Theoretical Computer Science 942 (2023) 21–32

Contents lists available at ScienceDirect

Theoretical Computer Science

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

Complexity and approximability of Minimum Path-Collection Exact Covers ☆

Santiago Valdés Ravelo [a,*], Cristina G. Fernandes [b]

[a] *Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, Brazil*
[b] *Department of Computer Science, University of São Paulo, Brazil*

ARTICLE INFO

Article history:
Received 24 March 2022
Received in revised form 17 November 2022
Accepted 20 November 2022
Available online 23 November 2022
Communicated by R. Klasing

Keywords:
Exact edge cover
Computational complexity
Approximability
Polynomial-time algorithm
Combinatorial optimization

ABSTRACT

This work considers the Minimum Path-Collection Exact Cover (PCEC) and the Minimum k-Path Splitting Exact Cover (κ-PSEC). Both problems receive a digraph G and a set $\mathcal{P}$ of paths in G, and their objective is to find a minimum cardinality set $\mathcal{S}$ of paths, whose elements are arc-disjoint and cover all arcs of G. Despite the similarities, the solutions for each problem must satisfy different properties. For κ-PSEC, the set $\mathcal{S}$ must be obtained by splitting the paths in $\mathcal{P}$ in order to guarantee that no element of $\mathcal{S}$ has length greater than a given integer k. For PCEC, the paths in $\mathcal{P}$ cannot be split, and the elements of $\mathcal{S}$ are single arcs of G or complete paths of $\mathcal{P}$.

PCEC and κ-PSEC have practical applications in network design and network monitoring systems, being also natural versions of graph covering problems. However, there are no theoretical studies on their complexity. This work not only presents NP-hardness results for the problems, but also proves that, unless P = NP, PCEC cannot be $|\mathcal{P}|^{O(1)}$-approximated in polynomial-time. Moreover, polynomial-time algorithms are presented for paths, cycles, and trees, and polynomial-time approximation algorithms are proposed for special cases of κ-PSEC.

1. Introduction

Path covers have been considered in the literature to refer to a set of paths in a graph such that every vertex belongs to at least one of them [1]. The goal is to find minimum, or small, path covers. In this paper, we deal with path covers of a different nature that have applications, for instance, in presenting workflow graphs into tabular formats [2] and in in-band network telemetry [3–5]. First, the goal is to cover the edges and not the graph's vertices, and we are interested in exact covers, that is, we want to cover each edge exactly once. Second, the set of paths that can be used in the cover is given in advance as part of the problem instance. If we did not have the second restriction, the problem would be solvable through flow algorithms [6]. Next, we describe one of the applications in more detail to show where the given paths might come from.

The quality of distributed and cloud-driven networks is highly dependent on reliable monitoring strategies to diagnose system anomalies. In-band network telemetry emerged as an alternative to gain granular visibility of the network in real-time [3–5]. This mechanism consists of embedding information of the links connecting the devices to packets traversing

☆ This article belongs to Section A: Algorithms, automata, complexity and games, Edited by Paul Spirakis.

* Corresponding author.
E-mail addresses: santiago.ravelo@inf.ufrgs.br (S. Valdés Ravelo), cris@ime.usp.br (C.G. Fernandes).

https://doi.org/10.1016/j.tcs.2022.11.022

them. To prevent telemetry overhead, the collected data of each packet must be dispatched before exceeding the transmission bandwidth of the network, and such dispatches should be as few as possible to avoid network degradation. So the goal is to determine in which subpaths of each packet's flow the information will be collected and sent [4].

The above problem can be formalized as the Minimum k-Path-Splitting Exact Cover (κ-PSEC). The input of this problem is a directed graph G (the network), a set $\mathcal{P}$ of paths of G (all packets' flows), and an integer k (bandwidth of the network). The objective is to cover each arc of G by exactly one element of a minimum cardinality set $\mathcal{S}$, consisting of paths with length at most k that are subpaths of paths in $\mathcal{P}$. The Minimum Path-Collection Exact Cover (PCEC) is a generalization of κ-PSEC for which the elements of $\mathcal{S}$ must be complete paths in $\mathcal{P}$ (instead of subpaths with length at most k) or arcs of G.

