



Polynomial algorithms for sparse spanners on subcubic graphs

R. Gómez¹ · F. K. Miyazawa² · Y. Wakabayashi³

Accepted: 13 July 2024 / Published online: 7 August 2024

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract

Let G be a connected graph and $t \geq 1$ a (rational) constant. A t -spanner of G is a spanning subgraph of G in which the distance between any pair of vertices is at most t times its distance in G . We address two problems on spanners. The first one, known as the *minimum t -spanner problem* (MINS_t), seeks in a connected graph a t -spanner with the smallest possible number of edges. In the second one, called *minimum cost tree t -spanner problem* (MCTS_t), the input graph has costs assigned to its edges and seeks a t -spanner that is a tree with minimum cost. It is an optimization version of the *tree t -spanner problem* (TREES_t), a decision problem concerning the existence of a t -spanner that is a tree. MINS_t is known to be NP-hard for every $t \geq 2$. On the other hand, TREES_t admits a polynomial-time algorithm for $t \leq 2$ and is NP-complete for $t \geq 4$; but its complexity for $t = 3$ remains open. We focus on the class of subcubic graphs. First, we show that for such graphs MINS_3 can be solved in polynomial time. These results yield a practical polynomial algorithm for TREES_3 that is of a combinatorial nature. We also show that MCTS_2 can be solved in polynomial time. To obtain this last result, we prove a complete linear characterization of the polytope defined by the incidence vectors of the tree 2-spanners of a subcubic graph. A recent result showing that MINS_3 on graphs with maximum degree at most 5 is NP-hard, together with the current result on subcubic graphs, leaves open only the complexity of MINS_3 on graphs with maximum degree 4.

Keywords Spanner · Tree spanner · Subcubic graph · Polyhedra

✉ Y. Wakabayashi
yw@ime.usp.br

R. Gómez
gomez.renzo@ufabc.edu.br

F. K. Miyazawa
fkm@ic.unicamp.br

¹ Centro de Matemática, Computação e Cognição, Universidade Federal do ABC, Santo André, Brazil

² Instituto de Computação, Universidade Estadual de Campinas, Campinas, Brazil

³ Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, Brazil

Mathematics Subject Classification 05C85 · 90C27

1 Introduction

Throughout this text, we consider that the input graph is always connected (even if this is not stated explicitly). The *distance* between two vertices u and v in a graph G , denoted by $d_G(u, v)$, is the minimum length of a path between u and v . For a (rational) constant $t \geq 1$, a t -*spanner* of a graph G is a spanning subgraph H of G in which

$$d_H(u, v) \leq t \cdot d_G(u, v), \text{ for all } u, v \in V(G).$$

A synchronizer is a technique in distributed computing, proposed by Awerbuch (1985), that transforms a synchronous algorithm into an asynchronous one. Motivated by this work, Peleg and Ullman (1989) introduced the concept of spanner, and discovered its relation with the efficiency of a synchronizer of a network. As a result, they showed how to construct an optimal synchronizer for the hypercube. Since then, spanners have raised attention, both from theoretical and practical point-of-view, by its wide range of applicability. It has applications in areas such as distributed systems and communication networks (synchronization, building succinct and efficient routing tables (Peleg and Upfal 1989), distance oracles (Thorup and Zwick 2005; Baswana and Sen 2006), roadmap planning (Wang et al. 2015), computational geometry, robotics, etc. The reader may refer to Ahmed et al. (2020) for a survey that gives an overview of the results regarding many different variants of graph spanners (Euclidean, weighted, on directed graphs, etc), mentioning applications and providing a list of open problems.

We consider here two problems regarding t -spanners. In the first one, the MINIMUM t - SPANNER PROBLEM (MINS_t), we are given a graph and we wish to find a t -spanner of this graph with the smallest possible number of edges. The second problem, called MINIMUM COST TREE t - SPANNER PROBLEM (MCTS_t), is a natural generalization of the TREE t - SPANNER PROBLEM (TREES_t). In TREES_t , we are interested in deciding whether a graph admits a t -spanner that is a tree. In MCTS_t , we are given a graph with edge-costs and we wish to find a tree t -spanner of minimum total cost (if one exists).

In 1989, Peleg and Schäffer (1989) introduced MINS_t and proved that MINS_2 is NP-hard. Later, Cai (1994) extended this result for every $t \geq 2$. In 1997, Venkatesan et al. (1997) improved further this result showing that MINS_t is NP-hard for $t \geq 2$ even if the graph is chordal. Searching for classes of graphs for which MINS_t can be solved efficiently, Cai and Keil (1994) studied graphs of bounded degree. Let $\Delta(G)$ (or simply Δ , when the graph under consideration is clear from the context) denote the maximum degree of a graph G . Cai and Keil designed a polynomial-time algorithm for MINS_2 on graphs with $\Delta \leq 4$. Moreover, they proved that MINS_t is NP-hard when $t \geq 2$ and $\Delta \leq 9$. Kobayashi (2018) showed that MINS_t remains NP-hard on planar graphs in the following cases: (a) $t = 2$ and $\Delta \leq 8$; and (b) $3 \leq t \leq 4$ and $\Delta \leq 6$. Recently, Gómez et al. (2023) proved that the maximum degree condition can be slightly lowered: MINS_t remains NP-hard on planar graphs when $t = 2$ and $\Delta \leq 7$, when $t = 3$ and $\Delta \leq 5$, and when $t \geq 4$ and $\Delta \leq 4$. Some of these results are summarized in the next table.

Table 1 Computational complexity of MIN_t on graphs with maximum degree at most k

$\Delta \leq k$	$t = 2$	$t = 3$	$t = 4$	$t \geq 5$
$k = 3$	P (Cai and Keil 1994)	P (*)	open	open
$k = 4$	P (Cai and Keil 1994)	open	NP-hard (Gómez et al. 2023)	NP-hard (Gómez et al. 2023)
$k = 5$	open	NP-hard Gómez et al. (2023)	NP-hard (Gómez et al. 2023)	NP-hard (Gómez et al. 2023)
$k = 6$	open	NP-hard Kobayashi (2018)	NP-hard Kobayashi 2018	NP-hard (Gómez et al. 2023)
$k = 7$	NP-hard (Gómez et al. 2023)	NP-hard (Kobayashi 2018)	NP-hard (Kobayashi 2018)	NP-hard (Gómez et al. 2023)
$k = 8$	NP-hard (Kobayashi 2018)	NP-hard (Kobayashi 2018)	NP-hard (Kobayashi 2018)	NP-hard (Gómez et al. 2023)
$k \geq 9$	NP-hard (Cai and Keil 1994)	NP-hard (Cai and Keil 1994)	NP-hard (Cai and Keil 1994)	NP-hard (Cai and Keil 1994)

All hardness results hold for planar graphs; (*) refers to this work

Regarding tree t -spanners, the following results are known. Cai and Corneil (1995) showed a linear-time algorithm for TREES_t when $t \leq 2$, and also showed that it becomes NP-complete when $t \geq 4$. The complexity status of TREES_3 has not been established. For bounded-degree graphs, in contrast to MINS_t , Fomin et al. (2011) showed a polynomial-time algorithm for TREES_t . It is also known that TREES_3 admits a polynomial-time algorithm on several classes of graphs such as planar graphs (Fekete and Kremer 2001), convex graphs (Venkatesan et al. 1997), split graphs (Venkatesan et al. 1997), line graphs (Couto et al. 2021), etc. We summarize some of these results in Table 2. The minimization version of TREES_t , denoted here MCTS_t , has not been investigated in the literature. We present here a result for MCTS_2 .

In this text, we focus on subcubic graphs (those with maximum degree 3). Our contributions to these problems are the following:

- (1) A simple polynomial-time algorithm for MINS_3 ;
- (2) A simple practical algorithm for TREES_3 ;
- (3) A polynomial-time algorithm for MCTS_2 .

To obtain the result (1), we designed an algorithm that is based on a technique introduced by Cai and Keil (1994) for MINS_2 on graphs of maximum degree 4. These authors believed that such a technique would yield a polynomial-time algorithm for MINS_3 on subcubic graphs. The result shown here confirms this, but the proof turned out to be quite involved, as many different graphs were obtained with this approach and we had to characterize (the families of) these graphs. This was the hardest part of our proof, but we were able to characterize which are the 11 small graphs and the 6 simple classes of graphs that we may obtain, and for each of them we could state explicitly how to find a minimum 3-spanner.

