Definition 2. *[11] Let G be an undirected graph.*

1. *A vertex $u \in V(G)$ is* maximally distant *to a vertex $v \in V(G)$ in G if $\forall u' \in N(u)$: $d_G(u', v) \leq d_G(u, v)$.*
2. *Two vertices $u, v \in V(G)$ are* mutually maximally distant *if u is maximally distant to v and v is maximally distant to u.*
3. *The* strong resolving graph *of G, denoted by G_{SR}, is an undirected graph with the same vertex set as G and an edge between two vertices $u, v \in V(G)$ if an only if u and v are mutually maximally distant in G.*

Theorem 1. *[11] Let G be an undirected graph, then* $\mathrm{smd}(G) = \tau(G_{\mathrm{SR}})$*, where $\tau(G_{\mathrm{SR}})$ is the size of a minimum vertex cover of G_{SR}.*

Instead of looking for a minimum vertex cover of G_{SR}, we can also look for a maximum independent set of G_{SR} or equivalently, for a maximum clique of $\overline{G_{\mathrm{SR}}}$. That is, we can calculate the strong metric dimension of G by

$$\tau(G_{\mathrm{SR}}) \;=\; n - \alpha(G_{\mathrm{SR}}) \;=\; n - \omega(\overline{G_{\mathrm{SR}}}),$$

where n is the number of vertices of G_{SR}, $\alpha(G_{\mathrm{SR}})$ is the size of a maximum independent set of G_{SR} and $\omega(\overline{G_{\mathrm{SR}}})$ is the size of a maximum clique of $\overline{G_{\mathrm{SR}}}$.

Lemma 1. *Let G be an undirected graph.*

1. *If two vertices $u, v \in V(G)$ of G are true (false) twins in G, then u and v are false (true) twins in $\overline{G}$.*
2. *If two vertices $u, v \in V(G)$ of G are twins in G, then u and v are true twins in G_{SR}.*
3. *If $\{u, v\}$ is an edge of G, then $\{u, v\}$ is an edge in G_{SR} if and only if u and v are true twins in G.*

Proof.

1. Obviously.
2. Let $w \in V(G) \setminus \{u, v\}$. If u and v are twins, then vertex w and vertex u are mutually maximally distant if and only if vertex w and vertex v are mutually maximally distant. That is, u and v are twins in G_{SR}. Since twins are mutually maximally distant from each other, they are true twins in G_{SR}.
3. Let $\{u, v\} \in E(G)$ be an edge of G.
   " $\Rightarrow$ ": From the definition of "mutually maximally distant" it follows that for every vertex $u' \in N(u)$, $d_G(u', v) \leq d_G(u, v)$ and for every vertex $v' \in N(v)$, $d_G(u, v') \leq d_G(u, v)$. Since $d_G(u, v) = 1$, every neighbour of u is either v or a neighbour of v and every neighbour of v is either u or a neighbour of u. Therefore, u and v are true twins in G.

" $\Leftarrow$ ": If u and v are true twins in G, every neighbour of u is either v or a neighbour of v and every neighbour of v is either u or a neighbour of u. Therefore, for every vertex $u' \in N(u)$ $d_G(u', v) \leq d_G(u, v)$ and for every vertex $v' \in N(v)$ $d_G(v', u) \leq d_G(v, u)$ and thus u and v are mutually maximally distant in G.

□

Lemma 2. *If G is an undirected graph of diameter at most 2, then*

$$E(G_{\mathrm{SR}}) = E(\overline{G}) \cup \{\{u, v\} \mid u, v \text{ are true twins in } G\}.$$

Proof. Since the diameter of G is ≤ 2, for every distinct pair of vertices $u, v \in V(G)$ either $d_G(u, v) = 1$ or $d_G(u, v) = 2$. If two vertices $u, v \in V(G)$ are mutually maximally distant in G, that is if $\{u, v\} \in E(G_{\mathrm{SR}})$, then either

1. $d_G(u, v) = 2$ and $\{u, v\} \in E(\overline{G})$ or
2. $d_G(u, v) = 1$ and $\{u, v\} \notin E(\overline{G})$ and by Lemma 1.3, u and v are true twins.

If $\{u, v\} \in E(\overline{G})$ then $\{u, v\} \notin E(G)$ and $d_G(u, v) = 2$ and u, v are mutually maximally distant in G and thus $\{u, v\} \in E(G_{\mathrm{SR}})$. □

Since the complement $\overline{G}$ of a co-graph G is a co-graph by the definition of co-graphs and since true twins of G are false twins in $\overline{G}$ by Lemma 1.1 and since the insertion of edges between false twins with non-empty open neighbourhood does not increase the distances between two vertices in a graph, it follows that G_{SR} is a co-graph if G is a co-graph.

Corollary 1. *If G is an undirected co-graph, then G_{SR} is an undirected co-graph.*

Lemma 3. *Let G be an undirected graph and $u, v \in V(G)$ be two true twins of G, then* $\mathrm{smd}(G) = \mathrm{smd}(G \setminus v) + 1$.

Proof. Let u and v be two true twins of G and let $G' = G \setminus v$. By Lemma 1.2 u and v are true twins in G_{SR}. Two vertices $w_1, w_2 \in V(G) \setminus \{u, v\}$ are mutually maximally distant in G if and only if they are mutually maximally distant in G'. Since u and v are true twins, for every vertex $w \in V(G) \setminus \{u, v\}$, the vertices u and w are mutually maximally distant in G if and only of they are mutually maximally distant in G'. Note that this does not hold if u and v are false twins. This reasoning shows that $G'_{\mathrm{SR}} = G_{\mathrm{SR}} \setminus v$.

