Journal of Combinatorial Optimization (2021) 41:603–624
https://doi.org/10.1007/s10878-021-00698-1

Minimum constellation covers: hardness, approximability and polynomial cases

Santiago Valdés Ravelo[1]

Accepted: 9 January 2021 / Published online: 30 January 2021

Abstract
This paper considers two graph covering problems, the Minimum Constellation Cover (CC) and the Minimum k-Split Constellation Cover (K- SCC). The input of these problems consists on a graph $G = (V, E)$ and a set $\mathcal{C}$ of stars of G, and the output is a minimum cardinality set of stars C, such that any two different stars of C are edge-disjoint and the union of the stars of C covers all edges of G. For CC, the set C must be compound by edges of G or stars of $\mathcal{C}$ while, for K- SCC, an integer k is given and the elements of C must be k-stars obtained by splitting stars of $\mathcal{C}$. This work proves that unless $P = NP$, CC does not admit polynomial time $|\mathcal{C}|^{\mathcal{O}(1)}$-approximation algorithms and K- SCC cannot be $((1-\epsilon)\ln|E|)$-approximated in polynomial time, for any $\epsilon > 0$. Additionally, approximation ratios are given for the worst feasible solutions of the problems and, for K- SCC, polynomial time approximation algorithms are proposed, achieving a $(\ln|E| - \ln\ln|E| + \Theta(1))$ approximation ratio. Also, polynomial time algorithms are presented for special classes of graphs that include bounded degree trees and cacti.

Keywords Exact graph cover · NP-hard · Inapproximability · Approximation algorithm · Polynomial time algorithm

1 Introduction

Graph coverings include several classical problems of graph theory and have been widely studied by the computer science community (Dinneen and Hua 2017; Fernandes et al. 2009; Fukunaga 2016; Hung and Chang 2007; Álvarez Miranda and Sinnl 2020; Pilipczuk et al. 2020; Rizzi et al. 2014). These problems find practical applications in diverse network scenarios, they may help to design monitoring systems (Marques et al. 2019), and to divide the network in modular information processing

✉ Santiago Valdés Ravelo
santiago.ravelo@inf.ufrgs.br

[1] Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, Brazil

units (Wegner 2014), among other functionalities. Some versions of these problems consider to minimize exact edge coverings, where the graph must be partitioned in a minimum number of subgraphs such that each edge is covered by exactly one subgraph. If the subgraphs must be stars selected from a previously given set, then the problem is the Minimum Constellation Cover (CC). If the subgraphs must be k-stars obtained by splitting the elements of a previously given set of stars, then the problem is the Minimum k-Split Constellation Cover (K- SCC).

This work introduces CC and K- SCC, proving that both problems are NP-hard and providing strong inapproximability results for them, even when the network is a star. The approximability of the problems is analyzed and approximation ratios are given for the worst feasible solution of each instance. Also, for K- SCC, polynomial time approximation algorithms are given, achieving an approximation ratio very close the inapproximability bound of the problem. Finally, a polynomial time algorithm is proposed to solve instances where the network is a bounded degree class of graphs that generalizes trees and cacti.

The rest of this paper is organized as follows. Section 2 gives some notations and introduces the problems definitions. Section 3 proves strong NP-hard and inapproximability results for both problems. Section 4 discusses on the approximability of the problems, finding approximation ratios associated with the worst solutions and, for K- SCC, giving different polynomial time approximation algorithms. Section 5 proposes a polynomial time algorithm for a special class of graphs that includes paths, cycles, trees, cacti, among others. The conclusions and future research directions are given in Sect. 6.

2 Definitions and notation

This work considers simple undirected graphs, where a **graph** $G = (V, E)$ is defined by a set of nodes V and a set of edges E, being each edge a pair of different nodes. Every edge $\langle u, v\rangle \in E$ is said to be incident on the nodes $u, v \in V$ and the number of edges incident over a node u is the **degree** of the node ($\delta(u)$). Given a graph G, its set of nodes can be denoted by V_G and its set of edges by E_G. Some graphs relations used along this document are the subgraph and induced subgraph concepts.

Definition 1 Given two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$, H is **subgraph** of G (denoted as $H \subseteq G$) iff $V_H \subseteq V_G$ and $E_H \subseteq E_G$. If $H \subseteq G$ and every edge in E_G defined between two nodes of V_H is contained in E_H, then H is the **induced subgraph** of G over the set of nodes V_H (denoted as $G[V_H]$).

To describe the special classes of graphs that will be studied, the definitions of paths, cycles and connectivity are needed.

Definition 2 Given a graph $G = (V, E)$, a **path** in G is a sequence of nodes $v_1, v_2, \ldots, v_n$ where, for all $1 \leq i < j \leq n$, $v_i, v_j \in V_G$, $v_i \neq v_j$ and $\langle v_i, v_{i+1}\rangle \in E_G$. The path $v_1, v_2, \ldots, v_n$ of G **connects** the nodes v_1 and v_n and if for every pair of nodes $u, v \in V$ there exist a path connecting them, then G is **connected**. If H is a maximal connected subgraph of G, then H is a **component** of G.

Definition 3 Given a graph $G = (V, E)$, a **cycle** in G is a sequence of at least three nodes $v_1, v_2, \ldots, v_{n-1}, v_n$ where $v_1 = v_n$, $\langle v_{n-1}, v_1 \rangle \in E$ and $v_1, v_2, \ldots, v_{n-1}$ is a path of G. If there are no cycle in G, then G is **acyclic**.

Using the above concepts, it is possible to define trees and cacti, which are special classes of graphs for whom polynomial time algorithms will be given.

Definition 4 A graph $G = (V, E)$ is a **tree** iff G is connected and acyclic.

Definition 5 A graph $G = (V, E)$ is a **cactus** iff G is connected and any two cycles of G have at most one node in common.

Before formalizing the studied problems, the concepts of stars, constellations, covering and k-splits should be defined.

Definition 6 A graph $G = (V, E)$ is a **star** iff there exists a node $u \in V$ (**center**) such that the edges in E are defined between u and each one of the others nodes in V, i.e. $E = \{\langle u, v \rangle | \forall v \in V \setminus \{u\}\}$. In order to simplify the notation, every single edge $\langle u, v \rangle$ is considered a star, where the center can be any of the nodes u or v. If G is a star, then G is a k-star for any $k \geq |E|$.

Definition 7 A set of graphs is **edge-disjoint** if no two graphs in the set have common edges. A **constellation** is an edge-disjoint set of stars. Notice that the stars of a constellation may share nodes.

Definition 8 Given a graph $G = (V, E)$ and an edge e, the edge e is **covered** by G iff $e \in E$. A set of graphs covers an edge if at least one of the graphs covers the edge.

Definition 9 Given a set of stars $\mathcal{C}$ and an integer k, a set of k-stars C is a k-**split of** $\mathcal{C}$ iff every k-star of C is subgraph of at least one star in $\mathcal{C}$.

With the graphs notation clarified, the problems approached in this work are defined as follows.

Problem 1 *Minimum Constellation Cover (*CC*)*
Input: *A tuple $\langle G = (V, E), \mathcal{C} \rangle$, where:*

- *G is a graph;*
- *$\mathcal{C}$ is a set of stars of G.*

Output: *a minimum cardinality constellation $C \subseteq \mathcal{C} \cup E$ that covers all edges of E.*

Problem 2 *Minimum k-Split Constellation Cover (*K- SCC*)*
Input: *A tuple $\langle G = (V, E), \mathcal{C}, k \rangle$, where:*

- *G is a graph;*
- *$\mathcal{C}$ is a set of stars of G, that covers every edge of G;*
- *k is a positive integer.*

Output: *a minimum cardinality k-split constellation C of $\mathcal{C}$, that covers all edges of E.*

Next section proves strong NP-hard results and approximation limits for both problems.

3 NP-hardness and inapproximability bounds

In this section the hardness of CC and K- SCC are analyzed, proving that both problems belong to the NP-hard complexity class. Moreover, strong inapproximability results are presented for these problems. For CC, a reduction from the 3-Exact Cover (3- EC) is given, establishing that CC does not admit a polynomial time $|\mathcal{C}|^{\mathcal{O}(1)}$-approximation unless $P = NP$. For K- SCC, the reduction is from the Minimum Set Cover (MIN- SC) proving that, for any $\epsilon > 0$, K- SCC does not admit a polynomial time $((1-\epsilon)\ln|E|)$-approximation unless $P = NP$.

First, the result for CC is given and the definition of 3- EC is as follows.

Problem 3 3-*Exact Cover (*3- EC*)*
Input: *A tuple* $\langle X, \mathcal{T}\rangle$*, where:*

- X *is a set of elements;*
- $\mathcal{T}$ *is a set of triples of* X*, whose union is equal to* X*, i.e. each element of* $\mathcal{T}$ *is a subset of* X *with cardinality* 3 *and* $\bigcup_{t\in\mathcal{T}} t = X$.