Let us turn now our attention to tree spanners. We observe that our algorithm for MINS_3 also gives a solution for TREES_3 . For this problem, Fomin et al. (2011) proposed a linear-time algorithm on bounded-degree graphs based on Courcelle's theorem (Courcelle and Engelfriet 2012). Although its complexity is theoretically optimal, it is not efficient in practice. Moreover, Papoutsakis (2018) also proposed a dynamic-programming based polynomial-time algorithm; however, its implementation is quite involved (see further comments on Sect. 3). Our contribution is an alternative efficient (practical) polynomial-time algorithm for TREES_3 on subcubic graphs.

Recent works have investigated the problems related to sparse spanners from a practical point-of-view (Álvarez-Miranda and Sinnl 2019; Ahmed et al. 2019). Some linear formulations have been proposed for TREES_t , however the experimental results were not very promising. Motivated by this fact, we studied the polytope associated with the incidence vectors of the tree 2-spanners of a graph. We obtained a complete linear description of the polytope associated with these vectors, which in turn implies (3).

We conclude this section describing the organization of this text. In Sect. 2, we introduce some concepts and the terminology that is used in the text. We define a partition of the edges of a graph G , denoted $\mathcal{C}_t(G)$, which is the central idea used to tackle MINS_3 and MCTS_2 on subcubic graphs. In Sect. 3, we focus on MINS_3 . This section is devoted to characterizing the subgraphs that arise from the partition $\mathcal{C}_3(G)$ when G is a subcubic graph. From this characterization, we derive a polynomial-time algorithm for MINS_3 . This result also implies a simple algorithm for TREES_3

Table 2 Computational complexity of $TREES_t$ on some classes of graphs

Graph class	$t = 3$	$t = 4$	$t \geq 5$
Chordal	Open	NP-complete (Dragan and Köhler 2014)	NP-complete (Dragan and Köhler 2014)
Planar	P (Fekete and Kremer 2001)	NP-complete (Dragan et al. 2011)	NP-complete (Dragan et al. 2011)
Bounded-degree	P (Fomin et al. 2011)	P (Fomin et al. 2011)	P (Fomin et al. 2011)
Prisms of graphs	P (Gómez et al. 2022)	Open	NP-complete (Couto and Cunha 2021)
Bipartite	Open	Open	NP-complete (Brandstädt et al. 2007)
Chordal bipartite	Open	Open	NP-complete (Brandstädt et al. 2007)

on this class of graphs. After that, we turn our attention to $MCTS_2$ in Sect. 4. We focus on the polytope associated with the feasible solutions of $TREES_2$, and show a linear description of this polytope. This result implies a polynomial-time algorithm for $MCTS_2$ on subcubic graphs. To the best of our knowledge, this is a novel approach and result. In Sect. 5, we mention some final remarks and possible directions for further research.

A preliminary version of this work (Gómez et al. 2022) was presented at the 16th International Conference and Workshops on Algorithms and Computation (WALCOM 2022).

2 Preliminaries

When we refer to a graph, say G , without specifying explicitly its vertex set and edge set, these are supposed to be $V(G)$ and $E(G)$. The *length* of a path or cycle in a graph is its number of edges. Moreover, if a path (resp. cycle) has length k , we say that it is a k -path (resp. k -cycle). We use the notation $\deg_G(v)$ for the *degree* of a vertex v in a graph G .

The following result gives an equivalent definition for t -spanner. It is very useful, as it tells us that, a spanning subgraph H of a graph G is a t -spanner of G if the distance condition in H holds for pairs of adjacent vertices in G (see also Cai and Corneil (1995)).

Proposition 1 (Peleg and Schäffer (1989)) *Let H be a spanning subgraph of a graph $G = (V, E)$. Then, H is a t -spanner of G if and only if for every edge $uv \in E$, $d_H(u, v) \leq t$.*

In what follows, we define a partition of the edges of a graph that helps us subdivide $MINS_t$ (and $TREES_t$) into (possibly) smaller subproblems. This idea was introduced by Cai and Keil (1994) for the case of 2-spanners. This partition is motivated by the following observation. Let $G = (V, E)$ be a graph, and let H be a t -spanner of G . Consider an edge $e \in E \setminus E(H)$. Then, there exists in H a path P linking the ends of e such that $|P| \leq t$. Thus, the edge e is contained in a cycle of length at most $t + 1$. Let now L be the graph defined from G as follows.

$$\begin{aligned} V(L) &= \{v_e : e \in E\}, \\ E(L) &= \{v_e v_f : e, f \in E \text{ belong to a } k\text{-cycle in } G, k \leq t + 1\}. \end{aligned}$$

Finally, let us denote by $\mathcal{C}_t(G)$ the partition of E defined as follows. Two edges $e, f \in E$ belong to the same class of $\mathcal{C}_t(G)$ if and only if v_e and v_f belong to the same connected component of L . Observe that, if an edge $uv \in E$ does not belong to H , the edges of a k -path, $k \leq t$, linking u and v in H belong to the class that contains uv in $\mathcal{C}_t(G)$.

The key idea behind the algorithm of Cai and Keil (1994) for $MINS_2$ on graphs of maximum degree 4 is to characterize the graphs in $\mathcal{C}_2(G)$. We follow this approach and study the partition $\mathcal{C}_3(G)$ on subcubic graphs. We show in Fig. 1 an example of a graph G , its associated graph L , and the classes in $\mathcal{C}_2(G)$. The vertices of L are

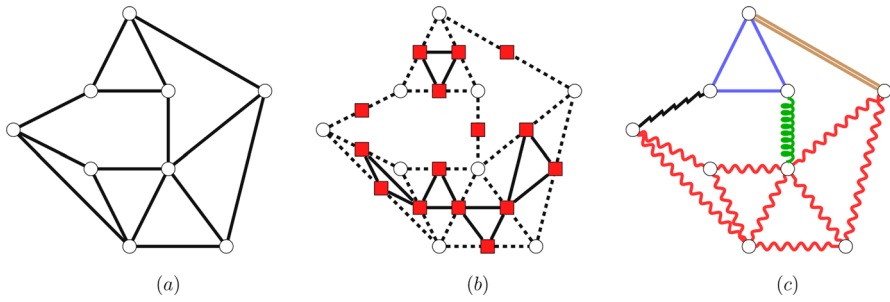


Fig. 1 **a** A graph G ; **b** the graph L (its vertices are shaded rectangles, and its edges are depicted by full black edges); **c** the five classes in $\mathcal{C}_2(G)$ (represented by five different types of edges) (Color figure online)

depicted by full squares, and its edges are depicted by (full) black edges. The classes in $\mathcal{C}_2(G)$ are represented by different types of edges. For simplicity, we consider each class in $\mathcal{C}_i(G)$ as a subgraph of G .

The following result is very useful, since it reduces MINS_t to finding a minimum t -spanner for each graph in $\mathcal{C}_i(G)$.

Proposition 2 *A subgraph S of a graph G is a t -spanner if and only if $S \cap H$ is a t -spanner of H , for every $H \in \mathcal{C}_i(G)$.*

Proof Let S be a subgraph of G . First, let us prove that if S is a t -spanner of G , then $S \cap H$ is a t -spanner of H , for each $H \in \mathcal{C}_i(G)$. Take $H \in \mathcal{C}_i(G)$ and $uv \in E(H) \setminus E(S)$. As $uv \notin E(S)$, there exists in S a path, say P , between u and v such that $|P| \leq t$. Note that P is also a path in H since $P + e$ is a k -cycle with $k \leq t$. Thus, P is a path in $S \cap H$, and therefore $S \cap H$ is a t -spanner of H .

Let us prove now that if $S \cap H$ is a t -spanner of H , for every $H \in \mathcal{C}_i(G)$, then S is a t -spanner of G . Let $uv \in E(G) \setminus E(S)$. Since $\mathcal{C}_i(G)$ is a collection of subgraphs that partitions $E(G)$, there exists a subgraph $H \in \mathcal{C}_i(G)$ such that $uv \in E(H)$. As $S \cap H$ is a t -spanner of H , there exists a k -path, $k \leq t$, between u and v in $S \cap H$. Therefore, S is a t -spanner of G .

Proposition 2 tells us that a t -spanner of a graph G is composed of t -spanners for each subgraph $H \in \mathcal{C}_i(G)$. This approach is also valid for tree t -spanners, however we have to carry out a final test: the union of the minimum t -spanners of each graph in $\mathcal{C}_i(G)$ needs to be a tree. Finally, we note that the partition $\mathcal{C}_i(G)$ can be obtained in polynomial time. This follows from the fact that determining whether two edges belong to the same class (in $\mathcal{C}_i(G)$) reduces to finding the length of a shortest cycle in G containing these edges.