Every vertex cover U' of G'_{SR} can be converted into a vertex cover U of G_{SR} by adding vertex v to U', that is $\mathrm{smd}(G') + 1 \geq \mathrm{smd}(G)$. Every vertex cover U of G_{SR} contains vertex u or vertex v or both vertices, because u and v are adjacent in G_{SR}. If U does not contain vertex v, then replace vertex u in U by vertex v. The resulting set U is again a vertex cover of G_{SR} and $U \setminus \{v\}$ is a vertex cover for G'_{SR}, which implies $\mathrm{smd}(G) - 1 \geq \mathrm{smd}(G')$. □

Removing a twin in an undirected graph G does not destroy the twin property of the remaining pairs of vertices. If $u_1, u_2 \in V(G)$ are twins in G and $v_1, v_2 \in V(G)$ are also twins in G, then u_1, u_2 remain twins in $G \setminus v_1$ and $G \setminus v_2$. This property is the prerequisite for the following theorem.

Theorem 2. *Let G be an undirected co-graph with n vertices. Let G' be the graph obtained by successively removing true twins from G, then*

$$\mathrm{smd}(G) = n - \omega(G'),$$

where $\omega(G')$ is the size of a maximum clique in G', compare also [31].

Proof. Let $n = |V(G)|$ and and $n' = |V(G')|$ then

$$\begin{array}{rcll}
\mathrm{smd}(G) & = & \mathrm{smd}(G') + n - n' & \text{by repeatedly applying Lemma 3} \\
 & = & \tau(G'_{\mathrm{SR}}) + n - n' & \text{by Theorem 1} \\
 & = & \tau(\overline{G'}) + n - n' & \text{by Lemma 2, because } G' \text{ has no true twins} \\
 & = & n' - \alpha(\overline{G'}) + n - n' & \text{because } \tau(\overline{G'}) = n' - \alpha(\overline{G'}) \\
 & = & n - \alpha(\overline{G'}) & \\
 & = & n - \omega(G') & \text{because } \alpha(\overline{G'}) = \omega(G'),
\end{array}$$

where $\tau(G)$ and $\alpha(G)$ is the size of a minimum vertex cover and maximum independent set of a graph G, respectively. □

The strong metric dimension of an undirected co-graph G can be computed in linear time by computing the size of a maximum clique in an undirected co-graph G, in which all true twins were removed beforehand. The computation of the size of a largest clique in G can be carried out in linear time with a bottom-up processing of the co-tree T of G. A leaf gets the value 1, an inner node u labelled $\cup$ gets the maximum of the values of the two successor nodes of u, and an inner node u labelled by $\times$ gets the sum of the values of the two successor nodes of u. The value of the root of T is the size of a maximum clique in G.

The removal of true twins from an undirected co-graph can be carried out together with the bottom-up processing for the calculation of the size of a maximum clique. A simplest way to do this, is to use the so-called *canonical co-tree* in which successive union and successive join compositions are combined to one union and one join composition, respectively. Then each path in T from the root to a leaf is an alternating sequence of union and join compositions. The canonical co-tree T is not necessarily a binary tree but can also be computed in linear time [32].

It is easy to see, that two vertices in an undirected co-graph G are true twins if and only if the two corresponding leaves in the canonical co-tree are successor nodes of the same join node. In order to compute the size of a maximum clique in G without true twins, it is sufficient to count all successor nodes of a join node together as one unit only once. This is taken into account in the following Algorithm 1 using a variable t, which is set to a maximum of 1.

Theorem 3. *The strong metric dimension of undirected co-graphs can be computed in linear time.*

Algorithm 1 Strong Metric Dimension of Undirected Co-graphs

```
function SMD(co-graph G)
    T ← canonical co-tree of G
    n ← |V(G)|
    return n−MAX_TWINLESS_CLIQUE(T)
end

function MAX_TWINLESS_CLIQUE(canonical co-tree T)
    w ← root of T
    if w is labelled by ∪ then
        m ← 0
        for all subtrees T' at root w do
            k =MAX_TWINLESS_CLIQUE(T')
            if k > m then
                m ← k
        return m
    else if w is labelled by × then
        s ← 0
        t ← 0
        for all subtrees T' at root w do
            if root of T' is a leaf then
                t ← 1
            else
                s ← s+MAX_TWINLESS_CLIQUE(T')
        return s + t
    else
        return 1
end
```

The above idea for the efficient computation of the strong metric dimension can of course also be used for all other graph classes that have a diameter of at most two, are closed with respect to the removal of twins and for which a maximum clique can be efficiently found. These would be, for example, perfect graphs with a diameter of at most two or graphs of bounded clique-width and a diameter of at most two. Here is an example of such a conclusion.

Corollary 2. *The strong metric dimension of perfect graphs of diameter at most 2 can be computed in polynomial time.*

Proof. Analogously to Theorem 2, the strong metric dimension of perfect graphs of diameter at most 2 can be computed by successively removing true twins from the graph by using Lemma 3 and calculating the size of a maximum clique in the remaining graph. Since removing true twins does not create holes or anti holes, the remaining graph is still perfect and of diameter at most 2. Therefore, Lemma 2 still holds where a maximum clique of a prefect graph can be computed in polynomial time. □

3. The strong metric dimension of strongly connected directed co-graphs

In this section we show how to determine the strong metric dimension of strongly connected directed co-graphs in linear time. Directed co-graphs are recursively defined as follows, see also [33]:

Definition 3 (**Directed Co-graphs).**

- *A directed graph G that consists of a single vertex v is a directed co-graph. The co-tree T of G consists of a single node u associated with vertex v of G. Node u is the* root *of T.*

- *If G_1 and G_2 are two directed co-graphs, then the* union *of G_1 and G_2, denoted by $G_1 \cup G_2$, is a directed co-graph G with vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2)$. Let T_1 and T_2 be the co-trees of G_1 and G_2 with root u_1 and u_2, respectively. The co-tree T of G is the disjoint union of T_1 and T_2 with an additional node u and two additional edges $\{u, u_1\}$ and $\{u, u_2\}$. Node u is the* root *of T labelled by $\cup$. Node u_1 is the* left successor node *of u and node u_2 is the* right successor node *of u.*