Output: *a subset* T *of* $\mathcal{T}$ *satisfying that each element of* X *is contained in exactly one element of* T*, or inform that* T *does not exist.*

The following theorem not only shows that CC is strongly NP-hard, but also proves that it cannot be approximated with a ratio less than or equal to a polynomial function on the number of stars in the collection $\mathcal{C}$.

Theorem 1 CC *is NP-hard and, unless* $P = NP$*, for any instance* $I = \langle G, \mathcal{C}\rangle$*,* CC *cannot be* $|\mathcal{C}|^{\mathcal{O}(1)}$*-approximated in polynomial time, even if* G *is a star.*

Proof Let $I = \langle X, \mathcal{T}\rangle$ be an instance of 3- EC and $\ell = |\mathcal{T}|^{\alpha} \times |X|$, for some integer constant $\alpha \geq 1$. Then, the instance $I' = \langle G = (V, E), \mathcal{C}\rangle$ of CC is constructed as follows:

- The node v_o is defined in V.
- For each element $x \in X$, the nodes $x_1, \ldots, x_\ell$ are defined in V and the edges $\{\langle v_o, x_i\rangle\}_{i=1}^{\ell}$ are defined in E. The structure associated with x determines an ℓ-star denoted by **element-star of** x. Figure 1 illustrates this construction.

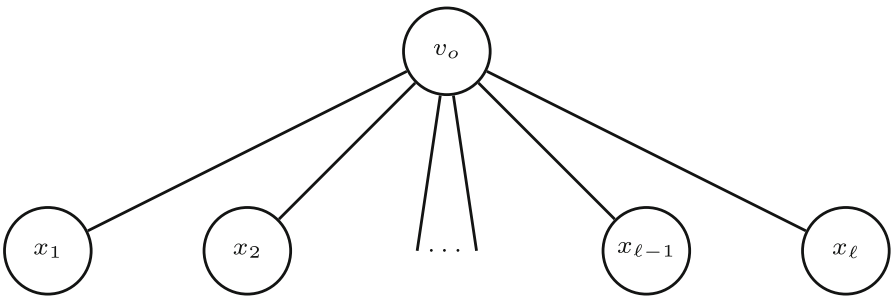

Fig. 1 Element-star of $x \in X$

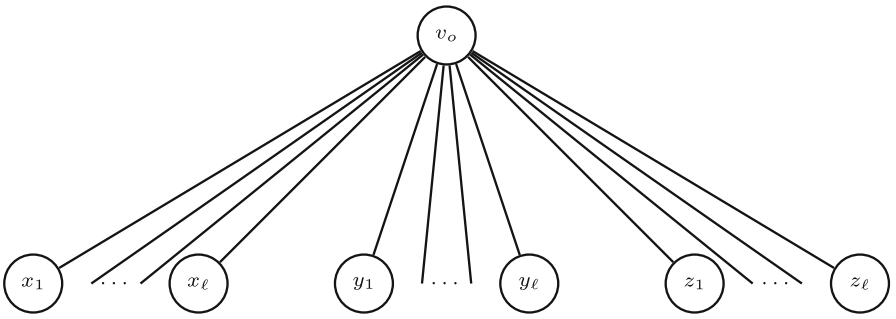

Fig. 2 Triple-star of $t = \langle x, y, z\rangle \in \mathcal{T}$

- For each triple $t = \langle x, y, z\rangle \in \mathcal{T}$, the $(3 \times \ell)$-star whose nodes are v_o, $\{x_i\}_{i=1}^{\ell}$, $\{y_i\}_{i=1}^{\ell}$ and $\{z_i\}_{i=1}^{\ell}$, and whose edges are $\{\langle v_o, x_i\rangle\}_{i=1}^{\ell}$, $\{\langle v_o, y_i\rangle\}_{i=1}^{\ell}$ and $\{\langle v_o, z_i\rangle\}_{i=1}^{\ell}$, is defined in $\mathcal{C}$. Such star is denoted by **triple-star of** t and Fig. 2 illustrates it.

Formally the elements of I' are defined as follows:

$$
\begin{aligned}
V &= \{v_o\} \cup \{x_i | \forall x \in X, 1 \leq i \leq \ell\}, \\
E &= \{\langle v_o, x_i\rangle | \forall x \in X, 1 \leq i \leq \ell\}, \\
\mathcal{C} &= \bigcup_{t \in \mathcal{T}} \{(\{v_o\} \cup \{x_i | \forall x \in t, 1 \leq i \leq \ell\}, \{\langle v_o, x_i\rangle | \forall x \in t, 1 \leq i \leq \ell\})\}.
\end{aligned}
$$

Trivially I' is a valid instance for CC and G is a star centered at v_o. Bellow, it is proved that there exists an exact cover for 3- EC with instance I iff there exists a solution whose cardinality is at most $\frac{|X|}{3}$ for CC with instance I'.

If there exists an exact cover $T \subseteq \mathcal{T}$ for 3- EC with instance I, then a solution C for CC with instance I' can be constructed by selecting for each triple $t \in T$ the triple-star of t, i.e. $S = \{\text{triple-star of } t | \forall t \in T\}$. Thus, each element of C is a triple-star of some triple in $\mathcal{T}$ and $C \subseteq \mathcal{C} \subseteq \mathcal{C} \cup E$. Since T is an exact cover, every element $x \in X$ is covered by exactly one triple $t \in T$, hence all edges of the element-star of x are covered by exactly one triple-star in C (the triple-star of t). Consequently, C is a constellation that covers all edges of G, implying that C is a feasible solution for CC with instance I'. Also, any triple of T covers three elements of X and each element of X is covered by exactly one triple in T, implying that $|T| = \frac{|X|}{3}$ and $|C| = |T| = \frac{|X|}{3}$.

For the other direction, suppose C is a feasible solution of CC with instance I' and $|S| \leq \frac{|X|}{3}$. Since $C \subseteq \mathcal{C} \cup E$, the elements of C can only be triple-stars or single edges. Thus, if for some element $x \in X$, the element-star of x is not subgraph of some triple-star in C, then each edge of the element-star of x is individually included in C and, since the element-star of x has ℓ edges, it follows:

$$
|C| \geq \ell = |\mathcal{T}|^{\alpha} \times |X| > \frac{|X|}{3}.
$$

Therefore, in order to guarantee that $|C| \leq \frac{|X|}{3}$, the element-star of each $x \in X$ must be subgraph of some triple-star in C. Furthermore, since C is a constellation, the elements of C do not share edges and each element-star of G must be subgraph of exactly one triple-star of C. Hence, the triples associated with the triple-stars of C are disjoint and cover all elements of X, so the set of such triples is an exact cover for 3- EC with instance I.

The constructed graph G has $1 + \ell \times |X|$ nodes and $\ell \times |X|$ edges, being $|\mathcal{T}|$ the number of stars in $\mathcal{C}$. Then, the size and the computational time required to construct I' from I are $\mathcal{O}(\ell \times |X| + |\mathcal{T}|) = \mathcal{O}\left(|\mathcal{T}|^{\alpha} \times |X|^2\right)$, which is polynomial on the size of I, and, since 3- EC is NP-complete (Garey and Johnson 1979), CC is NP-hard.

If CC admits a polynomial time $|\mathcal{C}|^{\theta}$-approximation algorithm, for some constant $\theta > 0$, then consider $\alpha = \lceil \theta \rceil$. Such an algorithm computes a solution C for CC with instance I' satisfying:

$$|C| \leq |\mathcal{C}|^{\theta} \times \left|C^*\right| = |\mathcal{T}|^{\theta} \times \left|C^*\right|,$$

where C^* is an optimal solution for CC with instance I'. If there exists an exact cover for 3- EC with instance I, then $|C^*| \leq \frac{|X|}{3}$, implying that:

$$|C| \leq |\mathcal{T}|^{\theta} \times \left|C^*\right| \leq |\mathcal{T}|^{\theta} \times \frac{|X|}{3}.$$

If an element-star for some $x \in X$ is not subgraph of a triple-star in C, then each edge of the element-star of x will be an element of S and it follows:

$$|C| \geq |\mathcal{T}|^{\alpha} \times |X| > |\mathcal{T}|^{\theta} \times \frac{|X|}{3}.$$

Thus, all element-stars are subgraphs of some triple-star in C. Therefore, the triples associated with triple-stars in C form an exact cover for 3- EC with instance I, and there exists a polynomial time algorithm for 3- EC. Moreover, unless $P = NP$, CC does not admit a polynomial time algorithm with approximation ratio less than or equal to $|\mathcal{C}|^{\theta}$, for any constant $\theta > 0$. Since the graph constructed in the reduction is a star, the result holds even for stars. □

The hardness and inapproximability results for K- SCC are obtained by a reduction from MIN- SC.