3 Minimum 3-spanner on subcubic graphs

In this section, we focus on 3-spanners. In particular, we are interested in characterizing the graphs in $\mathcal{C}_3(G)$, for any subcubic graph G . Throughout this section, we denote by $G = (V, E)$ a connected subcubic graph.

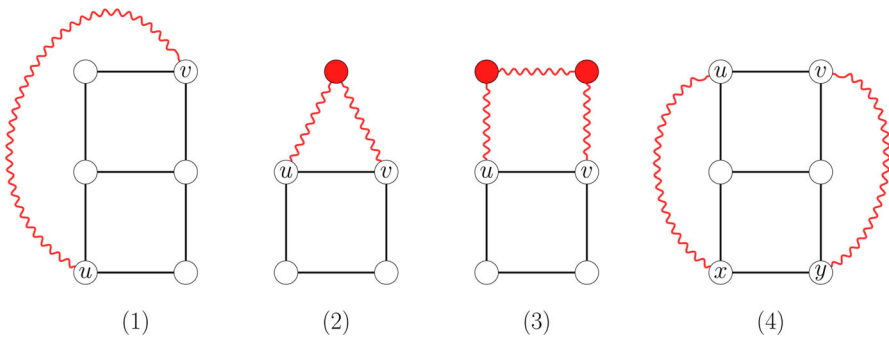


Fig. 2 Graphs obtained after applying operation (1), (2), (3), or (4)

We follow a constructive approach to characterize the graphs in $\mathcal{C}_3(G)$. More formally, we define four operations and show that, for any graph $H \in \mathcal{C}_3(G)$, there exists a sequence of graphs, say H_0, H_1, \dots, H_n , such that $H_0 \cong K_2, H_n \cong H$ and H_{i+1} is obtained by applying one of these operations to H_i , for $i = 0, 1, \dots, n - 1$.

Let H_i be a subcubic graph, and let u and v be distinct vertices of degree at most two in H_i . The first three operations are the following.

- (1) Add to H_i an edge linking u and v , if $uv \notin E(H_i)$ and $d_{H_i}(u, v) \leq 3$.
- (2) Add to H_i a 2-path between u and v , if $d_{H_i}(u, v) \leq 2$.
- (3) Add to H_i a 3-path between u and v , if $uv \in E(H_i)$.

To state the last operation, first we introduce the following definition. Let uv and xy be distinct edges in $E(H_i)$ such that u, v, x and y have degree two in H_i . Moreover, suppose that $ux, vy \notin E(H_i)$ or $uy, vx \notin E(H_i)$. We call any of these pairs of edges a *matching* between uv and xy .

- (4) Add to H_i a matching between the edges uv and xy .

Note that, each operation increases the degree of at least two vertices in the graph. Moreover, since $H_0 \cong K_2$, then H_1 is either a 3-cycle or a 4-cycle. Thus, $\delta(H_i) \geq 2$, for $i = 1, \dots, n$. (The notation $\delta(H)$ stands for the minimum degree of a graph H .) In Fig. 2, we show examples of the graphs that we obtain after applying these operations. We depict by solid vertices and wavy edges the vertices and edges that are added to H_i , respectively.

The next result gives us a constructive characterization of the graphs in $\mathcal{C}_3(G)$. With this lemma at hand, we will be able to list the graphs in $\mathcal{C}_3(G)$.

Lemma 3 *Let $H \in \mathcal{C}_3(G)$. Then, there exists a sequence of graphs H_0, \dots, H_n such that $H_0 \cong K_2, H_n \cong H$, and H_{i+1} is obtained from H_i by applying one of the operations (1), (2), (3) or (4).*

Proof First, if H is a single edge there is nothing to prove, since $H_0 \cong K_2$. So suppose that $H \neq K_2$. Observe that H contains a cycle of length at most four by the definition of $\mathcal{C}_3(G)$. Let H' be a maximal subgraph of H for which such sequence of graphs exists. As H contains a 3-cycle or 4-cycle, we have that $H' \neq K_2$, and therefore $\delta(H') \geq 2$.

Let us now suppose, by contradiction, that $H' \neq H$. First, we show that there is a cycle C of length at most four, in H , such that $E(C) \cap E(H') \neq \emptyset$ and $E(C) \setminus E(H') \neq \emptyset$. To show the previous claim, we recall that $\mathcal{C}_3(G)$ is obtained from a graph L , defined from G , such that there exists a vertex $v_e \in V(L)$ for each edge $e \in E(G)$. Moreover, the edge $v_e v_f \in E(L)$ if $e, f \in E(G)$ belong to a cycle in G of length at most four (see Sect. 2). Consider now the connected component in L , say L_H , that corresponds to $E(H)$. Since $H' \neq H$ and L_H is connected, there exist edges $e \in E(H) \setminus E(H')$ and $f \in E(H')$ such that v_e is adjacent to v_f in L_H . Then, by the definition of $E(L)$, there exists a cycle C in G of length at most four that contains e and f . Moreover, $E(C) \subseteq E(H)$. This implies that $E(C) \cap E(H') \neq \emptyset$ and $E(C) \setminus E(H') \neq \emptyset$. Let C be one such cycle. We distinguish three cases:

Case 1: $|E(C) \cap E(H')| = 3$

In this case, $V(C) \subseteq V(H')$ and $|C| = 4$. Let $C = \langle x, y, x', y' \rangle$, such that $xy' \notin E(H')$. Thus, $\deg_{H'}(x) \leq 2$, $\deg_{H'}(y') \leq 2$ and $d_{H'}(x, y') \leq 3$. Therefore, if we apply operation (1) on vertices x and y' , we obtain the graph $H' \cup C$, which contradicts the maximality of H' .

Case 2: $|E(C) \cap E(H')| = 2$

First, suppose that $|E(C) \cap E(H')|$ is a matching. This implies that C is a 4-cycle. Let $C = \langle x, y, x', y' \rangle$, such that xy and $x'y'$ belong to H' . As the edges xy' and $x'y$ do not belong to H' , these edges are a matching of xy and $x'y'$. Thus, if we apply operation (4) on xy and $x'y'$ we obtain the graph $H' \cup C$, a contradiction.

Suppose now that $E(H') \cap E(C)$ is a 2-path, and consider that $C = \langle x, y, x', y' \rangle$. Furthermore, suppose that the edges xy and yx' belong to H' . We claim that the vertex y' does not belong to H' . If this is not the case, as the edges xy' and $x'y'$ do not belong to H' and $\Delta(H) \leq 3$, we have that $\deg_{H'}(y') \leq 1$. But this contradicts the fact that $\delta(H') \geq 2$. Therefore, if we apply operation (2) on vertices x and x' , we obtain $H' \cup C$, a contradiction.

On the other hand, if C is a 3-cycle, we have that $V(C) \subseteq V(H')$. In this case, if we apply operation (1), we also obtain a contradiction.

Case 3: $|E(C) \cap E(H')| = 1$

Without loss of generality, suppose that $|C| = 3$ (the other case is analogous). Let $C = \langle x, y, z \rangle$ such that $xy \in H'$. Since $\Delta(H) \leq 3$ and the edges xz and yz do not belong to $E(H')$, we have that $z \notin V(H')$. As $\deg_{H'}(x) \leq 2$ and $\deg_{H'}(y) \leq 2$, we can apply operation (3) to obtain the graph $H' \cup C$, a contradiction.

Therefore, $H' = H$.

In what follows, we define some classes of graphs that are part of our characterization of $\mathcal{C}_3(G)$. The k -ladder graph, denoted by L_k , is the cartesian product of a k -path with K_2 . That is, L_k is obtained from the union of two copies of a k -path by adding a perfect matching between the corresponding vertices. We note that L_k has exactly four vertices of degree two, say x_1, x_2, y_1 and y_2 , such that x_1y_1, x_2y_2 belong to $E(L_k)$ (see Fig. 3a). We define now four graphs related to L_k . Let M_k (resp. N_k) be the graph obtained from L_k by adding the edges x_1y_2 and x_2y_1 (resp. x_1x_2 and y_1y_2), for $k \geq 2$. On the other hand, T_k^1 is the graph obtained from L_k by adding a 2-path between x_1 and y_1 . Moreover, if we also add a 2-path between x_2 and y_2 , we obtain the graph T_k^2 . Figure 3 shows an example of those graphs for $k = 2$.