- *If G_1 and G_2 are two directed co-graphs, then the* join *of G_1 and G_2, denoted by $G_1 \times G_2$, is a directed co-graph with vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2) \cup \{(u, v), (v, u) \mid u \in V(G_1), v \in V(G_2)\}$. Let T_1 and T_2 be the co-trees of G_1 and G_2 with root u_1 and u_2, respectively. The co-tree T of G is the disjoint union of T_1 and T_2 with an additional node u and two additional edges $\{u, u_1\}$ and $\{u, u_2\}$. Node u is the* root *of T labelled by $\times$. Node u_1 is the* left successor node *of u and node u_2 is the* right successor node *of u.*

- *If G_1 and G_2 are two directed co-graphs, then the* directed join *of G_1 and G_2, denoted by $G_1 \gg G_2$, is a directed co-graph with vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2) \cup \{(u, v) \mid u \in V(G_1), v \in V(G_2)\}$. Let T_1 and T_2 be the co-trees of G_1 and G_2 with root u_1 and u_2, respectively. The co-tree T of G is the disjoint union of T_1 and T_2 with an additional node u and two additional edges $\{u, u_1\}$ and $\{u, u_2\}$. Node u is the* root *of T labelled by $\gg$. Node u_1 is the* left successor node *of u and node u_2 is the* right successor node *of u.*

Definition 4. *[11] Let G be a strongly connected directed graph and $u, v \in V(G)$.*

- *Vertex u is maximally distant to vertex v, denoted by $u \,\mathrm{MDT}\, v$, if $\forall u' \in N^-(u) : d(u', v) \leq d(u, v)$.*

- *Vertex v is maximally distant from vertex u, denoted by $v \,\mathrm{MDF}\, u$, if $\forall v' \in N^+(v)$: $d(u, v') \leq d(u, v)$.*

- *Vertex u is mutually maximally distant to vertex v, denoted by $u \,\mathrm{MMDT}\, v$, if $u \,\mathrm{MDT}\, v$ and $v \,\mathrm{MDF}\, u$.*

- *The* strong resolving graph *of G, denoted by G_{SR}, is an undirected graph with the same vertex set as G and an edge between two vertices $u, v \in V(G)$ if and only if u is mutually maximally distant to vertex v, or v is mutually maximally distant to vertex u in G, i.e., if*
$$(u \,\mathrm{MMDT}\, v) \vee (v \,\mathrm{MMDT}\, u).$$

Note that $u \,\mathrm{MMDT}\, v$ does not imply $v \,\mathrm{MMDT}\, u$, see Figure 1.

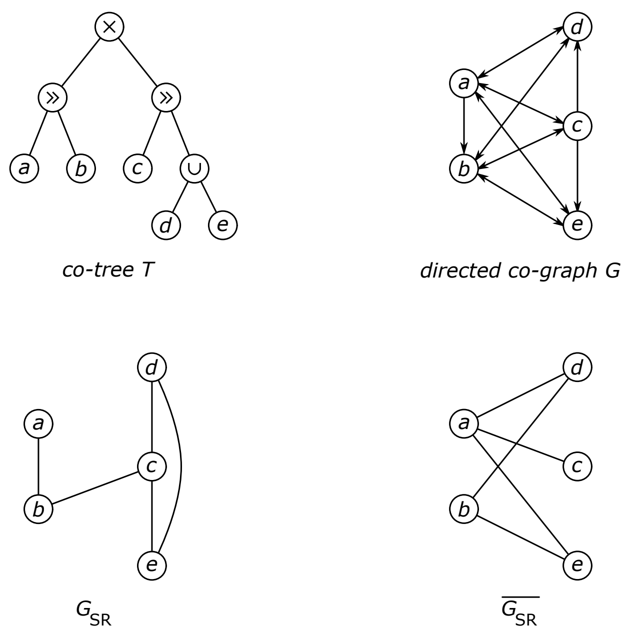


Figure 1: An example of a co-tree T (top left) of a directed co-graph G (top right), G_{SR} (bottom left) and $\overline{G_{\mathrm{SR}}}$ (bottom right). In this graph, the vertex a is not maximally distant to vertex c, because d and e are in-neighbours of a, but not in-neighbours of c, thus $e \in N^-(a)$ and $d(e, c) > d(a, c)$. However, the vertex c is maximally distant from vertex a, because every out-neighbour of c is an out-neighbour of a as well. In this example, we also have $c \,\mathrm{MMDT}\, b$, but not $b \,\mathrm{MMDT}\, c$. In this example, $\omega(\overline{G_{\mathrm{SR}}}) = 2 = \alpha(G_{\mathrm{SR}})$ and $|V_G| - \alpha(G_{\mathrm{SR}}) = \tau(G_{\mathrm{SR}}) = 3 = \mathrm{smd}(G)$.

Theorem 4. *[11] Let G be a strongly connected directed graph, then* $\mathrm{smd}(G) = \tau(G_{\mathrm{SR}})$.

We will now compute the strong metric dimension of directed co-graphs G based on a similar idea as we used in Section 2. Since we only consider strongly connected directed co-graphs, the last composition step is always a join operation if G has more than one vertex. Instead of finding a vertex cover in G_{SR}, we determine a clique in $\overline{G_{\mathrm{SR}}}$ as in Theorem 2.

For the next two lemmas and the explanations of the algorithm for the computation of a strong metric basis, let G be a strongly connected directed co-graph and T be the co-tree of G. Let w be a node of T and T_w be the complete subtree of T with root w. Let T_l and T_r be the left and right subtree of T_w at root w. Let G_w, G_l and G_r be the subgraphs of G induced by the vertices associated to the leaves of T_w, T_l and T_r, respectively.