Problem 4 *Minimum Set Cover (*MIN- SC*)*
Input: *A tuple $\langle X, \mathcal{T} \rangle$, where:*

- *X is a set of elements;*
- *$\mathcal{S}$ is a set of subsets of X, whose union is equal to X, i.e. each element of $\mathcal{S}$ is a subset of X and $\bigcup_{s \in \mathcal{S}} s = X$.*

Output: *a minimum cardinality subset S of $\mathcal{S}$ satisfying that each element of X is contained in at least one element of S ($\bigcup_{s \in S} s = X$).*

Fig. 3 Node and element-edge of $x \in X$

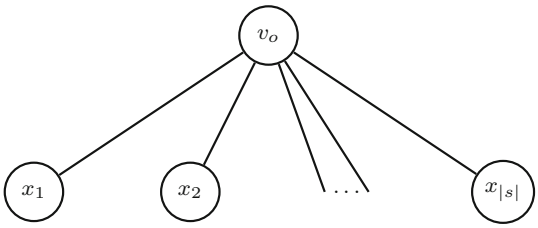

Fig. 4 Set-star of $s = \{x_1, x_2, \ldots, x_{|s|}\} \in \mathcal{S}$

The following proposition shows that any α-approximation algorithm for K- SCC will also α-approximate MIN- SC.

Proposition 1 *If, for some α, there exists a polynomial time α-approximation algorithm for* K- SC*C, then there exists a polynomial time α-approximation algorithm for* MIN- SC.

Proof Consider an instance $I = \langle X, \mathcal{S} \rangle$ of MIN- SC and construct an instance $I' = \langle G = (V, E), \mathcal{C}, k \rangle$ of K- SCC as follows:

- Define the node v_o in V.
- For each element $x \in X$, define the node x in V and the edge $\langle v_o, x \rangle$ in E. The edge $\langle v_o, x \rangle$ is denoted by **element-edge of** x. Figure 3 illustrates this construction.
- For each set $s = \{x_1, x_2, \ldots, x_{|s|}\} \in \mathcal{S}$, define in $\mathcal{C}$ the star whose nodes are v_o and $\{x_i\}_{i=1}^{|s|}$, and whose edges are $\{\langle v_o, x_i \rangle\}_{i=1}^{|s|}$. Such star is denoted by **set-star of** s. Figure 4 illustrates this construction.
- Finally, the integer k is set equal to $|X|$, so every set-star is a k-star.

The elements of I' can be formally defined as follows:

$$
\begin{aligned}
V &= \{v_o\} \cup \{x | \forall x \in X\}, \\
E &= \{\langle v_o, x \rangle | \forall x \in X\}, \\
\mathcal{C} &= \bigcup_{s \in \mathcal{S}} \{(\{v_o\} \cup \{x | \forall x \in s\}, \{\langle v_o, x \rangle | \forall x \in s\})\}, \\
k &= |X|.
\end{aligned}
$$

Since each element $x \in X$ is covered by at least one set $s \in \mathcal{S}$, the element-edge of x is covered by the set-star of s. Thus, every edge of G is covered by at least one

star of $\mathcal{C}$. Also, all set-stars are subgraphs of G, implying that I' is a valid instance for K- SCC.

Below, it is proved that there exists a feasible solution S for MIN- SC with instance I if there exists a feasible solution C with same cardinality as S, for K- SCC with instance I', i.e. $|S| = |C|$.

If $S \subseteq \mathcal{S}$ is a solution for MIN- SC with instance I, then a solution C for K- SCC with instance I' can be constructed by selecting, for each set $s \in S$, the set-star of s, i.e. $C = \{\text{set-star of } s | \forall s \in S\}$. After that, whenever two stars of C share an edge, the edge and the corresponding non-center node are removed from one of the stars, transforming C in a constellation. The edge deletion does not increase the size of C, so $|C| = |S|$. Also, the elements of C are subgraphs of set-stars associated with sets in $\mathcal{S}$, hence they are subgraphs of some stars in $\mathcal{C}$. Besides that, any star of G has at most $k = |X|$ edges, implying that C is a k-split constellation of $\mathcal{C}$. Since each $x \in X$ is covered by at least one $s \in S$, the element-edge of x is covered by a star in C and, consequently, C covers all edges in E. Therefore, C is a feasible solution of K- SCC with instance I' and $|C| = |S|$.

Furthermore, if S^* is an optimal solution of MIN- SC with instance I, then there exists a feasible solution C for K- SCC with instance I', such that $|S^*| = |C|$, implying that the optimal value of MIN- SC with instance I is greater than or equal to the optimal value of K- SCC with instance I'. Then, for any α, a feasible solution C that α-approximates an optimal solution C^* of K- SCC with instance I' satisfies:

$$|C| \leq \alpha \times \left|C^*\right| \leq \alpha \times \left|S^*\right|.$$

From any feasible solution C of K- SCC with instance I', a solution S for MIN- SC can be constructed by selecting, for each star c in C, the set of $\mathcal{S}$ associated with a set-star of $\mathcal{C}$ that contains the star c. The construction guarantees that $S \subseteq \mathcal{S}$ and, since C covers all edges of G, the union of the sets in S covers all elements of X, implying that S is feasible for MIN- SC with instance I. Moreover, since for each star of C at most one set was included in S, it follows that $|S| \leq |C|$ and:

$$|S| \leq |C| \leq \alpha \times \left|S^*\right|.$$

The instance I' has $1 + |X|$ nodes, $|X|$ edges and $|\mathcal{S}|$ stars in $\mathcal{C}$. Thus, the size of I' is $\mathcal{O}(|X| + |\mathcal{S}|)$, which is also the computational time required to construct I' from I. Also, the construction of S from C can be done in $\mathcal{O}(|X| \times |\mathcal{S}|)$ time. Hence, if there exists a polynomial time algorithm that α-approximates K- SCC, then there also exists a polynomial time algorithm that α-approximates MIN- SC. □

Since the graph on the above reduction is a star and, unless $P = NP$, MIN- SC does not admit a polynomial time $((1-\epsilon) \times \ln |X|)$-approximation for any $\epsilon > 0$ (Dinur and Steurer 2014), it follows.

Theorem 2 K- SCC *is NP-hard and, unless* $P = NP$*, for any instance* $I = \langle G = (V, E), \mathcal{C}, k\rangle$ *and* $\epsilon > 0$, K- SCC *cannot be* $((1-\epsilon) \times \ln |E|)$*-approximated in polynomial time, even if* G *is a star.*

Despite the strong inapproximability results for CC and K- SCC, the next section analyzes approximation ratios that can be obtained in polynomial time for both problems.

4 Approximation ratios

This section discuss on the approximability of CC and K- SCC, presenting the worst approximation ratios any feasible solution can have and, for K- SCC, different approximation algorithms are given.

The following proposition states an approximation ratio for the worst solution of CC. In this context, the worst solution is considered among the feasible constellations where every star covers at least one edge (if a feasible constellation contains an star without edges, then such star can be removed and the constellation remains feasible).

Proposition 2 *For any fixed integer $i \geq 1$, the worst feasible solution for* CC *with instance $I = \langle G, \mathcal{C} \rangle$ is a $\frac{\eta_i}{i}$-approximation of the optimal value, where η_i is the largest number of edges covered by any constellation $C \subseteq \mathcal{C}$ with at most i stars.*

Proof Consider an instance $I = \langle G, \mathcal{C} \rangle$, an optimal solution C^* of CC with instance I and a fixed integer $i \geq 1$. If η_i is the largest number of edges covered by any constellation $C \subseteq \mathcal{C}$ with at most i stars, then any subset with at most i stars of C^* covers at most η_i edges. Consequently, since C^* covers every edge of $|E|$, it follows:

$$\frac{|C^*|}{i} \geq \frac{|E|}{\eta_i} \quad \equiv \quad |E| \leq \frac{\eta_i}{i} \left|C^*\right|.$$

The domain of feasible solutions only contemplates constellations not containing 0-stars, so the worst feasible solution is the one where each star covers only one edge of E, being the cardinality of that solution $|E|$ which is a $\frac{\eta_i}{i}$-approximation of the optimal value. □

An immediate result of the above proposition is obtained when $i = 1$.

Remark 1 The worst feasible solution for CC with instance $I = \langle G, \mathcal{C} \rangle$ is a η-approximation of the optimal solution value, where η is the number of edges of the largest star in $\mathcal{C}$.

An analogous proof can be considered for K- SCC with the difference that in any optimal solution each star covers at most k edges, obtaining the following result.

Proposition 3 *The worst feasible solution for* K- SCC *with instance $I = \langle G, \mathcal{C}, k \rangle$ is a k-approximation of the optimal solution.*

The proposal of *"good"* approximation algorithms for CC may be very difficult given the very strong inapproximability result of Theorem 1. However, even when the inapproximability for K- SCC is also strong, the relation of this problem with MIN- SC allows to obtain some interesting approximation ratios.