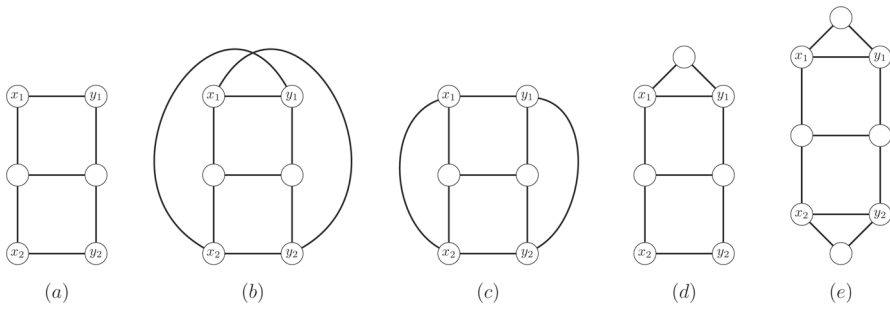


Fig. 3 **a** L_2 ; **b** M_2 ; **c** N_2 ; **d** T_2^1 ; **e** T_2^2

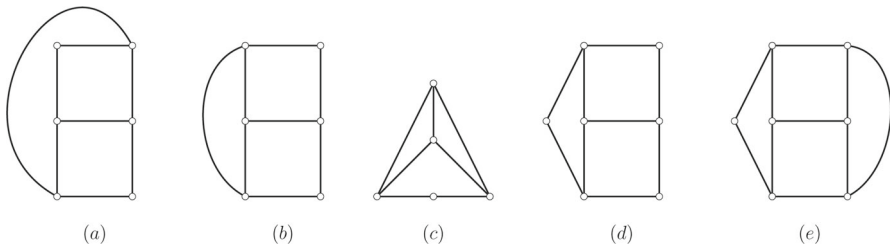


Fig. 4 **a** $M_2 - e$; **b** $N_2 - e$; **c** G_1 ; **d** G_2 ; **e** G_3

We denote by $M_k - e$ (resp. $N_k - e$) the graph obtained from M_k (resp. N_k) by removing the edge x_1y_2 (resp. y_1y_2). We show an example of $M_2 - e$ and $N_2 - e$ in Fig. 4. In this figure, we also show three graphs G_1 , G_2 and G_3 that are part of our characterization.

Let \mathcal{F}_3 be the family that consists of the following graphs (see Fig. 5):

- (a) K_2 (e) $K_{2,3}$ (i) G_1 (m) $M_k, k \geq 2$
- (b) K_3 (f) $M_2 - e$ (j) G_2 (n) $N_k, k \geq 2$
- (c) $K_4 - e$ (g) $N_2 - e$ (k) G_3 (o) $T_k^1, k \geq 1$
- (d) K_4 (h) $N_3 - e$ (l) $L_k, k \geq 1$ (p) $T_k^2, k \geq 1$

We show now the main result of this section.

Theorem 4 *Let G be a connected subcubic graph. If $H \in \mathcal{C}_3(G)$, then $H \in \mathcal{F}_3$.*

Proof Let $H \in \mathcal{C}_3(G)$. By Lemma 3, there is a sequence of graphs H_0, \dots, H_n such that

- $H_0 \cong K_2$,
- $H_n = H$,
- H_{i+1} is obtained from H_i by applying one of the operations (1), (2), (3) or (4).

In what follows, we show that $H_i \in \mathcal{F}_3$, for $i = 0, \dots, n$. For this, if it is possible to apply an operation to H_i , we will analyse each case. In particular, we will suppose that $H \neq H_i$, and consider each possibility for H_{i+1} . First, since $H_0 = K_2 \in \mathcal{F}_3$, the

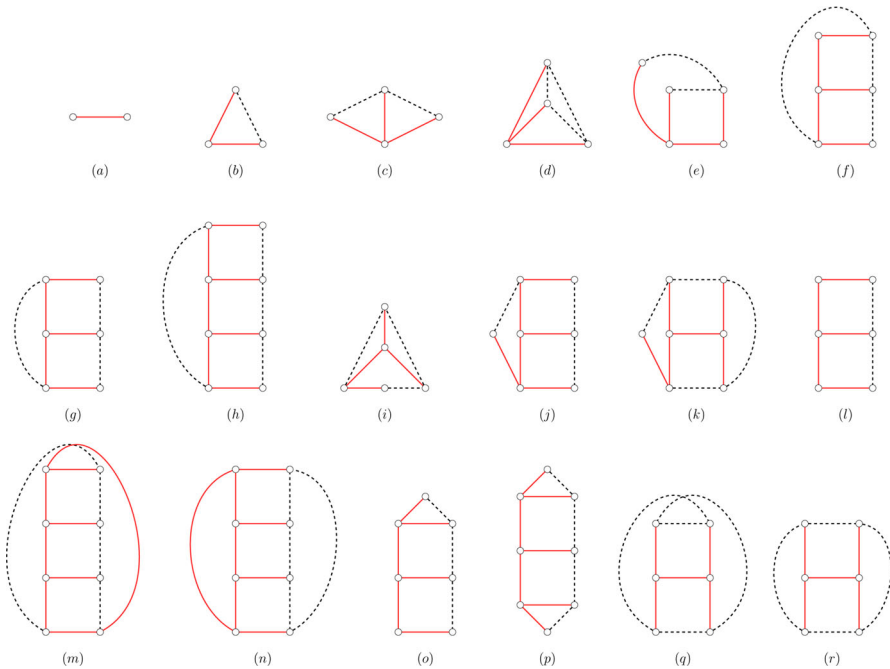


Fig. 5 In red are depicted minimum 3-spanners of some representative graphs in \mathcal{F}_3 . The graphs depicted in **l**, **m** and **n** are L_2 , M_3 and N_3 , respectively. The graphs depicted in **o**, **p**, **q** and **r** are T_2^1 , T_2^2 , M_2 and N_2 , respectively (Color figure online)

graph H_1 is the result of applying operation (2) or operation (3) to H_0 . We distinguish these two cases.

Case 1: $H_1 \cong K_3$

Note that we cannot apply operations (1) or (4) to H_1 . Thus, H_2 is obtained by applying either operation (2) or operation (3) to H_1 . In the first case, $H_2 \cong K_4 - e$; and in the second case, $H_2 \cong T_1^1$.

Case 1.1: $H_2 \cong K_4 - e$

In this case, H_2 contains only two vertices of degree two. Since these vertices are not adjacent, the graph H_3 is obtained by applying operations (1) or (2) to H_2 . Thus, either $H_3 \cong K_4$ or $H_3 \cong G_1$ (see Fig. 4). Since, in both cases H_3 has at most one vertex of degree two, we have that $H = H_3 \in \mathcal{F}_3$.

Case 1.2: $H_2 \cong T_1^1$

Let a , b and c be the vertices of degree two in H_2 such that the neighbors of a have degree three (see Fig. 6 a). Note that H_3 is obtained by applying (1), (2) or (3) to H_2 . In case we apply operation (1), H_3 is obtained by adding the edge ab or ac to H_2 . In both cases, we have that $H_3 \cong G_1$ (see Fig. 6b). Moreover, we have that $H = H_3 \in \mathcal{F}_3$, since it has only one vertex of degree two.

Suppose now that H_3 is obtained by applying operation (2) to H_2 . We can apply this operation on vertices a and b (or c), or on vertices b and c . In the first case, we have that $H_3 \cong N_2 - e$ (see Fig. 6c). In the second case, we have that $H_3 \cong T_1^2$. Finally, we

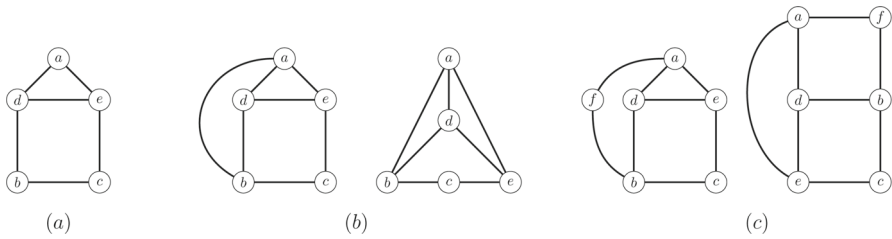


Fig. 6 **a** $H_2 \cong T_1^1$; **b** $H_3 \cong G_1$; and **c** $H_3 \cong N_2 - e$

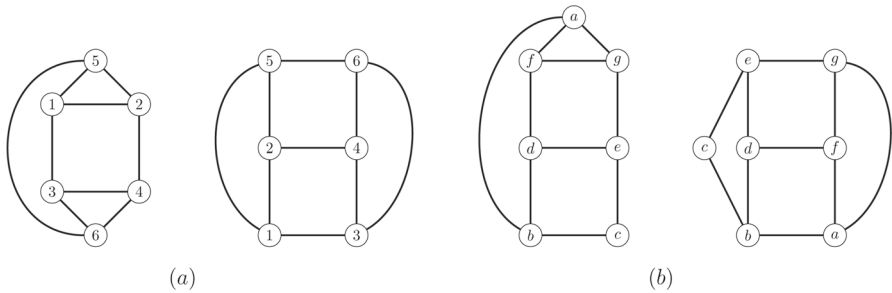


Fig. 7 **a** $H_3 \cong N_2$; and **b** $H_3 \cong G_3$

can only apply operation (3) on vertices b and c . In this case, $H_3 \cong T_2^1$. We analyse these three cases for H_3 .