Lemma 4. *If node w is labelled by $\cup$ or $\gg$ then no vertex of G_l is adjacent with a vertex of G_r in $\overline{G_{\mathrm{SR}}}$.*

Proof. For every pair of vertices $u, v \in V(G)$ in a strongly connected directed co-graph G, we have $d_G(u,v) \leq 2$. That is, if $d_G(u,v) = 2$, then $u \,\mathrm{MMDT}\, v$. If w is labelled by $\cup$, then $d_G(u,v) = 2$ and $d_G(v,u) = 2$ and thus $u \,\mathrm{MMDT}\, v$ and $v \,\mathrm{MMDT}\, u$. If w is labelled by $\gg$, then obviously $d_G(v,u) = 2$ and $v \,\mathrm{MMDT}\, u$. Although the last statement is sufficient for the proof, note that the directed join composition which creates the edge (u,v), also creates an edge from every in-neighbour of u to v and from u to very out-neighbour of v. That is, in G_w and thus also in G there is no shortest path from an in-neighbour of u via u to v and no shortest path from u via v to an out-neighbour of v, which implies $u \,\mathrm{MMDT}\, v$ in G. $\square$

Lemma 4 shows that G_{SR} contains all edges between the vertices of G_l and the vertices of G_r or equivalently, there is no edge between a vertex of G_l and a vertex of G_r in $\overline{G_{\mathrm{SR}}}$.

Consider now the cases in which w is labelled by $\times$. By definition, $\overline{G_{\mathrm{SR}}}$ has an edge between u and v if and only if

$$\neg((u \,\mathrm{MMDT}\, v) \vee (v \,\mathrm{MMDT}\, u))$$

or equivalently,

$$\neg(u \,\mathrm{MMDT}\, v) \wedge \neg(v \,\mathrm{MMDT}\, u)$$

or equivalently,

$$(\neg(u \,\mathrm{MDT}\, v) \vee \neg(v \,\mathrm{MDF}\, u)) \wedge (\neg(v \,\mathrm{MDT}\, u) \vee \neg(u \,\mathrm{MDF}\, v))$$

or equivalently by Definition 4,

1. (a) $\neg(\forall y \in N^-(u) : d(y,v) \leq d(u,v))$ or
   (b) $\neg(\forall x \in N^+(v) : d(u,x) \leq d(u,v))$
   and
2. (a) $\neg(\forall x \in N^-(v) : d(x,u) \leq d(v,u))$ or
   (b) $\neg(\forall y \in N^+(u) : d(v,y) \leq d(v,u))$,

or equivalently because we consider a join operation,

1. (a) $\exists y \in V(G)$ such that $(y,u) \in E(G)$ and $(y,v) \notin E(G)$ or
   (b) $\exists x \in V(G)$ such that $(v,x) \in E(G)$ and $(u,x) \notin E(G)$
   and
2. (a) $\exists x \in V(G)$ such that $(x,v) \in E(G)$ and $(x,u) \notin E(G)$ or

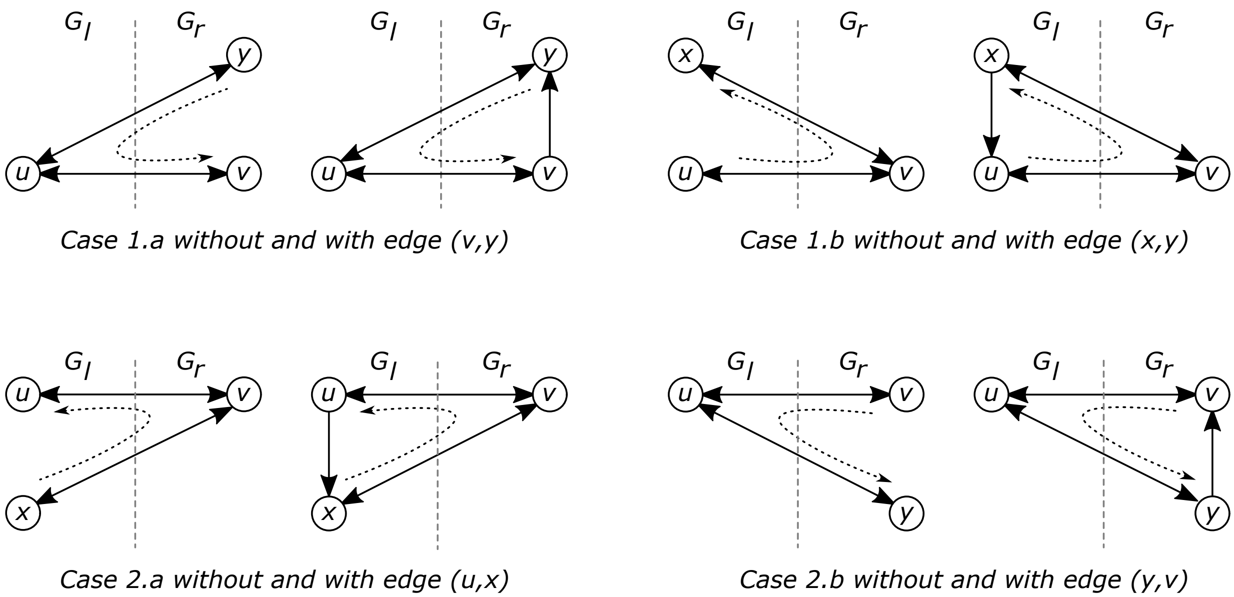


Figure 2: The four cases for the case that after a $\times$ operation between G_l and G_r there is an edge in $\overline{G_{\text{SR}}}$ between a vertex u from G_l and a vertex v from G_r. The figures shows the edges in G. Since we consider a join operation, there is an edge from every vertex of G_l to every vertex of G_r in G and vice versa.

(b) $\exists y \in V(G)$ such that $(u, y) \in E(G)$ and $(v, y) \notin E(G)$.

Since we consider a join operation of G_l and G_r, it follows that in Case 1.a and Case 2.b $y \in V(G_r)$ and in Case 1.b and Case 2.a $x \in V(G_l)$, see also Figure 2.