4.1 K-SCC and MIN-SC

Besides the reduction from MIN- SC to K- SCC given by Proposition 1, a reduction from K- SCC to MIN- SC is also possible. The following proposition shows that any approximation algorithm for MIN- SC can be adapted to approximate K- SCC.

Proposition 4 *If, for some α and polynomial function p, there exists an α-approximation algorithm for* MIN- SC *with any instance $\langle X, \mathcal{S}\rangle$, whose time complexity is $\mathcal{O}(p(|X|, |\mathcal{S}|))$, then, for any instance $\langle G = (V, E), \mathcal{C}, k\rangle$ of* K- SCC, *there exists an $\frac{\alpha\times\eta}{k}$-approximation algorithm whose time complexity is $\mathcal{O}(p(|E|, |\mathcal{C}|) + |E| \times |\mathcal{C}|)$, being η the number of edges of the largest star in $\mathcal{C}$.*

Proof Consider the instance $I = \langle G = (V, E), \mathcal{C}, k\rangle$ of K- SCC and define an instance $I' = \langle X, \mathcal{S}\rangle$ of MIN- SC, where the elements of X are the edges of G and the subsets in $\mathcal{S}$ are the sets of edges associated with each star of $\mathcal{C}$, i.e. $X = E$ and $\mathcal{S} = \{E_c | \forall c \in \mathcal{C}\}$.

Let S be a feasible solution of MIN- SC with instance I' and, for each $s \in S$, include in a set C the star of $\mathcal{C}$ associated with s. After that, whenever an edge is covered by two stars of C, remove the edge and the corresponding non-center node from one of the stars. Then, C is a constellation whose stars are subgraphs of elements in $\mathcal{C}$ and, since S covers every element of X, it follows that C covers all edges of G. Any star $c \in C$ with more than k edges can be splitted into $\frac{|E_c|}{k}$ edge-disjoint k-stars (take the edges of c in some order and define one star with the first k edges, the second one with the next k edges and so on until the last one). Therefore, after splitting the stars, the constellation C is a k-split of $\mathcal{C}$, implying that C is a feasible solution for K- SCC with instance I.

Before splitting the stars in C, the cardinalities of C and S were the same and, after the splitting, each star in C was replaced by at most $\frac{\eta}{k}$ stars, being η the number of edges of the largest star in $\mathcal{C}$. Thus, $|C| \leq \frac{\eta}{k} \times |S|$.

If C^* is an optimal solution for K- SCC with instance I, then a feasible solution S' for MIN- SC with instance I' can be constructed by selecting, for each star $c \in C$, the set in $\mathcal{C}$ associated with a star in $\mathcal{C}$ that covers all edges of c. The construction of S' guarantees that $|C^*| \geq |S'|$. Hence, if S^* is an optimal solution of MIN- SC with instance I' and S is an α-approximation of S^*, for some α, then:

$$|C| \leq \frac{\eta}{k} \times |S| \leq \frac{\eta}{k} \times \alpha \times |S^*| \leq \frac{\eta}{k} \times \alpha \times |S'| \leq \frac{\eta}{k} \times \alpha \times |C^*|.$$

Consequently, C is an $\frac{\alpha\times\eta}{k}$-approximation for C^*. Since I' has an element in X for each edge in E and I' also has a set in $\mathbf{S}$ with size at most $|E|$ for each star in $\mathcal{C}$, it follows that the size of I' is $\mathcal{O}(|\mathcal{C}| \times |E|)$, which is also the computational time required to construct I' from I and to construct C from S. If, for some polynomial function p, there exist an $\mathcal{O}(p(|X|, |\mathcal{S}|))$ time α-approximation algorithm for MIN- SC, then there exists an $\mathcal{O}(p(|E|, |\mathcal{C}|) + |E| \times |\mathcal{C}|)$ time $\frac{\alpha\times\eta}{k}$-approximation algorithm for K- SCC, where η is the number of edges of the largest star in $\mathcal{C}$. □

In Williamson and Shmoys (2011), polynomial time algorithms were given to f-approximate MIN- SC and by applying the above proposition a polynomial time approximation algorithm is obtained for K- SCC.

Remark 2 For any instance $I = \langle G = (V, E), \mathcal{C}, k\rangle$ of K- SCC where η is the number of edges of the largest star in $\mathcal{C}$ and f is the maximum number of stars in $\mathcal{C}$ sharing a same edge, there exists a polynomial time algorithm with approximation ratio equals to $\frac{f\times\eta}{k}$.

Another approximation result for MIN- SC was given in Slavík (1996), where the proposed approximation ratio was $(\ln|X| - \ln\ln|X| + \Theta(1))$, allowing to obtain the following result for K- SCC.

Remark 3 For any instance $I = \langle G = (V, E), \mathcal{C}, k\rangle$ of K- SCC where η is the number of edges of the largest star in $\mathcal{C}$, there exists a polynomial time algorithm with approximation ratio equals to $\frac{(\ln|E| - \ln\ln|E| + \Theta(1))\times\eta}{k}$.

The approximations ratios given by the above remarks can be improved by proposing new approaches for K- SCC inspired in the results for MIN- SC of Slavík (1996), and Williamson and Shmoys (2011).

4.2 Integer linear program for K-SCC

An strategy to obtain an approximation ratio for K- SCC, that can be better than the one given by Remark 2 for some instances, is to propose an integer linear formulation for the problem, rounding an optimal solution of the linear relaxation. In that direction, consider the following integer linear program.

Model 1 *Formulation for* K- SCC *with instance* $I = \langle G = (V, E), \mathcal{C}, k\rangle$*:*

$$\min \sum_{s\in\mathcal{C}} x_s \tag{1}$$

$$s.t:$$

$$\sum_{s\in\{s'|s'\in\mathcal{C}, e\in E_{s'}\}} y_{se} = 1 \qquad \forall e \in E \tag{2}$$

$$\sum_{e\in E_s} y_{se} \leq k \times x_s \qquad \forall s \in \mathcal{C} \tag{3}$$

$$x_s \in \mathbb{Z},\ y_{se} \in \{0, 1\} \qquad \forall s \in \mathcal{C}, e \in E_s \tag{4}$$

The following lemma shows that, given an instance I, for any feasible solution C of K- SCC with instance I, there exists a feasible solution XY of Model 1 with instance I such that C and XY have the same value. This result implies that the optimal value of Model 1 with instance I is less than or equal to the optimal value of K- SCC with instance I.

Lemma 1 *For any instance* $I = \langle G = (V, E), \mathcal{C}, k\rangle$ *of* K- SCC *and each feasible solution* C *of* K- SCC *with instance* I*, there exists a feasible solution for Model 1 with instance* I *satisfying* $|C| = \sum_{s\in\mathcal{C}} x_s$.

Proof Let $I = \langle G = (V, E), \mathcal{C}, k\rangle$ be an instance of K- SCC. Given a feasible solution C for K- SCC with instance I, a solution for Model 1 can be constructed as follows:

1. For each star $s \in C$, select one star $s_{\mathcal{C}} \in \mathcal{C}$ such that s is subgraph of $s_{\mathcal{C}}$.
2. For each edge $e \in E$, select the star $s \in C$ that covers e, then set $y_{s_{\mathcal{C}}e} = 1$ and, for each $s' \in \mathcal{C} \setminus \{s_{\mathcal{C}}\}$ with $e \in E_{c'}$, set $y_{s'e} = 0$.
3. For each star $s \in \mathcal{C}$, set x_s equals to the number of stars in C for whom s was selected on first step, i.e. $x_s = \left|\left\{s' | s' \in C \text{ and } s'_{\mathcal{C}} = s\right\}\right|$.

Trivially, $y_{se} \in \{0, 1\}$ and $x_s \in \mathbb{Z}$ for each $s \in \mathcal{C}$ and $e \in E$, so the set of constraints 4 is satisfied. Also, for each edge $e \in E$, $y_{se} = 1$ only for one star $s \in \mathcal{C}$ that covers e, being zero for the rest of the stars in $\mathcal{C}$, hence the set of constraints 2 is also satisfied. Since each star in C covers at most k edges, it follows that for each $s \in \mathcal{C}$ the number of edges with $y_{se} = 1$ is at most k multiplied by the number of stars in C where s was selected on the first step, i.e. $|\{e | e \in E_s \text{ and } y_{se} = 1\}| \leq k \times \left|\left\{s' | s' \in C \text{ and } s'_{\mathcal{C}} = s\right\}\right|$, implying that:

$$\sum_{e \in E_s} y_{es} = |\{e | e \in E_s \text{ and } y_{se} = 1\}| \leq k \times \left|\left\{s' | s' \in C \text{ and } s'_{\mathcal{C}} = s\right\}\right| = k \times x_s.$$

Thus, the solution satisfies all constrains and is feasible for Model 1 with instance I. Furthermore, exactly one $s_{\mathcal{C}}$ was selected for each $s \in C$, so:

$$\sum_{s \in \mathcal{C}} x_s = \sum_{s \in \mathcal{C}} \left|\left\{s' | s' \in C \text{ and } s'_{\mathcal{C}} = s\right\}\right| = |C|.$$

□

The next lemma gives a rounding algorithm that receives a feasible solution of Model 1 with an instance I and constructs a feasible solution for K- SCC with instance I. Moreover, the lemma proves that both solutions have the same value.