Case 1.2.1: $H_3 \cong N_2 - e$

In this case, H_3 contains only two vertices of degree two. Note that, we can only apply operation (1) or (2) to obtain H_4 . In the first case, we have $H_4 \cong N_2$, and in the second case, $H_4 \cong G_3$. In either case H_4 has at most one vertex of degree two, thus $H = H_4 \in \mathcal{F}_3$.

Case 1.2.2: $H_3 \cong T_1^2$

As in the previous case, H_3 has only two vertices of degree two. By distance constraints, we can only apply operation (1) to H_3 on those vertices. Therefore, $H_4 \cong N_2$ (see Fig. 7a). As N_2 is cubic, we have that $H = H_4 \in \mathcal{F}_3$.

Case 1.2.3: $H_3 \cong T_2^1$

Let a, b and c be the three vertices of degree two in T_2^1 such that b is adjacent to c . Note that, we can apply operations (1), (2) or (3) to H_3 . First, suppose that we apply operation (1) to H_3 . By distance constraints, we can only apply this operation on vertices a and b (or c). In either case, we have that $H_4 \cong G_3$ (see Fig. 7b). As G_3 has only one vertex of degree two, we have that $H = H_4 \in \mathcal{F}_3$.

Suppose now that we apply (2) to H_3 . In this case, we have to apply this operation on vertices b and c and, thus $H_4 \cong T_2^2$. Observe that, we cannot apply any operation to T_2^2 . Therefore, $H = H_4 \in \mathcal{F}_3$.

Finally, suppose that we apply operation (3) to H_3 . We can apply this operation only on vertices b and c ; therefore, $H_4 \cong T_3^1$. Furthermore, we can only apply operations (2) and (3) to T_3^1 . Thus, by analogous arguments as before, $H \cong T_k^2$, for $k \geq 3$.

Case 2: $H_1 \cong L_1$

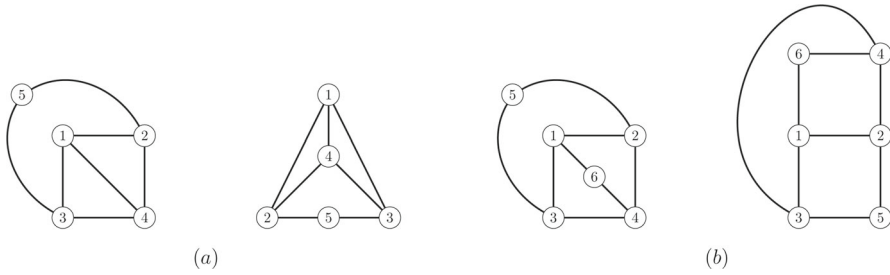


Fig. 8 **a** $H_3 \cong G_1$; and **b** $H_3 \cong M_2 - e$

In this case, we can apply any of the four operations to H_1 . First observe the following

- (i) if we apply operation (1), we have that $H_2 \cong K_4 - e$ (as in Case 1.1.).
- (ii) if we apply operation (2) on adjacent vertices, we have that $H_2 \cong T_1^1$ (as in Case 1.2.).
- (iii) if we apply operation (4), then $H_2 \cong K_4$ (as H_2 is cubic, we have that $H = H_2$).

Since we have already considered those cases, let us suppose that H_2 was obtained by applying (2) on nonadjacent vertices, or by applying (3) to H_1 . In the first case, we have $H_2 \cong K_{2,3}$, and in the second case $H_2 \cong L_2$. We distinguish these two cases.

Case 2.1: $H_2 \cong K_{2,3}$

In this case, we can only apply operation (1) or (2) to H_2 . If we apply (1), we have that $H_3 \cong G_1$ (see Fig. 8a). Since G_1 has just one vertex of degree two, we have that $H = H_3$. Otherwise, suppose that H_3 was obtained by applying (2). In this case, we have that $H_3 \cong M_2 - e$. (see Fig. 8b). Observe that, if $H \neq H_3$, then we have that $H \cong M_2$.

Case 2.2: $H_2 \cong L_2$

In this case, we can apply any of the four operations to H_2 . First, if we apply (1), we have that $H_3 \cong M_2 - e$ or $H_3 \cong N_2 - e$. If $H \neq H_3$, then $H \in \{M_2, N_2, G_3\}$. Next, if we apply operation (4) to H_2 , we have that $H \cong M_2$ or $H \cong N_2$.

Suppose now that we apply operation (2) to H_2 . In this case, if we choose two adjacent vertices, we have $H_3 \cong T_2^1$ (as in Case 1.2.3). Otherwise, we have $H_3 \cong G_2$. Finally, if we apply operation (3), we have that $H_3 \cong L_3$. We distinguish these last two cases.

Case 2.2.1: $H_3 \cong G_2$

Consider that x, y and z are the vertices of degree two, in H_3 , as in Fig. 9a. Observe that, we can only apply (1) or (2) to H_3 . First, suppose that we apply operation (1) to obtain H_4 . Observe that

- if we apply (1) on the vertices y and z , we have $H_4 \cong G_3$.
- Moreover, if we choose vertices x and y (or z), we also obtain $H_4 \cong G_3$ (see Fig. 9b).

Since G_3 has only one vertex of degree two, we have that $H = H_4 \in \mathcal{F}_3$.

Suppose that we apply (2) to H_3 . Observe that, if we apply this operation either on x and y , or on y and z , we have $H_4 \cong N_3 - e$. Both cases are depicted in Fig. 10. In case $H \neq H_3$, we have that $H \cong N_3 \in \mathcal{F}_3$.

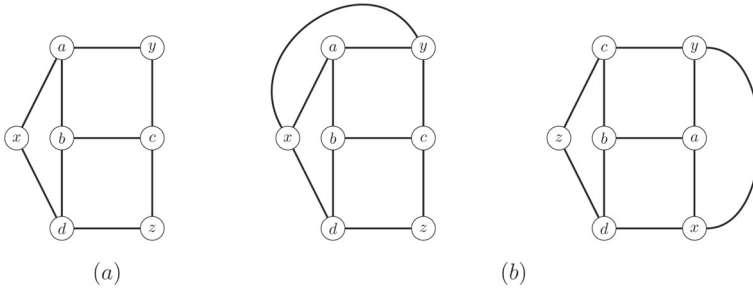


Fig. 9 a $H_3 \cong G_2$; and b $H_4 \cong G_3$

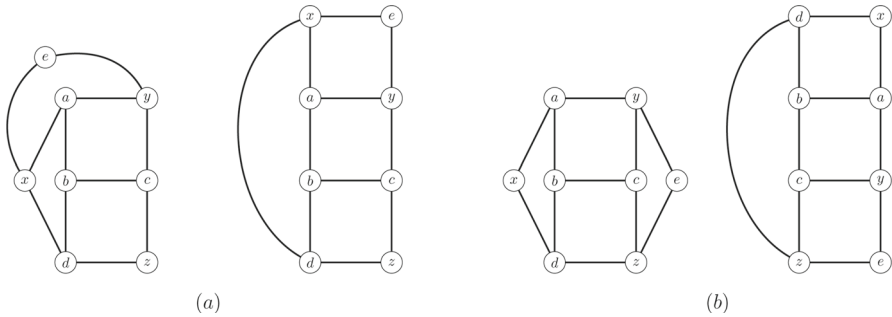


Fig. 10 Adding a 2-path H_3 between a x and y ; and b y and z

Case 2.2.2: $H_3 \cong L_3$

In this case, we can apply any of the four operations to H_3 . First, suppose that we apply (1). Observe that, we can only apply this operation on vertices that are at distance three in L_3 . Thus, $H_4 \cong N_3 - e$. Moreover, if $H \neq H_4$, then $H \cong N_3 \in \mathcal{F}_3$.

Suppose now that we apply operation (4) to H_3 . Then, either $H_4 \cong M_3$ or $H_4 \cong N_3$. Since M_3 and N_3 are cubic, we have that $H = H_4$.