We call a vertex $u \in V(G)$

- a *solitary vertex* of G if there is a vertex $v \in V(G)$ such that $(u, v) \notin E(G)$ and $(v, u) \notin E(G)$,
- an *in-vertex* in G if there is a vertex $v \in V(G)$ such that $(v, u) \in E(G)$ and $(u, v) \notin E(G)$,
- an *out-vertex* in G if there is a vertex $v \in V(G)$ such that $(u, v) \in E(G)$ and $(v, u) \notin E(G)$,
- an *in-out-vertex* in G if u is an in-vertex and an out-vertex in G.

A vertex u can be a solitary, an in-vertex, an out-vertex and an in-out-vertex at the same time. If u is an in-out-vertex, then it is also an in-vertex and an out-vertex.

Lemma 5. *If node w is labelled by $\times$, then two vertices $u \in V(G_l)$ and $v \in V(G_r)$ are adjacent in $\overline{G_{\text{SR}}}$ if and only if*

1. *u is a solitary vertex in G_l or*
2. *v is a solitary vertex in G_r or*
3. *u is an in-out-vertex in G_l or*

4. *v is an in-out-vertex in G_r or*
5. *u is an in-vertex in G_l and v is an in-vertex in G_r or*
6. *u is an out-vertex in G_l and v is an out-vertex in G_r.*

Proof. $\Rightarrow$

1. If the conditions of Case 1.a and Case 2.a are satisfied, then either one of the two vertices u and v is a solitary vertex or u and v are both out-vertices in G.
2. If the conditions of Case 1.a and Case 2.b are satisfied, then either v is a solitary vertex or an in-out-vertex in G.
3. If the conditions of Case 1.b and Case 2.a are satisfied, then either u is a solitary vertex or an in-out-vertex in G.
4. If the conditions of Case 1.b and Case 2.b are satisfied, then either one of the two vertices u and v is a solitary vertex, or u and v are both in-vertices in G.

$\Leftarrow$

1. If u is a solitary vertex in G_l, then the conditions of Case 1.b and Case 2.a are satisfied.
2. If v is a solitary vertex in G_r, then the conditions of Case 1.a and Case 2.b are satisfied.
3. If u is an in-out-vertex in G_l, then the conditions of Case 1.b and Case 2.a are satisfied.
4. If v is an in-out-vertex in G_r, then the conditions of Case 1.a and Case 2.b are satisfied.
5. If u is an out-vertex in G_l and v is an out-vertex in G_r, then the conditions of Case 1.a and Case 2.a are satisfied.
6. If u is an in-vertex in G_l and v is an in-vertex in G_r, then the conditions of Case 1.b and Case 2.b are satisfied.

□

A maximum clique in $\overline{G_{\mathrm{SR}}}$ can now easily be computed with a bottom-up analysis of the co-tree T for G. For this purpose, only four size values denoted by m, s, i, o are required for each subtree T_w of the co-tree T.

1. m is the size of a maximum clique in $\overline{G_{\mathrm{SR}}}|_{V(G_w)}$,
2. s is the size of a maximum clique in $\overline{G_{\mathrm{SR}}}|_{V(G_w)}$ with exclusively solitary or in-out vertices,
3. i is the size of a maximum clique in $\overline{G_{\mathrm{SR}}}|_{V(G_w)}$ with exclusively in-vertices and
4. o is the size of a maximum clique in $\overline{G_{\mathrm{SR}}}|_{V(G_w)}$ with exclusively out-vertices.

The four size values m, s, i, o for subtree T_w can be easily computed from the four size values m_l, s_l, i_l, o_l and the four size values m_r, s_r, i_r, o_r for the left and right subtree T_l and T_r at root w, respectively. In parallel to these computations, it is also possible to manage four vertex sets M, S, O, I for maximum cliques in G_w with any vertices, exclusively solitary or in-out vertices, exclusively in-vertices, and exclusively out-vertices, respectively.

The size values m, s, i, o for G_w result from the maximum of several expressions formed from m_l, s_l, i_l, o_l for G_l and m_r, s_r, i_r, o_r for G_r. Depending on which expression yields the maximum, the corresponding vertex sets M, S, I, O for G_w are assembled from the vertex sets M_l, S_l, I_l, O_l for G_l and the vertex sets M_r, S_r, I_r, O_r for G_r.

For a leaf node w we have, $m = 1, s = 0, i = 0, o = 0$ and $M = \{\hat{w}\}, S = \emptyset, I = \emptyset, O = \emptyset$, where $\hat{w}$ is the vertex of G associated with w.

If w is labelled by $\cup$, then

1. m is the maximum of m_l (let $M = M_l$) and m_r (let $M = M_r$),
2. s is the maximum of m_l (let $S = M_l$) and m_r (let $S = M_r$),
3. i is the maximum of i_l (let $I = I_l$) and i_r (let $I = I_r$) and
4. o is the maximum of o_l (let $O = O_l$) and o_r (let $O = O_r$).

This operation does not create any edges in $\overline{G_{\text{SR}}}$. All vertices become solitary vertices.

If w is labelled by $\gg$, then

1. m is the maximum of m_l (let $M = M_l$) and m_r (let $M = M_r$),
2. s is the maximum of s_l (let $S = S_l$), s_r (let $S = S_r$), i_r (let $S = I_r$) and o_l (let $S = O_l$),
3. i is the maximum of m_r (let $I = M_r$) and i_l (let $I = I_l$) and
4. o is the maximum of m_l (let $O = M_l$) and o_r (let $O = O_r$).

This operation also does not create any edges in $\overline{G_{\text{SR}}}$. The in-vertices of G_l and the out-vertices of G_r become in-out-vertices in G_w. All vertices of G_l become out-vertices in G_w, all vertices of G_r become in-vertices in G_w.