Lemma 2 *Given an instance $I = \langle G = (V, E), \mathcal{C}, k\rangle$ and a feasible solution XY of Model 1 with instance I, if XY satisfies that $x_s = \left\lceil \frac{\sum_{e \in E_s} y_{se}}{k} \right\rceil$ for each $s \in \mathcal{C}$, then there exists a feasible solution C for K- SCC with instance I, that can be constructed from XY within $\mathcal{O}(|E| \times |\mathcal{C}|)$ time complexity and satisfies $|C| = \sum_{s \in \mathcal{C}} x_s$.*

Proof Let $XY = \left\langle \{x_s\}_{s \in \mathcal{C}}, \{y_{se}\}_{s \in \mathcal{C}, e \in E_s} \right\rangle$ be a feasible solution of Model 1 with instance $I = \langle G = (V, E), \mathcal{C}, k\rangle$. Also, suppose that, for each $s \in \mathcal{C}$, XY satisfies that $x_s = \left\lceil \frac{\sum_{e \in E_s} y_{se}}{k} \right\rceil$.

Consider the set of stars C where, for each star $s \in \mathcal{C}$, C contains the subgraph of s with all edges $e \in E_s$ satisfying $y_{se} = 1$ iff such subgraph is not a 0-star. That is,

$$C = \left\{s' | s' \subseteq s \in \mathcal{C}, |E_{s'}| > 0 \text{ and } \forall e \in E_s, e \in E_{s'} \text{ iff } y_{se} = 1\right\}.$$

Since XY satisfies constraints set 2, it follows that for each edge $e \in E$ there exists exactly one set $s \in \mathcal{C}$ for whom $y_{se} = 1$, being the subgraph of s the only star in C covering e. Thus, C is a constellation that covers every edge of E.

In order to guarantee C contains only k-stars, the constellation passes through a splitting/replacement process. For each star $s \in C$, if $|E_s| > k$, then replace s by the edge-disjoint stars in the sequence $\{s_i\}_{i=1}^{\left\lceil \frac{E_s}{k} \right\rceil}$, where the star s_i is the subgraph of s covering the edges from $k \times i$ to $k \times i + k - 1$ (considering some sorting for the edges of s). Therefore, the resulting C is a k-split constellation of $\mathcal{C}$ that covers all edges in E, implying that C is a feasible solution for K- SCC with instance I.

If $x_s = 0$ for some star $s \in \mathcal{C}$, then $y_{se} = 0$ for every edge $e \in E_s$, so only if $x_s > 0$ there will be subgraphs included in C for $s \in \mathcal{C}$. Moreover, if $s' \in C$ is the subgraph of s before the above splitting/replacement, then $|E_{s'}| = \sum_{e \in E_s} y_{se}$ and, after the splitting/replacement, the number of stars included for s in C is equal to $\left\lceil \frac{\sum_{e \in E} y_{se}}{k} \right\rceil = x_s$. Thus, $|C| = \sum_{s \in \mathcal{C}} x_s$ and the construction of C from XY can be done within $\mathcal{O}(|E| \times |\mathcal{C}|)$ time complexity by selecting for each star in $s \in \mathcal{C}$ the edges $e \in E_s$ for whom $y_{se} = 1$. □

Supported by the results of lemmata 1 and 2, the following theorem proves the existence of a polynomial time $(f \times k)$-approximation algorithm that improves the $\frac{f \times \eta}{k}$ ratio given by Remark 2 if $\eta > k^2$ (e.g., instances where k is a constant value and $\eta = \omega(1)$).

Theorem 3 *For any instance $I = \langle G = (V, E), \mathcal{C}, k \rangle$ of* K- SCC *there exists a polynomial time $(f \times k)$-approximation algorithm, where f is the maximum number of stars in $\mathcal{C}$ sharing a same edge.*

Proof Consider an instance $I = \langle G = (V, E), \mathcal{C}, k \rangle$ of K- SCC, an integer f equals to the maximum number of stars in $\mathcal{C}$ that share a same edge, and an optimal solution $XY^r = \left\langle \left\{x_s^r\right\}_{s \in \mathcal{C}}, \left\{y_{se}^r\right\}_{s \in \mathcal{C}, e \in E_s} \right\rangle$ of the linear relaxation of Model 1. Construct an integer solution $XY = \left\langle \{x_s\}_{s \in \mathcal{C}}, \{y_{se}\}_{s \in \mathcal{C}, e \in E_s} \right\rangle$ for Model 1 using the following rounding strategy:

- For each edge $e \in E$, select one star $s \in \mathcal{C}$ (with $e \in E_s$) for whom y_{se}^r is maximum. For that star set $y_{se} = 1$, setting $y_{s'e} = 0$ for the other stars $s' \in \mathcal{C}$ with $e \in E_{s'}$. So, the set of constraints 2 is satisfied.
- For each star $s \in \mathcal{C}$, set x_s equals to the minimum integer that satisfies constraints 3, i.e. $x_s = \left\lceil \frac{\sum_{e \in E_s} y_{se}}{k} \right\rceil$. Guaranteeing the set of constraints 3 is satisfied.

Since, $y_{se} \in \{0, 1\}$ and $x_s \in \mathbb{Z}$ for each $s \in \mathcal{C}$ and $e \in E$, the set of constraints 4 is satisfied and the rounded solution XY is feasible for Model 1. Furthermore, for each edge $e \in E$ the number of stars in $\mathcal{C}$ that cover e is at most f and since XY^r satisfies constraints 2, it follows:

$$f \times \max_{s \in \mathcal{C}, e \in E_s} \left\{y_{se}^r\right\} \geq \sum_{s \in \mathcal{C}, e \in E_s} y_{se}^r = 1.$$

For any $s \in \mathcal{C}$ and $e \in E_s$, if $y_{se} = 1$, then the rounding strategy guarantees that $y_{se}^r = \max_{s' \in \mathcal{C}, e \in E_{s'}} \left\{y_{s'e}^r\right\} \geq \frac{1}{f} = \frac{y_{se}}{f}$, implying that $y_{se} \leq f \times y_{se}^r$. Otherwise,

$y_{se} = 0 \leq y^r_{se} \leq f \times y^r_{se}$. Therefore, for each $s \in \mathcal{C}$:

$$\sum_{e \in E_s} y_{se} \leq \sum_{e \in E_s} f \times y^r_{se} \leq f \times k \times x^r_s.$$

The last inequality is given by constraints 3, which XY^r satisfies. Since $x_s = \left\lceil \frac{\sum_{e \in E_s} y_{se}}{k} \right\rceil$ for each $s \in \mathcal{C}$, trivially $x_s \leq \sum_{e \in E_s} y_{se}$ and:

$$\sum_{s \in \mathcal{C}} x_s \leq \sum_{s \in \mathcal{C}} \sum_{e \in E_s} y_{se} \leq (f \times k) \times \sum_{s \in \mathcal{C}} x^r_s.$$

The optimal value of the linear relaxation ($\sum_{s \in \mathcal{C}} x^r_s$) is less than or equal to the optimal solution value of Model 1 which, by Lemma 1 is less than or equal to the cardinality of an optimal solution of K- SCC. Hence, a constellation C obtained from XY by Lemma 2, is a $(f \times k)$-approximation for K- SCC with instance I. The existence of polynomial time algorithms for linear programs was proved in Khachiyan (1980) and the construction of XY from XY^r can be done in $\mathcal{O}(|E| \times |\mathcal{C}|)$, which is the same time required to construct C from XY in Lemma 2. Thus, there exists a polynomial time $(f \times k)$-approximation algorithm for K- SCC with instance I. □

4.3 Greedy algorithm for K-SCC

Another approach for approximating K- SCC may be to consider a greedy algorithm and to analyze its approximation ratio. Algorithm 1 is an example of this strategy.

```
   input  : I = ⟨G = (V, E), C, k⟩ instance of K- SCC
   output : C a k-split constellation of C covering all edges in E.
 1 begin
 2     C = ∅
 3     E_C = ∅
 4     while |E_C| < |E| do
 5         s ← star in C that covers most edges of E \ E_C
 6         s ← subgraph of s obtained by removing the edges in E_C
 7         if s has more than k edges then
 8             s ← subgraph of s with k edges
 9         end
10         C ← C ∪ {s}
11         E_C ← E_C ∪ E_s
12     end
13     return C
14 end
```

Algorithm 1: Greedy algorithm for K- SCC.

The next lemma shows that Algorithm 1 constructs in polynomial time a feasible solution for K- SCC over any valid instance.