Finally, suppose that we apply (2) or (3) to H_3 . Note that, we can only apply these operations on adjacent vertices of degree two. Thus, we have that $H_4 \cong T_3^1$ or $H_4 \cong L_4$. We distinguish these last two cases.

Case 2.2.2.1: $H_4 \cong T_3^1$

By the same arguments given at the end of Case 1.2.3, we have that $H \cong T_k^2 \in \mathcal{F}_3$, for $k \geq 3$.

Case 2.2.2.2: $H_4 \cong L_4$

First, observe that we cannot apply (1) to L_k , for $k \geq 4$. If we apply (2) to H_4 , we have that $H_4 \cong T_4^1$. By using arguments similar to the previous case, we have that $H \cong T_k^2$, for $k \geq 4$. If we apply (4) to H_4 , we have that $H \cong M_4$ or $H \cong N_4$. To conclude, observe that, if we apply (3) to H_4 , we obtain $H_5 \cong L_5$. Since this is analogous to the case in which $H_4 \cong L_4$, we have that $H \cong T_k^2$, $H \cong M_k$ or $H \cong N_k$, for $k \geq 5$.

Proposition 2 and Theorem 4 reduce MINS_3 on subcubic graphs to the problem of finding a minimum 3-spanner for each subgraph in \mathcal{F}_3 . Figure 5 shows a minimum 3-spanner (in solid edges) for each of these subgraphs.

We observe that most of the graphs in \mathcal{F}_3 admit a tree 3-spanner (which is clearly a minimum 3-spanner). The only exceptions are M_k and N_k , for $k \geq 3$. In what follows, we show how to construct a tree 3-spanner of L_k . We will use this construction later to obtain minimum 3-spanners for M_k and N_k .

Let x_1, y_1, x_2 and y_2 be the vertices of degree two in L_k , such that x_i is adjacent to y_i , for $i = 1, 2$. Moreover, let P be the k -path between x_1 and x_2 in L_k . Finally, let S be the tree obtained from P by linking every vertex $v \in V(L_k) \setminus V(P)$ to its neighbor in P . Since the ends of any edge in L_k are at distance at most three, in S , this is a tree 3-spanner of L_k . We show an example of this construction in Fig. 5l. Observe that, we can obtain a tree 3-spanner for T_k^1 and T_k^2 in a similar way. For this, we add the edges that link each vertex in $V(T_k^i) \setminus V(L_k)$ to P . This construction is depicted in Fig. 5o and p.

We show next how to obtain a 3-spanner of M_k and N_k , for $k \geq 3$, with $|V(L_k)|$ edges. Recall that M_k and N_k arise from L_k . After that, we show that this construction is optimal. Consider the tree 3-spanner S of L_k that was constructed above. Note that, if we add the edge x_1y_2 (resp. x_1x_2) to S , we obtain a 3-spanner of M_k (resp. N_k), since the distance between the vertices x_2 and y_1 (resp. y_1 and y_2) in the resulting graph is three. In Fig. 5m and n, we show an example for the case $k = 3$. Also, we note that for the case $k = 2$, the graphs M_2 and N_2 admit a tree 3-spanner (see Fig. 5q and r).

In what follows, we show that any minimum 3-spanner of M_k and N_k , $k \geq 3$, needs at least $|V(L_k)|$ edges. First, as N_k is the Cartesian product of a $(k + 1)$ -cycle with K_2 , a result obtained by Lin and Lin (2020) implies that N_k does not admit a tree 3-spanner for $k \geq 3$, and therefore any 3-spanner of N_k must contain at least $|V(L_k)|$ edges. It remains now to prove that M_k does not admit a tree 3-spanner when $k \geq 3$.

Lemma 5 *If $k \geq 3$, then the graph M_k does not admit a tree 3-spanner.*

Proof Take $k \geq 3$. We consider that L_k is a subgraph of M_k . In particular, $V(L_k) = V(M_k)$. Let x_1, x_2, y_1 and y_2 be the vertices of degree two, in L_k , such that x_1y_1 and x_2y_2 are edges in L_k . Suppose by contradiction that M_k admits a tree 3-spanner, S . Observe that $E(S) \cap \{x_1y_2, y_1x_2\} \neq \emptyset$. Otherwise, S is a subgraph of L_k . This implies that the distance between x_1 and y_2 in S is at least $k + 1$, a contradiction, since $k \geq 3$.

By symmetry, let us suppose that the edge x_1y_2 belongs to $E(S)$, and consider the forest $S' = S - x_1y_2$. In what follows, we show that every pair of adjacent vertices in $L' = L_k - \{x_1y_1, x_2y_2\}$ belongs to the same connected component in S' . Since L' is a spanning subgraph of M_k , this implies that S' is also connected which is a contradiction.

Let $uv \in E(L')$. Suppose by contradiction that the path P linking u and v in S contains the edge x_1y_2 . Now, consider the cycle $C = P + uv$. Since x_1y_2 belongs to P , then $P' = C - x_1y_2$ is a path between x_1 and y_2 in S such that $|P| = |P'|$. To conclude, we consider two cases. If P' is contained in L_k , then $|P| = |P'| \geq k + 1 \geq 4$, a contradiction. Otherwise, P' contains the edge x_2y_1 . But, this implies that P contains

both x_1y_2 and x_2y_1 . Since $uv \neq x_1y_1$ and $uv \neq x_2y_2$, we have that $|P| \geq 4$ which contradicts the fact that S is a 3-spanner of M_k .

Therefore, $V(L') = V(M_k)$ induces a connected component in S' . But, this implies that S contains a cycle, a contradiction.

Since the 3-spanners of M_k and N_k , $k \geq 3$, that we constructed previously use $|V(L_k)|$ edges, we have the following result.

Corollary 6 *Let S^* be a minimum 3-spanner of M_k (or N_k). Then S^* has $|V(L_k)|$ edges, for $k \geq 3$.*

We are now ready to show the main result of this section.

Theorem 7 *MINS₃ can be solved in polynomial time on subcubic graphs.*

Proof Let $G = (V, E)$ be a subcubic graph. First, we find $\mathcal{C}_3(G)$. We have shown how to find a minimum 3-spanner for each graph in \mathcal{F}_3 . Thus, given $H \in \mathcal{C}_3(G)$, we just need to recognize which graph, in \mathcal{F}_3 , it is isomorphic to. For the case $|V(H)| \leq 9$, this is done by a brute-force algorithm. Suppose now that $|V(H)| \geq 10$. In this case, the only graphs left are L_k, T_k^1, T_k^2, M_k and N_k , for $k \geq 4$. Let d be the number of vertices of degree two in H . If $d = 4$, then $H \cong L_k$; if $d = 3$, then $H \cong T_k^1$; if $d = 2$, then $H \cong T_k^2$. Finally, if $d = 0$, then $H \cong M_k$ or $H \cong N_k$.

We distinguish between M_k and N_k as follows. Let E' be the set of edges in H that belong to just one 4-cycle. In the case $H \cong M_k$, the set E' induces a Hamiltonian cycle of H . Otherwise, E' induces two disjoint $(k + 1)$ -cycles.

To conclude this section, we comment on the implication Theorem 7 has for TREES₃. Fomin et al. (2011) showed that TREES _{t} can be solved in polynomial time on the class of bounded-degree graphs. For this, they showed that if a graph G has a tree t -spanner, then its treewidth is at most $\Delta(G)^t$. Thus, if G has maximum degree d , then TREES _{t} can be solved in linear time, as follows:

- (1) check whether G has treewidth at most d^t ; and
- (2) look for a tree t -spanner if (1) holds.

Bodlaender (1996) showed how to test (1) in linear time. Step (2) can be solved in linear time via Courcelle's theorem (Courcelle and Engelfriet 2012), since G has bounded treewidth (by (1)) and the property of admitting a tree t -spanner is expressible in monadic second order logic (Fomin et al. 2011). Although, this algorithm has the best time complexity, it is quite inefficient in practice. The algorithm that tests (1) has a very large constant factor, even for checking whether a graph has treewidth at most 4. In our case, $d^t = 9$, so its constant factor is too large for practical purposes.

Recently, Papoutsakis (2018) showed a dynamic programming algorithm that solves TREES _{t} in polynomial time for fixed t and maximum degree. Each state of this dynamic programming saves a complete subgraph of the input graph. This makes its implementation quite involved and also memory inefficient.