If w is labelled by $\times$, then

1. m is the maximum of $m_l + s_r$ (let $M = M_l \cup S_r$), $m_r + s_l$ (let $M = M_r \cup S_l$), $i_l + i_r$ (let $M = I_l \cup I_r$) and $o_l + o_r$ (let $M = O_l \cup O_r$),
2. $s = s_l + s_r$ (let $S = S_l \cup S_r$),
3. $i = i_l + i_r$ (let $I = I_l \cup I_r$) and
4. $o = o_l + o_r$ (let $O = O_l \cup O_r$).

This fully describes the bottom-up processing of the co-tree T for the computation of a strong metric basis for co-graph G.

Algorithm 2 shows such a bottom-up processing of the co-tree T, in that the size values m, s, i, o are computed. See Figure 3 for an example. The algorithm can easily be extended to compute a minimum strong resolving set by keeping track of the vertex sets M, S, I, O. Since a binary co-tree of a directed co-graph can be found in linear time [34], we have proven the following theorem.

Theorem 5. *A strong metric basis of a strongly connected directed co-graph is computable in linear time.*

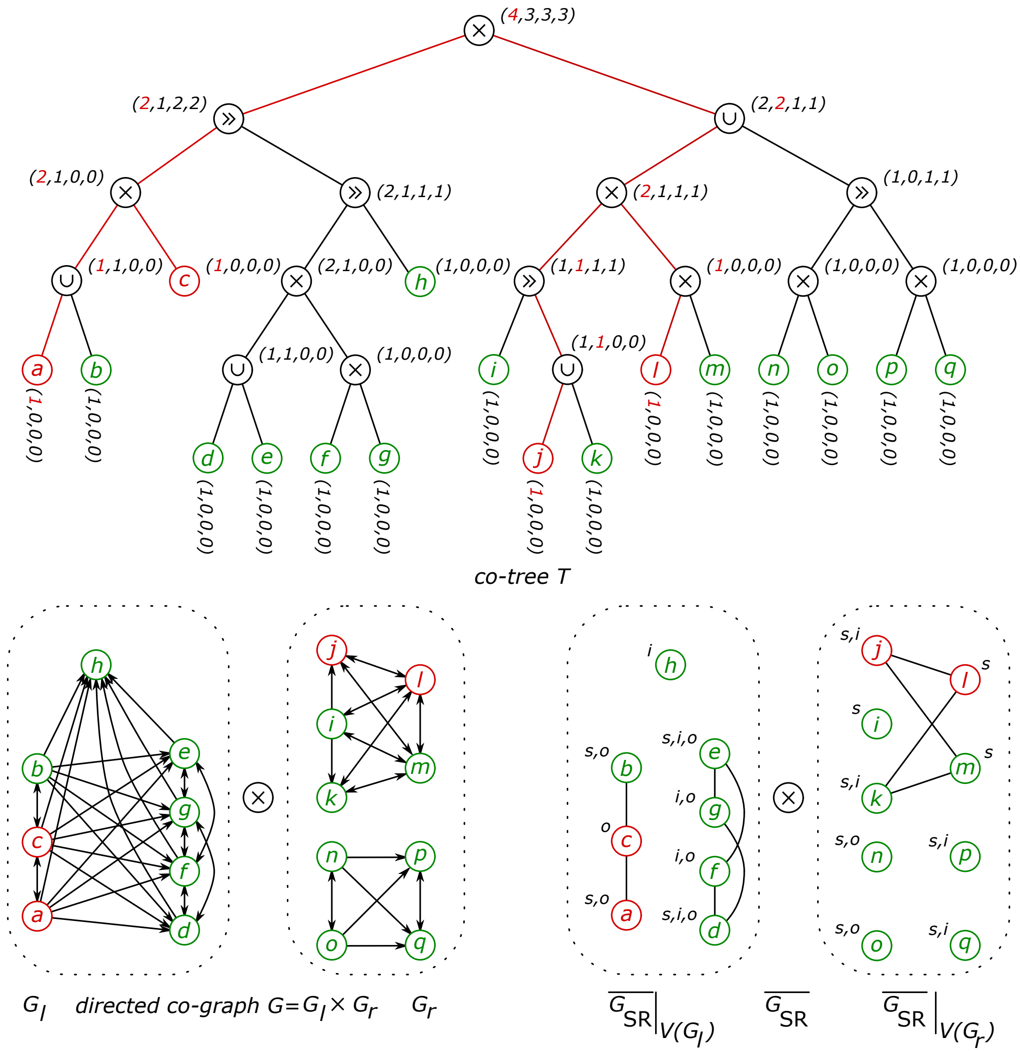


Figure 3: The figure shows a co-tree T with the size values at the nodes calculated by Algorithm 2. The figure also shows bottom left the co-graph G and bottom right the complement $\overline{G_{\mathrm{SR}}}$ of the strong resolving graph G_{SR}. The set of red vertices $\{a, c, j, l\}$ forms a maximum clique in $\overline{G_{\mathrm{SR}}}$. There are several options to select a clique with four vertices in $\overline{G_{\mathrm{SR}}}$. The set of green vertices $\{b, d, e, f, g, h, i, k, m, n, o, p, q\}$ is a strong metric basis for the directed co-graph G. Let G_l and G_r be the graphs defined by the subtrees left and right at the root of T. Then $G = G_l \times G_r$. In G_l the vertices a, b, d, e are solitary vertices, d, e, f, g, h are in-vertices and a, b, c, d, e, f, g are out-vertices. In G_r, the vertices i, j, k, l, m are solitary vertices, i, j, p, q are in-vertices and n, o are out-vertices. Graph $\overline{G_{\mathrm{SR}}}$ is formed from the disjoint union of $\overline{G_{\mathrm{SR}}}|_{V(G_l)}$ and $\overline{G_{\mathrm{SR}}}|_{V(G_r)}$ and the edges between $u \in V(G_l)$ and $v \in V(G_r)$, where u or v is a solitary vertex or in-out-vertex, or u and v are both in-vertices, or both are out-vertices, see Lemma 5. In the example all edges between vertices of G_l and G_r are inserted to get $\overline{G_{\mathrm{SR}}}$. For the sake of clarity, these edges are not shown in the figure.