Lemma 3 *For any instance $I = \langle G = (V, E), \mathcal{C}, k\rangle$ of* K- SCC, *Algorithm 1 computes a feasible solution within $\mathcal{O}\left(|E|^2 \times |\mathcal{C}|\right)$ time.*

Proof Consider an instance $I = \langle G = (V, E), \mathcal{C}, k\rangle$ of K- SCC as input of Algorithm 1.

If, at the beginning of an iteration of the loop between lines 4 and 12, the set C is a k-split constellation of $\mathcal{C}$ that covers the edges in $E_C \subset E$, then the star $s \in \mathcal{C}$ selected at line 5 covers at least one edge in $E \setminus E_C$ and after the update at line 6, the star s still covers such edge. Moreover, all edges covered by s after executing line 6 are not covered by any star of C and, if s covers more than k edges, then s is updated to a maximal k-star that is subgraph of its previous value (between lines 7 and 9). Thus, at line 9 the star s is a k-star that is subgraph of some star in $\mathcal{C}$, whose edges are not covered by any star of C, hence $C \cup \{s\}$ is a k-split constellation of $\mathcal{C}$ that covers the edges in $E_C \cup E_s$ and, since s covers at least one edge, $|E_C| < |E_C \cup E_s|$. Therefore, after executing lines 10 and 11, the new value of C is a k-split constellation of $\mathcal{C}$ that covers the edges of the new value of E_C and E_C contains at least one new edge.

The first value of C is the empty constellation (line 2) which covers the first value of $E_C \subset E$ (the empty set of edges, line 3), implying that at each iteration of the loop between lines 4 and 12, the set C is a k-split constellation of $\mathcal{C}$ that covers $E_C \subseteq E$. Since the loop stops when $E_C = E$, it follows that, at line 13, C is a feasible solution for K- SCC with instance I.

The computation of lines 2, 3 and 13 requires $\mathcal{O}(1)$ elementary operations. The number of iterations of the loop between lines 4 and 12 is $\mathcal{O}(|E|)$ because at least one new edge of E is included in E_C at each iteration. The computation of line 5 can be done in $\mathcal{O}(|E| \times |\mathcal{C}|)$ time, while the computation between lines 6 and 11 can be done in $\mathcal{O}(|E| + |\mathcal{C}|)$ time. In consequence, the overall time complexity of Algorithm 1 with instance I is $\mathcal{O}\left(|E|^2 \times |\mathcal{C}|\right)$. □

Inspired on the results of Slavík (1996), next lemma demonstrates that Algorithm 1 provides a solution whose approximation ratio is very close to the inapproximability limit given by Theorem 2.

Lemma 4 *For any instance $I = \langle G = (V, E), \mathcal{C}, k\rangle$ of* K- SCC, *the solution of Algorithm 1 is an $(\ln|E| - \ln\ln|E| + \Theta(1))$-approximation to the optimal value.*

Proof Consider an instance $I = \langle G = (V, E), \mathcal{C}, k\rangle$ of K- SCC, where C^* is an optimal solution and C is the solution given by Algorithm 1, being $\{s_i\}_{i=1}^{|C|}$ the sequence in which the stars were added to C by Algorithm 1.

Since C^* is a feasible solution of K- SCC with instance I, the edges in E can be covered by $|C^*|$ stars that are subgraphs of elements in $\mathcal{C}$, implying that at least one of those stars covers $\left\lceil \frac{|E|}{|C^*|} \right\rceil$ edges of E. The star $s_1 \in C$ is selected by Algorithm 1 in order to cover the greater quantity of edges, thus:

$$\left|E_{s_1}\right| \geq \left\lceil \frac{|E|}{|C^*|} \right\rceil = \left\lceil \frac{\sum_{j=1}^{|C|} \left|E_{s_j}\right|}{|C^*|} \right\rceil,$$

where $\sum_{j=1}^{|C|} \left|E_{s_j}\right| = |E|$ because the stars of C are edge-disjoint and cover all edges in E.

Analogously, at least one star of C^* covers $\left\lceil \frac{|E|-\left|E_{s_1}\right|}{|C^*|} \right\rceil$ edges of $E \setminus E_{s_1}$ and the star $s_2 \in C$ satisfies:

$$\left|E_{s_2}\right| \geq \left\lceil \frac{|E| - \left|E_{s_1}\right|}{|C^*|} \right\rceil = \left\lceil \frac{\sum_{j=2}^{|C|} \left|E_{s_j}\right|}{|C^*|} \right\rceil.$$

Generalizing the analysis for each $1 \leq i \leq |C|$, it follows:

$$\left|E_{s_i}\right| \geq \left\lceil \frac{|E| - \sum_{j=1}^{i-1} \left|E_{s_j}\right|}{|C^*|} \right\rceil = \left\lceil \frac{\sum_{j=i}^{|C|} \left|E_{s_j}\right|}{|C^*|} \right\rceil \quad \equiv \quad \left|E_{s_i}\right| \geq \left\lceil \frac{\sum_{j=i+1}^{|C|} \left|E_{s_j}\right|}{|C^*| - 1} \right\rceil$$

Any star $s \in C$ must cover at least one edge, otherwise s would not be included in C by Algorithm 1. Hence, $\left|E_{s_{|C|}}\right| \geq 1 = a_1$ and:

$$\left|E_{s_{|C|-1}}\right| \geq \left\lceil \frac{a_1}{|C^*| - 1} \right\rceil = a_2, \quad \ldots \quad \left|E_{s_1}\right| \geq \left\lceil \frac{\sum_{j=1}^{|C|-1} a_j}{|C^*| - 1} \right\rceil = a_{|C|}.$$

If the **greedy number** of Slavík (1996) ($\mathcal{N}\left(|C|, |C^*|\right) = \sum_{i=1}^{|C|} a_i$) is considered, then:

$$|E| = \sum_{i=1}^{|C|} \left|E_{s_i}\right| \geq \sum_{i=1}^{|C|} a_i = \mathcal{N}\left(|C|, \left|C^*\right|\right),$$

where, if $\mathcal{N}(l, m) \leq n$, then $\frac{l}{m} \leq (\ln n - \ln\ln n + \Theta(1))$ (Slavík 1996). Therefore:

$$|C| \leq (\ln |E| - \ln\ln |E| + \Theta(1)) \times \left|C^*\right|.$$

□

The next theorem is obtained by joining the results of lemmata 3 and 4, improving the approximation ratio given by Remark 3.

Theorem 4 *For any instance $I = \langle G = (V, E), \mathcal{C}, k\rangle$ of* K- SCC*, there exists an $\mathcal{O}\left(|E|^2 \times |\mathcal{C}|\right)$ time $(\ln |E| - \ln\ln |E| + \Theta(1))$-approximation algorithm.*

5 Polynomial time special cases

In addition to the complexity and approximability results, this work considers special classes of graphs for whom CC and K- SCC can be solved in polynomial time. This section presents a study over bounded degree recursively (ℓ, μ, L)-balanced graphs,

which is a class that includes paths, cycles, bounded degree trees and cacti, among other graph subclasses. The definition of this class is presented below.

Definition 10 Given an integer number $\ell \geq 1$ and a rational number $\mu > 1$, a graph $G = (V, E)$ with $|V| > \ell$ is (ℓ, μ)-**balanced** if there exists a set of nodes $B \subset V$ (named (ℓ, μ)-**balance set** of G), such that $|B| = \ell$ and each component of the induced subgraph of G over the nodes in $V \setminus B$ has at most $\frac{|V|}{\mu} - \ell$ nodes ($|V_H| \leq \frac{|V|}{\mu} - \ell$ for each component $H \subseteq G[V \setminus B]$). If $|V| \leq \ell$, then G is considered (ℓ, μ)-balanced.

Definition 11 Given two integers $L \geq \ell \geq 1$ and a rational $\mu > 1$, a graph $G = (V, E)$ is **recursively** (ℓ, μ, L)-**balanced** if $|V| \leq L$ or if G is (ℓ, μ)-balanced and for any (ℓ, μ)-balance set B of G, every component $H \subseteq G[V \setminus B]$ is a recursively (ℓ, μ, L)-balanced graph.

Next theorem shows that CC and K- SCC can be solved in polynomial time for recursively (ℓ, μ, L)-balanced graphs with bounded degree.