We observe that, by Proposition 2, our approach on MINS₃ gives an alternative simple algorithm for TREES₃ on subcubic graphs. First, we find a minimum 3-spanner for each $H \in \mathcal{C}_3(G)$. Let \mathcal{T} be the union of these minimum 3-spanners. Then, G admits a 3-tree spanner if and only if \mathcal{T} is a tree.

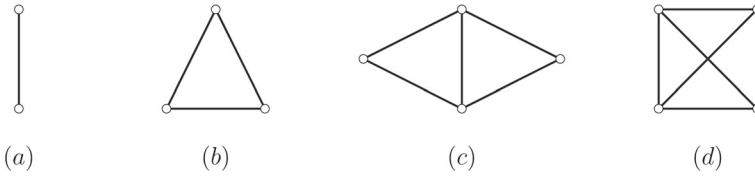


Fig. 11 a K_2 ; b K_3 ; c $K_4 - e$; and d K_4

4 Polytope of the TREE 2-SPANNERS

Throughout this section $G = (V, E)$ denotes a connected subcubic graph. Let $F \subseteq E$. The vector $\chi^F \in \mathbb{R}^E$ denotes the *incidence vector* of the set F , in other words, the binary vector whose nonzero entries correspond to the elements in F . For simplicity, if H is a subgraph of G , we write χ^H instead of $\chi^{E(H)}$. This section is devoted to the study of the following polytope:

$$\mathcal{T}_2(G) := \text{conv}(\{\chi^T \in \mathbb{R}^E : T \text{ is a tree 2-spanner of } G\}),$$

where $\text{conv}(X)$ denotes the convex hull of the vectors in X . Our aim is to find a set of linear inequalities that define $\mathcal{T}_2(G)$. By Proposition 2, we only need to describe the set $\mathcal{T}_2(H)$, for each $H \in \mathcal{C}_2(G)$, and combine those inequalities. The set $\mathcal{C}_2(G)$ was characterized by Cai and Keil (1994) for graphs of maximum degree 4. When restricted to subcubic graphs, their characterization yields the following result.

Lemma 8 (Cai and Keil 1994) *Let G be a graph such that $\Delta(G) \leq 3$. If $H \in \mathcal{C}_2(G)$, then H is isomorphic to $K_2, K_3, K_4 - e$, or K_4 .*

The graphs in $\mathcal{C}_2(G)$ are shown in Fig. 11. Our formulation consists mainly of two sets of inequalities that are based on the following observations. In case H is a complete graph, then any tree 2-spanner of H has diameter at most two. Thus, such tree is a star, so any tree 2-spanner has no matching of size two. Consider now a graph G that contains a 4-cycle C . Note that, no tree 2-spanner of G contains three edges in G . Otherwise, such edges induce a 3-path between two vertices of C , which violates the 2-spanner condition.

In what follows, we present our formulation. For this, consider the decision variables $x \in \mathbb{R}^E$ such that $x_e = 1$ if and only if e belongs to the solution. Let $P(G)$ be the polytope defined by the following set of inequalities.

$$\begin{aligned} x(E(G)) &= |V(G)| - 1, \\ x(E(H)) &= |V(H)| - 1, \quad \forall H \in \mathcal{C}_2(G), \\ x(F) &\leq 1, \quad \forall F \subseteq E(H), F \text{ matching}, H \in \mathcal{C}_2(G), H \text{ clique}, \\ x(C) &\leq 2, \quad \forall C \subseteq E(H), C \text{ is a 4-cycle}, H \in \mathcal{C}_2(G), \\ x_e &\leq 1, \quad \forall e \in E, \\ x_e &\geq 0, \quad \forall e \in E. \end{aligned}$$

We show now the main result of this section.

Theorem 9 *Let $G = (V, E)$ be a connected subcubic graph. Then*

$$\mathcal{T}_2(G) = P(G).$$

Proof First, we show that $\mathcal{T}_2(G) \subseteq P(G)$. Let T be a tree 2-spanner of G , and let H be a subgraph in $\mathcal{C}_2(G)$. Consider that $T^H = H \cap T$. By Proposition 2, T^H is a tree 2-spanner of H . Since T^H is a tree, we have that $\chi^{T^H}(E(H)) = |V(H)| - 1$. Now, let C be a 4-cycle in H . Observe that $\chi^{T^H}(C) \leq 3$. We will show that $\chi^{T^H}(C) \neq 3$. Suppose by contradiction that $\chi^{T^H}(C) = 3$, and let $uv \in E(C)$ be the unique edge that is not in the support of χ^{T^H} . Since T is a tree, the unique path that links u and v , in T , is precisely $C - uv$. But, this implies that $d_T(u, v) = 3$, which contradicts the fact that T is a 2-spanner of G . Thus, $\chi^{T^H}(C) \leq 2$. Finally, if H is a clique, then T^H has diameter at most two, so it is a star. As all the edges in a star share a common vertex, then $\chi^{T^H}(F) \leq 1$, for any matching F in H . Therefore, we have that $\mathcal{T}_2(G) \subseteq P(G)$.

We show now that $P(G) \subseteq \mathcal{T}_2(G)$. For this, it suffices to show that every vertex of $P(G)$ has integer coordinates. Let x^* be a vertex of $P(G)$. We say that an edge $e \in E$ is *fractional* if $0 < x_e^* < 1$. Moreover, we say that an edge e is *full* if $x_e = 1$. Let $e \in E$, and let $H \in \mathcal{C}_2(G)$ be the subgraph that contains e . To show that $x_e^* = 0$ or $x_e^* = 1$, we distinguish four cases.

Case 1: $H \cong K_2$

In this case, $E(H) = \{e\}$. Then, $x^*(E(H)) = x_e^* = 1$.

Case 2: $H \cong K_3$

By contradiction, suppose that e is fractional. Since $x^*(E(H)) = 2$, we have that $x^*(E(H)) - x_e^* > 1$. This implies that there is another fractional edge $f \in E(H)$. Let

$$\varepsilon = \min\{x_e^*, x_f^*, 1 - x_e^*, 1 - x_f^*\}.$$

Consider the vectors x^1 and x^2 defined as follows:

$$x_k^1 = \begin{cases} x_e^* - \varepsilon, & \text{if } k = e, \\ x_f^* + \varepsilon, & \text{if } k = f, \\ x_k^*, & \text{otherwise.} \end{cases} \quad x_k^2 = \begin{cases} x_e^* + \varepsilon, & \text{if } k = e, \\ x_f^* - \varepsilon, & \text{if } k = f, \\ x_k^*, & \text{otherwise.} \end{cases}$$

Note that $x^1(E(H)) = x^2(E(H)) = 2$. Furthermore, by the definition of ε , all the other inequalities are satisfied. Thus x^1 and x^2 belong to $P(G)$. But this contradicts the fact that x^* is a vertex, since $x^* = \frac{1}{2}(x^1 + x^2)$.

Case 3: $H \cong K_4 - e$

Let C be the 4-cycle in H . First, we show the following claim.

Claim 10 *If C contains a fractional edge, then $x^*(C) < 2$.*

Proof of Claim First, we will show that C has at most one fractional edge. Suppose by contradiction that there exist edges $f, g \in E(C)$ such that $0 < x_f^*, x_g^* < 1$. Let

$$\varepsilon = \min\{x_f^*, x_g^*, 1 - x_f^*, 1 - x_g^*\}.$$

Define now vectors x^1 and x^2 as in Case 2. Then, $x^1(E(H)) = x^2(E(H)) = x^*(E(H))$ and $x^1(C) = x^2(C) = x^*(C)$. Thus, x^1 and x^2 belong to $P(G)$. But this contradicts the fact that x^* is a vertex.

Therefore, the vector x^* restricted to C contains at most one fractional edge. Now suppose that C contains a fractional edge, say f . Since $x^*(C) \leq 2$, there is at most one full edge in $E(C) - f$. Therefore, $x^*(C) < 2$.

Since $E(H) \setminus E(C)$ contains a unique edge, the previous claim implies that C has no fractional edge, otherwise we would have $x^*(E(H)) < 3$, a contradiction. Finally, since $x^*(E(H)) = 3$ and $x^*(C) \leq 2$, the unique edge in $E(H) \setminus E(C)$ is also integral.

Case 4: $H \cong K_4$

Let f be an edge of H . We will denote by f' the unique edge in H such that $\{f, f'\}$ is a matching. In this case, we first show the following claim.