Algorithm 2 Directed Strong Metric Dimension

function DSMD_DI_CO-GRAPH(directed co-graph G)
    $T \leftarrow$ binary co-tree of G
    $n \leftarrow |V(G)|$
    $(m, s, i, o) \leftarrow$ CLIQUE_VECTOR(T)
    return $n - m$
end

function CLIQUE_VECTOR(co-tree T)
    $w \leftarrow$ root of T
    if w is a leaf **then**
        return $(1, 0, 0, 0)$
    else
        $T_l \leftarrow$ left subtree of w
        $(m_l, s_l, i_l, o_l) \leftarrow$ CLIQUE_VECTOR(T_l)
        $T_r \leftarrow$ right subtree of w
        $(m_r, s_r, i_r, o_r) \leftarrow$ CLIQUE_VECTOR(T_r)
        if w is labelled $\cup$ **then**
            return $(\max\{m_l, m_r\}, \max\{m_l, m_r\}, \max\{i_l, i_r\}, \max\{o_l, o_r\})$
        else if w is labelled $\gg$ **then**
            return $(\max\{m_l, m_r\}, \max\{s_l, s_r, i_l, o_r\}, \max\{m_r, i_l\}, \max\{m_l, o_r\})$
        else $\triangleright$ w is labelled $\times$
            return $(\max\{m_l + s_r, m_r + s_l, i_l + i_r, o_l + o_r\}, s_l + s_r, i_l + i_r, o_l + o_r)$
        end
    end
end

4. Conclusions

In this paper we have shown that (Directed) Strong Metric Dimension for undirected and directed co-graphs is decidable in linear time. We even presented linear time algorithms for the computation of minimum strong resolving sets for undirected and directed co-graphs.

The strong metric dimension as well as the directed strong metric dimension have not yet been extensively discussed in the literature. Developing efficient algorithms to compute the strong metric dimension for specific graph classes (both undirected and directed) is one of the most interesting challenges for us.

References

[1] P. J. Slater, Leaves of trees, Congressum Numerantium 14 (1975) 549–559.

[2] F. Harary, R. A. Melter, On the metric dimension of a graph, Ars Combinatoria 2 (1976) 191–195.

[3] Z. Beerliova, F. Eberhard, T. Erlebach, A. Hall, M. Hoffmann, M. Mihaľák, L. S. Ram, Network discovery and verification, in: D. Kratsch (Ed.), Graph-Theoretic Concepts in Computer Science, Springer Berlin Heidelberg, 2005, pp. 127–138. doi:10.1007/11604686_12.
URL https://doi.org/10.1007/11604686_12

[4] K. Liu, N. B. Abu-Ghazaleh, Virtual coordinates with backtracking for void traversal in geographic routing, in: Ad-Hoc, Mobile, and Wireless Networks, 5th International Conference, ADHOC-NOW 2006, Ottawa, Canada, August 17-19, 2006, Proceedings, 2006, pp. 46–59. doi:10.1007/11814764_6.
URL https://doi.org/10.1007/11814764_6

[5] A. Sebö, E. Tannier, On metric generators of graphs, Mathematics of Operations Research 29 (2) (2004) 383–393. doi:10.1287/moor.1030.0070.
URL https://doi.org/10.1287/moor.1030.0070

[6] S. Hoffmann, E. Wanke, Metric dimension for gabriel unit disk graphs is np-complete, in: Algorithms for Sensor Systems, 8th International Symposium on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile Entities, ALGOSENSORS 2012, Ljubljana, Slovenia, September 13-14, 2012. Revised Selected Papers, 2012, pp. 90–92. doi:10.1007/978-3-642-36092-3_10.
URL https://doi.org/10.1007/978-3-642-36092-3_10

[7] S. Khuller, B. Raghavachari, A. Rosenfeld, Landmarks in graphs, Discrete Applied Mathematics 70 (3) (1996) 217–229. doi:10.1016/0166-218X(95)00106-2.
URL https://doi.org/10.1016/0166-218X(95)00106-2

[8] G. Chartrand, L. Eroh, M. A. Johnson, O. Oellermann, Resolvability in graphs and the metric dimension of a graph, Discrete Applied Mathematics 105 (1-3) (2000) 99–113. doi:10.1016/S0166-218X(00)00198-0.
URL https://doi.org/10.1016/S0166-218X(00)00198-0

[9] S. Hayat, Computing distance-based topological descriptors of complex chemical networks: New theoretical techniques, Chemical Physics Letters 688 (1) (1977) 51–58. doi:10.1016/j.cplett.2017.09.055.
URL https://doi.org/10.1016/j.cplett.2017.09.055

[10] S. Khuller, B. Raghavachari, A. Rosenfeld, Localization in graphs, 1994.

[11] O. R. Oellermann, J. Peters-Fransen, The strong metric dimension of graphs and digraphs, Discrete Applied Mathematics 155 (3) (2007) 356–364. doi:10.1016/j.dam.2006.06.009.
URL `https://doi.org/10.1016/j.dam.2006.06.009`

[12] M. C. Hernando, M. Mora, I. M. Pelayo, C. Seara, J. Cáceres, M. L. Puertas, On the metric dimension of some families of graphs, Electronic Notes in Discrete Mathematics 22 (2005) 129–133. doi:10.1016/j.endm.2005.06.023.
URL `https://doi.org/10.1016/j.endm.2005.06.023`

[13] R. A. Melter, I. Tomescu, Metric bases in digital geometry, Computer Vision, Graphics, and Image Processing 25 (1) (1984) 113–121. doi:10.1016/0734-189X(84)90051-3.
URL `https://doi.org/10.1016/0734-189X(84)90051-3`

[14] M. Bača, E. T. Baskoro, A. N. M. Salman, S. W. Saputro, D. Suprijanto, The metric dimension of regular bipartite graphs, Bulletin mathématiques de la Société des sciences mathématiques de Roumanie 54 (1) (2011) 15–28.