Theorem 5 *Given three fixed integers $\Delta \geq 0$ and $L \geq \ell \geq 1$ and a rational constant $\mu > 1$, for any instance $I = \langle G = (V, E), \mathcal{C} \rangle$ ($I = \langle G = (V, E), \mathcal{C}, k \rangle$) where G is recursively (ℓ, μ, L)-balanced and* $\max_{u \in V} \delta(u) = \Delta$, CC (K- SCC) *can be solved within the following time complexity:*

$$\begin{cases} \mathcal{O}\left(|V|^{\Delta \times \ell \times \log_\mu 2 + 1}\right), & \text{if } 2^\Delta > \mu \\ \mathcal{O}\left(|V|^{\Delta \times \ell \times \log_\mu 2 + 1} \times \log |V|\right), & \text{if } 2^\Delta = \mu \\ \mathcal{O}\left(|V|^{\ell+1}\right), & \text{if} 2^\Delta < \mu \end{cases}$$

Proof Let $I = \langle G = (V, E), \mathcal{C} \rangle$ ($I = \langle G = (V, E), \mathcal{C}, k \rangle$) be an instance of CC (K-SCC), where the maximum degree of the nodes in V is less than or equal to a constant Δ, being G recursively (ℓ, μ, L)-balanced, for some constant values $L \geq \ell \geq 1$ and $\mu > 1$.

If $\Delta < 2$, trivially there are no star with more than one edge and an optimal solution would be E. So, the following analysis is made for $\Delta \geq 2$.

Consider a (ℓ, μ)-balance set B of G and an optimal solution C^* for CC (K-SCC) with instance I. Denote by B^* the subset of C^* containing the stars centered at some node of B and, for each component $H = (V_H, E_H) \subseteq G[V \setminus B]$ denote by $H_{B^*} = \left(V_{H_{B^*}}, E_{H_{B^*}}\right)$ the subgraph of G obtained after including in H the edges of E that are not covered by B^* and have one node in B and the other in H, i.e.,

$$V_{H_{B^*}} = V_H \cup \left\{u | u \in B, v \in V_H, \langle u, v \rangle \in E, B^* \text{ does not cover } \langle u, v \rangle\right\},$$
$$E_{H_{B^*}} = E_H \cup \left\{\langle u, v \rangle | u \in B, v \in V_H, \langle u, v \rangle \in E, B^* \text{ does not cover } \langle u, v \rangle\right\}.$$

The above definition guarantees that, for each pair of different components $H_1, H_2 \subseteq G[V \setminus B]$, the graphs H_{1B^*} and H_{2B^*} are edge-disjoint and if they have some common nodes, then those nodes belong to B. Thus, for any component $H \subseteq G[V \setminus B]$, the stars of C^* covering the edges of H_{B^*} form an optimal solution of CC(K- SCC) over the instance that considers as input the graph H_{B^*} and the stars $\mathcal{C}$ that

are subgraphs of H_{B^*} (for K- SCCare also considered the subgraphs of the elements in $\mathcal{C}$ that are subgraphs of H_{B^*}).

Therefore, given a (ℓ, μ)-balance set B of G, an optimal solution for CC(K- SCC) with instance I can be computed as follows:

- Iterate over each constellation $B' \subseteq C$ (for K- SCCB' also considers subgraphs of elements in $\mathcal{C}$) whose stars are centered in B. Construct a solution $C_{B'}$ with the stars of B' (for K- SCCwith any minimum cardinality k-split of B') and, for each $H \subseteq G[V \setminus B]$, include in $C_{B'}$ an optimal solution of the subproblem considering the graph $H_{B'}$ and the respective stars of $\mathcal{C}$.
- Select a constellation $C_{B'}$ with minimum cardinality.

If two constellations B' and B'' of the above iteration satisfy that the edges covered by the stars of B' are the same edges covered by the stars of B'', then $H_{B'} = H_{B''}$ for each $H \subseteq G[V \setminus B]$. Hence, the computation of the subproblems optimal solutions need to be done only over constellations centered at B that cover different edges.

Algorithm 2 uses the above idea to exactly solve CC(K- SCC) when the graph is recursively (ℓ, μ, L)-balanced. Between lines 2 and 4 the algorithm starts checking if the instance is small enough to be exactly solved. If not, between lines 5 and 11 the algorithm searches for an (ℓ, μ)-balance set B by iterating over all possible subsets of size ℓ in G. If such set does not exists, then the graph is not recursively (ℓ, μ, L)-balanced and the algorithm halts with an error at Line 13, otherwise, between lines 15 and 32 the algorithm implements the idea previously described and at Line 33 an optimal solution is returned. □

Given a set of nodes S, the total number of constellations whose stars are centered at elements of S is bounded by the multiplication, over each $u \in S$, of the total number of edge-disjoint sets of stars centered at u. For each $0 \leq i \leq \delta(u)$, there are at most $\binom{\delta(u)}{i}$ stars with i edges centered at u, being $\sum_{i=0}^{\delta(u)} \binom{\delta(u)}{j} = 2^{\delta(u)}$ an upper bound for the number of stars centered at u. Thus, the number of sets with i stars centered at u is bounded by $\binom{2^{\delta(u)}}{i}$, and the total number of edge-disjoint sets of stars centered at u is bounded by $\sum_{i=0}^{\delta(u)} \binom{2^{\delta(u)}}{i} \leq 2^{\delta(u)^2} \leq 2^{\Delta^2}$. Furthermore, the number of constellations whose stars are centered at elements of S is less than or equal to $\prod_{u \in S} 2^{\Delta^2} \leq 2^{|S| \times \Delta^2}$.

From the above discussion, when Algorithm 2 executes Line 3, the computational time required is $\mathcal{O}\left(2^{L \times \Delta^2}\right)$ and since L and Δ are constant values the overall computation between lines 2 and 5 results $\mathcal{O}(1)$. There are $\binom{|V|}{\ell} \leq |V|^{\ell}$ possible sets of ℓ nodes and, by counting the nodes at each component in a *Deep First Search* (Cormen et al. 2009) one may check if a set is an (ℓ, μ)-balance of G. Hence, since $2 \times |E| = \sum_{u \in V} \delta(u) \leq \Delta \times |V|$ (*Handshaking Lemma*, Bollobás 1998), the computation between lines 6 and 11 can be done in $\mathcal{O}\left(|V|^{\ell} \times (|V| + |E|)\right) = \mathcal{O}\left(|V|^{\ell+1}\right)$ time.

input : $I = \langle G = (V, E), \mathcal{C} \rangle$ ($I = \langle G = (V, E), \mathcal{C}, k \rangle$) instance of CC (K- SCC) and constants $L \geq \ell \geq 1$ and $\mu > 1$
output : C^* optimal solution of CC (K- SCC) with instance I, if G is recursively (ℓ, μ)-balanced.

1 **begin**
2 **if** $|V| \leq L$ **then**
3 Enumerate all feasible constellations and return the best one
4 **end**
5 $B = \emptyset$
6 **for** *each set* S *of* ℓ *nodes of* V **do**
7 **if** S *is an* (ℓ, μ)*-balance of* G **then**
8 $B \leftarrow S$
9 `break`
10 **end**
11 **end**
12 **if** $B = \emptyset$ **then**
13 **ERROR** *"*G *is not recursively* (ℓ, μ, L)*-balanced"*
14 **end**
15 $C^* \leftarrow \emptyset$
16 $T \leftarrow$ empty dictionary
17 **for** *each constellation* B' *whose stars are centered at some node in* B **do**
18 $C' \leftarrow \emptyset$
19 **if** *the set* $E_{B'}$ *of the edges covered by* B' *is a* `key` *in* T **then**
20 $C' \leftarrow B' \cup T\left[E_{B'}\right]$
21 **else**
22 **for** *each component* $H \subseteq G[V \setminus B]$ **do**
23 $C^*_{H_{B'}} \leftarrow$ Algorithm 2 recursive call over the subproblem with $H_{B'}$
24 $C' \leftarrow C' \cup C^*_{H_{B'}}$
25 **end**
26 $T\left[E_{B'}\right] \leftarrow C'$
27 $C' \leftarrow C' \cup B'$
28 **end**
29 **if** $\left|C'\right| < \left|C^*\right|$ *or* $C^* = \emptyset$ **then**
30 $C^* \leftarrow C'$
31 **end**
32 **end**
33 **return** C^*
34 **end**

Algorithm 2: Exact algorithm for CC (K- SCC) when the graph is recursively (ℓ, μ, L)-balanced.