Claim 11 *If x_f^* is fractional, then $x_{f'}^* = 0$.*

Proof of Claim If x_f^* is fractional, then $x_{f'}^* < 1$. Suppose by contradiction that $x_{f'}^* > 0$, and let $\varepsilon = \min\{x_f^*, x_{f'}^*, 1 - x_f^*, 1 - x_{f'}^*\}$. Then, if we define vectors x^1, x^2 as in Case 2, we have that $x^*(E(H)) = x^1(E(H)) = x^2(E(H))$, and also, for any 4-cycle C , we have $x^*(C) = x^1(C) = x^2(C)$. Finally, by the way we defined x^1 and x^2 , we have that $x^1(F) \leq 1$ and $x^2(F) \leq 1$, for any matching F in H . Therefore, both vectors x^1 and x^2 belong to $P(G)$, a contradiction.

Consider that $E(H) = \{e, e', f, f', g, g'\}$. As $x^* \in P(G)$, we have that $x_e^* + x_{e'}^* \leq 1$, $x_f^* + x_{f'}^* \leq 1$, and $x_g^* + x_{g'}^* \leq 1$. Since $x^*(E(H)) = 3$, all the previous inequalities must be satisfied with equality. Therefore, the above claim implies that H has no fractional edge.

Since each graph in $\mathcal{C}_2(G)$ has constant size, the polytope $P(G)$ consists of $\mathcal{O}(|E|)$ inequalities and, therefore, we can find an optimal solution of $P(G)$ in polynomial time on the size of the input graph G (Dantzig and Thapa 2003).

Suppose now that G has costs $c_e \in \mathbb{R}$, $e \in E$, assigned to its edges. Let us consider the *minimum cost tree t -spanner problem* (MCTS_t) that seeks a tree t -spanner of minimum total cost. We observe that, when measuring the distance between two vertices u and v , we disregard the costs of the edges. So, $d_G(u, v)$ is the minimum length of a path linking u and v in G .

Theorem 9 implies that an optimal solution of $\min\{cx : x \in P(G)\}$ induces a tree 2-spanner of G that is an optimal solution for MCTS_2 . Therefore, we obtain the following result.

Theorem 12 *MCTS_2 can be solved in polynomial time on subcubic graphs.*

5 Concluding remarks

We showed a polynomial-time algorithm for MINS_3 on subcubic graphs. This result answers partially an open question regarding the complexity of MINS_t on bounded-degree graphs. This algorithm also yields an alternative algorithm for TREES_3 on this

class of graphs. Gómez et al. (2023) have proved recently that MINS_3 on (planar) graphs of maximum degree at least 5 is an NP-hard problem. Thus, for bounded-degree graphs, it remains only to establish the complexity of MINS_3 on graphs with maximum degree 4. Such a result would solve the unique open question for $t = 3$ (see Table 1). We consider this a challenging and interesting problem.

We also investigated TREES_t from a polyhedral point-of-view. In particular, we focused on the incidence vectors of the tree 2-spanners of a subcubic graph, and studied the polytope defined by the convex hull of these vectors. We showed a complete linear description of this polytope (of polynomial size). As a byproduct, we obtained a polynomial-time algorithm for MCTS_2 on subcubic graphs. As far as we know, this is a novel approach and result for this problem. It was motivated by the results obtained by Álvarez-Miranda and Sinnl (2019), and by Ahmed et al. (2019) on (mixed) integer linear formulations for MINS_t (and its variants). They are able to solve only small instances in a reasonable amount of time. Thus, finding strong and tight inequalities for the relaxed formulations may lead to approaches with better performance. However, it seems to be a hard and challenging problem.

Acknowledgements The authors would like to thank the referees for the useful remarks.

Author Contributions All authors contributed equally to the study conception and design. All authors read and approved the final manuscript.

Funding This research has been partially supported by FAPESP - São Paulo Research Foundation (Proc. 2015/11937-9) and by CNPq (Proc. 404315/2023-2). R. Gómez is supported by FAPESP (Proc. 2019/14471-1); F.K. Miyazawa is supported by FAPESP (Proc. 2022/05803-3) and by CNPq (Proc. 313146/2022-5); Y. Wakabayashi is supported by CNPq (Proc. 311892/2021-3), Brazil.

Data Availability Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare that they have no conflict of interest. The authors have no relevant financial or non-financial interests to disclose.

References

- Ahmed R, Bodwin G, Sahneh F, Hamm K, Latifi M, Kobourov S, Spence R (2020) Graph spanners: a tutorial review. *Comput Sci Rev* 37:100253
- Ahmed R, Hamm K, Latifi Jebelli M, Kobourov S, Sahneh F, Spence R (2019) Approximation algorithms and an integer program for multi-level graph spanners. In: *Analysis of experimental algorithms, lecture notes in computer science*, vol. 11544, Springer, Cham, pp 541–562
- Álvarez-Miranda E, Sinnl M (2019) Mixed-integer programming approaches for the tree t^* -spanner problem. *Optim Lett* 13(7):1693–1709
- Awerbuch B (1985) Complexity of network synchronization. *J ACM* 32(4):804–823
- Baswana S, Sen S (2006) Approximate distance oracles for unweighted graphs in expected $O(n^2)$ time. *ACM Trans Algorithms* 2(4):557–577
- Bodlaender H (1996) A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J Comput* 25(6):1305–1317
- Brandstädt A, Dragan F, Le H, Le V, Uehara R (2007) Tree spanners for bipartite graphs and probe interval graphs. *Algorithmica* 47(1):27–51

- Cai L (1994) NP-completeness of minimum spanner problems. *Discrete Appl Math* 48(2):187–194
- Cai L, Cornel D (1995) Tree spanners. *SIAM J Discrete Math* 8(3):359–387
- Cai L, Keil M (1994) Spanners in graphs of bounded degree. *Networks* 24(4):233–249
- Courcelle B, Engelfriet J (2012) Graph structure and monadic second-order logic, *Encyclopedia of Mathematics and its Applications*, vol 138. Cambridge University Press, Cambridge
- Couto F, Cunha L (2021) Hardness and efficiency on t -admissibility for graph operations. *Discrete Appl Math* 304:342–348
- Couto F, Cunha L, Posner D (2021) Edge tree spanners, graphs and combinatorial optimization: from theory to applications: CTW2020 Proceedings. Springer, Cham, pp 195–207
- Dantzig G, Thapa M (2003) *Linear programming 2: theory and extensions*. Springer, New York
- Dragan F, Fomin F, Golovach P (2011) Spanners in sparse graphs. *J Comput Syst Sci* 77(6):1108–1119
- Dragan F, Köhler E (2014) An approximation algorithm for the tree t -spanner problem on unweighted graphs via generalized chordal graphs. *Algorithmica* 69(4):884–905
- Fekete S, Kremer J (2001) Tree spanners in planar graphs. *Discrete Appl Math* 108(1–2):85–103
- Fomin F, Golovach P, van Leeuwen E (2011) Spanners of bounded degree graphs. *Inform Process Lett* 111(3):142–144
- Gómez R, Miyazawa FK, Wakabayashi Y (2022) Minimum t -spanners on subcubic graphs, WALCOM: algorithms and computation. In: *Lecture notes in computer science*, vol. 13174, Springer, Cham, pp 365–380
- Gómez R, Miyazawa FK, Wakabayashi Y (2023) Improved NP-hardness results for the minimum t -spanner problem on bounded-degree graphs. *Theor Comput Sci* 947:113691
- Gómez R, Miyazawa FK, Wakabayashi Y (2022) Tree 3-spanners on generalized prisms of graphs, LATIN, theoretical informatics. In: *Lecture notes in computer science*, vol. 13568, Springer, Cham, pp 557–573
- Kobayashi Y (2018) NP-hardness and fixed-parameter tractability of the minimum spanner problem. *Theor Comput Sci* 746:88–97
- Lin L, Lin Y (2020) Optimality computation of the minimum stretch spanning tree problem. *Appl Math Comput* 386:125502
- Papoutsakis I (2018) Tree spanners of bounded degree graphs. *Discrete Appl Math* 236:395–407
- Peleg D, Schäffer A (1989) Graph spanners. *J Graph Theory* 13(1):99–116
- Peleg D, Ullman J (1989) An optimal synchronizer for the hypercube. *SIAM J Comput* 18(4):740–747
- Peleg D, Upfal E (1989) A trade-off between space and efficiency for routing tables. *J ACM* 36(3):510–530
- Thorup M, Zwick U (2005) Approximate distance oracles. *J ACM* 52(1):1–24
- Venkatesan G, Rotics U, Madanlal M, Makowsky J, Pandu Rangan C (1997) Restrictions of minimum spanner problems. *Inform Comput* 136(2):143–164
- Wang W, Balkcom D, Chakrabarti A (2015) A fast online spanner for roadmap construction. *Int J Rob Res* 34(11):1418–1432

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.