[15] H. Iswadi, E. T. Baskoro, A. Salman, R. Simanjuntak, The metric dimension of amalgamation of cycles, Far East Journal of Mathematical Sciences (FJMS) 41 (1) (2010) 19–31.

[16] S. Hoffmann, A. Elterman, E. Wanke, A linear time algorithm for metric dimension of cactus block graphs, Theoretical Computer Science 630 (2016) 43–62. doi:10.1016/j.tcs.2016.03.024.
URL `https://doi.org/10.1016/j.tcs.2016.03.024`

[17] D. Vietz, S. Hoffmann, E. Wanke, Computing the metric dimension by decomposing graphs into extended biconnected components - (extended abstract), in: WALCOM: Algorithms and Computation - 13th International Conference, WALCOM 2019, Guwahati, India, February 27 - March 2, 2019, Proceedings, 2019, pp. 175–187. doi:10.1007/978-3-030-10564-8_14.
URL `https://doi.org/10.1007/978-3-030-10564-8_14`

[18] J. Díaz, O. Pottonen, M. J. Serna, E. J. van Leeuwen, On the complexity of metric dimension, in: Algorithms - ESA 2012 - 20th Annual European Symposium, Ljubljana, Slovenia, September 10-12, 2012. Proceedings, 2012, pp. 419–430. doi:10.1007/978-3-642-33090-2_37.
URL `https://doi.org/10.1007/978-3-642-33090-2_37`

[19] M. Hauptmann, R. Schmied, C. Viehmann, Approximation complexity of metric dimension problem, Journal of Discrete Algorithms 14 (2012) 214–222. doi:10.1016/j.jda.2011.12.010.
URL `https://doi.org/10.1016/j.jda.2011.12.010`

[20] G. G. Chappell, J. G. Gimbel, C. Hartman, Bounds on the metric and partition dimensions of a graph, Ars Combinatoria 88 (2008).

[21] G. Chartrand, C. Poisson, P. Zhang, Resolvability and the upper dimension of graphs, Computers and Mathematics with Applications 39 (12) (2000) 19–28.

[22] D. Kuziak, I. G. Yero, J. A. Rodríguez-Velázquez, On the strong metric dimension of corona product graphs and join graphs, Discrete Applied Mathematics 161 (7-8) (2013) 1022–1027. doi:10.1016/j.dam.2012.10.009.
URL `https://doi.org/10.1016/j.dam.2012.10.009`

[23] D. Kuziak, J. A. Rodríguez-Velázquez, I. G. Yero, On the strong metric dimension of product graphs, Electronic Notes in Discrete Mathematics 46 (2014) 169–176. doi:10.1016/j.endm.2014.08.023.
URL `https://doi.org/10.1016/j.endm.2014.08.023`

[24] J. A. Rodríguez-Velázquez, I. G. Yero, D. Kuziak, O. R. Oellermann, On the strong metric dimension of cartesian and direct products of graphs, Discrete Mathematics 335 (2014) 8–19. doi:10.1016/j.disc.2014.06.023.
URL `https://doi.org/10.1016/j.disc.2014.06.023`

[25] D. Kuziak, I. G. Yero, J. A. Rodríguez-Velázquez, On the strong metric dimension of cartesian sum graphs, Fundamenta Informaticae 141 (1) (2015) 57–69. doi:10.3233/FI-2015-1263.
URL `https://doi.org/10.3233/FI-2015-1263`

[26] D. Kuziak, I. G. Yero, J. A. Rodríguez-Velázquez, Strong metric dimension of rooted product graphs, International Journal of Computer Mathematics 93 (8) (2016) 1265–1280. doi:10.1080/00207160.2015.1061656.
URL `https://doi.org/10.1080/00207160.2015.1061656`

[27] D. Kuziak, J. A. Rodríguez-Velázquez, I. G. Yero, Closed formulae for the strong metric dimension of lexicographic product graphs, Discussiones Mathematicae Graph Theory 36 (4) (2016) 1051–1064. doi:10.7151/dmgt.1911.
URL `https://doi.org/10.7151/dmgt.1911`

[28] T. R. May, O. R. Oellermann, The strong dimension of distance-hereditary graphs, JCMCC-Journal of Combinatorial Mathematics and Combinatorial Computing 76 (2011) 59.

[29] G. Moravcik, O. R. Oellermann, S. Yusim, Comparing the metric and strong dimensions of graphs, Discrete Applied Mathematics 220 (2017) 68–79. doi:10.1016/j.dam.2016.12.020.
URL `https://doi.org/10.1016/j.dam.2016.12.020`

[30] D. G. Corneil, H. Lerchs, L. S. Burlingham, Complement reducible graphs, Discrete Applied Mathematics 3 (3) (1981) 163–174.

[31] X. Ma, M. Feng, K. Wang, The strong metric dimension of the power graph of a finite group, Discrete Applied Mathematics 239 (2018) 159–164. doi:10.1016/j.dam.2017.12.021.
URL `https://doi.org/10.1016/j.dam.2017.12.021`

[32] M. Habib, C. Paul, A simple linear time algorithm for cograph recognition, Discrete Applied Mathematics 145 (2) (2005) 183–197.

[33] D. Bechet, P. De Groote, C. Retoré, A complete axiomatisation for the inclusion of series-parallel partial orders, in: International Conference on Rewriting Techniques and Applications, Springer, 1997, pp. 230–240.

[34] C. Crespelle, C. Paul, Fully dynamic recognition algorithm and certificate for directed cographs, Discrete Applied Mathematics 154 (12) (2006) 1722–1741.