The computational time required between lines 12 and 16 is $\mathcal{O}(1)$ and the loop from line 17 – 32 is executed at most $\mathcal{O}\left(2^{|B| \times \Delta^2}\right) = \mathcal{O}\left(2^{\ell \times \Delta^2}\right) = \mathcal{O}(1)$ times, which corresponds to the number of constellations with stars centered at nodes of B. At each iteration of the loop, the number of edges in $E_{B'}$ is limited by $\Delta \times \ell$, so almost all instructions of the loop can be executed in $\mathcal{O}(1)$ time, with exception of the recursive calls between lines 22 and 25. The recursive calls are executed when a constellation B' covers edges not covered by a previous constellation and, since any constellation whose stars are centered at nodes of B may cover at most $\Delta \times \ell$ edges, it follows that the total number of different sets of edges is limited by $2^{\Delta \times \ell}$, which is an upper bound for the maximum number of times Algorithm 2 executes the instructions

from line 22 – 25. Moreover, the following recursive function is an upper bound for the overall computational time required by Algorithm 2:

$$\mathcal{F}(|V|) = 2^{\Delta\times\ell} \times \sum_{H\subseteq G[V\setminus B]} \mathcal{F}(|H_{B'}|) + \Theta\left(|V|^{\ell+1}\right).$$

Each component H at line 22 has at most $\frac{|V|}{\mu} - \ell$ nodes, implying that $H_{B'}$ has at most $\frac{|V|}{\mu}$ nodes. Also, $\mathcal{F}(a) + \mathcal{F}(b) \leq \mathcal{F}(a+b)$ for any two values $a, b \geq 1$, allowing to consider the union of components whose nodes sum is less than or equal to $\frac{|V|}{\mu}$, hence:

$$\mathcal{F}(|V|) \leq 2^{\Delta\times\ell} \times \mu \times \mathcal{F}\left(\frac{|V|}{\mu}\right) + \Theta\left(|V|^{\ell+1}\right).$$

The proof is completed by using the *Master Theorem* (Cormen et al. 2009) on $\mathcal{F}$:

$$\mathcal{F}(|V|) = \begin{cases} \mathcal{O}\left(|V|^{\Delta\times\ell\times\log_\mu 2+1}\right), & \text{if } 2^\Delta > \mu \\ \mathcal{O}\left(|V|^{\Delta\times\ell\times\log_\mu 2+1} \times \log|V|\right), & \text{if } 2^\Delta = \mu \\ \mathcal{O}\left(|V|^{\ell+1}\right), & \text{if } 2^\Delta < \mu \end{cases}$$

□

The above theorem can be applied to different special classes of graph such as trees. Every tree $G = (V, E)$ has at least one **centroid** (Jordan 1869), which is a node $u \in V$, satisfying that each component of $G[V \setminus \{u\}]$ has at most $\frac{|V|}{2}$ nodes. Hence, if $|V| \geq 6$, then each component of $G[V \setminus \{u\}]$ has at most $\frac{2\times|V|}{3} - 1$ nodes. Since each connected subgraph of a tree is also a tree, it follows that any tree with is recursively $\left(1, \frac{3}{2}, 6\right)$-balanced.

Corollary 1 *For any instance $I = \langle G, \mathcal{C}\rangle$ ($I = \langle G, \mathcal{C}, k\rangle$) where G is a tree with maximum degree equals to a constant Δ,* CC *(*K- SCC*) can be solved within $\mathcal{O}\left(|V|^{1.71\times\Delta+1}\right)$ time complexity.*

An extension of the concept of centroid can be used to obtain an analogous result for cacti, allowing to obtain the following corollary.

Corollary 2 *For any instance $I = \langle G, \mathcal{C}\rangle$ ($I = \langle G, \mathcal{C}, k\rangle$) where G is a cactus with maximum degree equals to a constant Δ,* CC *(*K- SCC*) can be solved within $\mathcal{O}\left(|V|^{3.42\times\Delta+1}\right)$ time complexity.*

Proof Consider a cactus $G = (V, E)$, a *spanning tree* T of G (Bollobás 1998) and a centroid u of T. If each component of $G[V \setminus \{u\}]$ has at most $\frac{|V|}{2}$ nodes, then, for any node $v \in V$ with $v \neq u$, each component of $G[V \setminus \{u, v\}]$ also has at most $\frac{|V|}{2}$ nodes. Otherwise, there exist exactly one component $H \subseteq G[V \setminus \{u\}]$ with more than $\frac{|V|}{2}$ nodes, and H contains at least two components $H_1, H_2 \subset T\,|V \setminus \{u\}|$. Moreover, the nodes of H are exactly the nodes of the components $H_1, H_2 \subset T[V \setminus \{u\}]$, otherwise,

if H contains a node $v \notin V_{H_1} \cup V_{H_2}$, then there exist two cycles in G with two common nodes u and v, and G would not be a cactus. Furthermore, since there exist already a path through u connecting H_1 to H_2, it follows that there exist only one edge $\langle v_1, v_2 \rangle$ with $v_1 \in H_1$ and $v_2 \in H_2$. Hence, $H[V \setminus \{v_1\}]$ has two components, each one with at most $\frac{|V|}{2}$ nodes.

Therefore, $G = (V, E)$ has two nodes $u, v \in V$ for whom each component of $G[V \setminus \{u, v\}]$ has at most $\frac{|V|}{2}$ nodes. If $|V| \geq 12$, then each component of $G[V \setminus \{u, v\}]$ has at most $\frac{2\times|V|}{3} - 2$ nodes, implying that every cactus with at least twelve nodes is $\left(2, \frac{3}{2}\right)$-balanced and, because every connected subgraph of a cactus is also a cactus, it follows that all cacti are recursively $\left(2, \frac{3}{2}, 12\right)$-balanced. Thus, by using Theorem 5 the proof is completed. □

6 Final comments

This paper introduced CC and K- SCC, which are two exact graph covering problems that belong to the complexity class NP-hard and whose solutions may help in network design and network monitoring applications. Strong inapproximability results were proved for both problems and approximation ratios were given for the worst feasible solutions. Besides, for K- SCC, approximation algorithms were presented achieving an approximation ratio very close to the inapproximability bound proved for the problem. Additionally, a polynomial time algorithm was discussed for a special class of bounded degree graphs that includes common network layouts as trees and cacti. Many questions remain open regarding CC, K- SCC and related problems. Does CC admits a polynomial time approximation algorithm whose ratio is a logarithmic function on the number of nodes? Is it possible to obtain tighter approximation ratios for K- SCC in polynomial time? Are there more special cases that can be solved efficiently? What is the problems complexity if instead of constellations there are considered other collections of graphs? These and other questions will be approached in future works.

Funding Not applicable.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

Bollobás B (1998) Modern Graph Theory, 1st edn. Springer-Verlag, New York

Cormen TH, Leiserson CE, Rivest RL, Stein C (2009) Introduction to algorithms, 3rd edn. The MIT Press, Cambridge

Dinneen MJ, Hua R (2017) Formulating graph covering problems for adiabatic quantum computers. In: Proceedings of the Australasian Computer Science Week Multiconference, ACSW '17. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3014812.3014830

Dinur I, Steurer D (2014) Analytical approach to parallel repetition. In: Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing, STOC '14, p. 624–633. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/2591796.2591884

Fernandes CG, Lee O, Wakabayashi Y (2009) Minimum cycle cover and chinese postman problems on mixed graphs with bounded tree-width. Discret Appl Math 157(2):272–279. https://doi.org/10.1016/j.dam.2007.10.032

Fukunaga T (2016) Covering problems in edge- and node-weighted graphs. Discret Optim 20:40–61. https://doi.org/10.1016/j.disopt.2016.03.001

Garey MR, Johnson DS (1979) Computers and Intractability: a guide to the theory of NP-Completeness. W. H. Freeman & Co., USA

Hung RW, Chang MS (2007) Finding a minimum path cover of a distance-hereditary graph in polynomial time. Discret Appl Math 155(17):2242–2256. https://doi.org/10.1016/j.dam.2007.06.001

Jordan C (1869) Sur les assemblages de lignes. Journal für die reine und angewandte Mathematik 1869(70):185–190 https://doi.org/10.1515/crll.1869.70.185. https://www.degruyter.com/view/journals/crll/1869/70/article-p185.xml

Khachiyan L (1980) Polynomial algorithms in linear programming. USSR Computational Mathematics and Mathematical Physics 20(1):53–72 https://doi.org/10.1016/0041-5553(80)90061-0. http://www.sciencedirect.com/science/article/pii/0041555380900610

Marques JA, Luizelli MC, da Costa Tavares, Filho RI, Gaspary LP (2019) An optimization-based approach for efficient network monitoring using in-band network telemetry. J Internet Serv Appl 10(1):12. https://doi.org/10.1186/s13174-019-0112-0

Álvarez Miranda E, Sinnl M (2020) A branch-and-cut algorithm for the maximum covering cycle problem. Annal Op Res 284(2):487–499. https://doi.org/10.1007/s10479-018-2856-5

Pilipczuk M, van Leeuwen EJ, Wiese A (2020) Quasi-polynomial time approximation schemes for packing and covering problems in planar graphs. Algorithmica 82(6):1703–1739. https://doi.org/10.1007/s00453-019-00670-w

Rizzi R, Tomescu AI, Mäkinen V (2014) On the complexity of minimum path cover with subpath constraints for multi-assembly. BMC Bioinform 15(9):S5. https://doi.org/10.1186/1471-2105-15-S9-S5

Slavík P (1996) A tight analysis of the greedy algorithm for set cover. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, STOC '96, p. 435–441. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/237814.237991

Wegner AE (2014) Subgraph covers: an information-theoretic approach to motif analysis in networks. Phys Rev X 4:041026. https://doi.org/10.1103/PhysRevX.4.041026

Williamson DP, Shmoys DB (2011) The design of approximation algorithms, 1st edn. Cambridge University Press, USA