Solutions that do not cover the same set of arcs are usually not comparable, so the problems address instances for which every arc is covered by some given path and require that all arcs are covered in a solution. To achieve this, for PCEC, the given paths are complemented by arcs of the graph. This also makes the problems amenable to the design of approximation algorithms, as all such instances are feasible.

PCEC and κ-PSEC are natural variants of the widely studied family of path covering/partitioning problems [1]. However, only one previous study contemplates problems on undirected graphs receiving a collection of subgraphs to cover the edges [7]. The findings of such a study include NP-hardness and strong inapproximability results, approximation algorithms, and a polynomial-time algorithm for variants aiming to cover the edges by stars instead of paths. Hence, to the best of our knowledge, there are no complexity or approximability results for PCEC or κ-PSEC.

Motivated by the lack of theoretical results for PCEC and κ-PSEC, and also by their practical applicability, this work approaches these problems. We consider both directed graphs (digraphs, for short) and (undirected) graphs. The main contributions of this paper are listed below:

- NP-hardness proof for κ-PSEC and a strong inapproximability result for PCEC, even when only acyclic digraphs with bounded degree are considered (Section 2).
- Polynomial-time algorithms for PCEC and κ-PSEC for polytrees or polytrees plus one arc (Section 3).
- Polynomial-time approximation algorithms for κ-PSEC on special classes of digraphs:
  - $\frac{3}{2}$-approximation for digraphs with maximum degree 3 (Subsection 4.1).
  - $\frac{k+1}{2}$-approximation on general digraphs and, for every $\Delta \geq 1$, $\frac{\Delta+1}{2}$-approximation on digraphs where each node has odd degree bounded by Δ (Subsection 4.2).

1.1. Notation and formalization of the problems

As most of our results will be stated and proved for digraphs, most of the time, we will adopt the terms nodes and arcs respectively to refer also to the vertices and edges of a graph. For a (directed) graph G, we will use V_G to denote the node set of G, and A_G to denote the arc set of G. The **degree** of a node v in a directed graph is the number of incoming and outgoing arcs incident to v, and in an undirected graph is the number of edges incident to v. The maximum degree of a (directed) graph G, denoted by $\Delta(G)$, is the maximum degree among all nodes of G.

Given two (directed) graphs G and H, we say H is a **subgraph of** G if the set of nodes of H is a subset of the nodes of G and the set of arcs of H is a subset of the arcs of G. A subgraph H of G **covers** an arc $a \in A_G$ if $a \in A_H$. In such case, it is also said that a **is covered by** H. A set of subgraphs of G covers an arc of G if the arc is covered by at least one of the subgraphs in the set. If all subgraphs in the set are paths, we call the set a path-collection. Next, we define the two problems we will address in this paper in terms of digraphs.

Problem 1.1. Minimum Path-Collection Exact Cover (PCEC)
Input: a tuple $\langle G, \mathcal{P} \rangle$, where G is a digraph and $\mathcal{P}$ is a set of paths of G.
Output: a minimum cardinality set $\mathcal{S} \subseteq \mathcal{P} \cup A_G$ such that each arc of A_G is covered by exactly one element of $\mathcal{S}$.

The second problem is a particular case of the first one, represented in a compact way.

Problem 1.2. Minimum k-Path-Splitting Exact Cover (κ-PSEC)
Input: a tuple $\langle G, \mathcal{P}, k \rangle$, where G is a digraph, $\mathcal{P}$ is a set of paths of G that covers every arc of G, and k is a positive integer.
Output: a minimum cardinality set $\mathcal{S}$ of paths of G such that each arc of G is covered by exactly one path of $\mathcal{S}$, each path of $\mathcal{S}$ is a subpath of at least one path of $\mathcal{P}$ and each path of $\mathcal{S}$ has length at most k.

These problems can be defined also for undirected graphs, and most of our results hold for these variants. When referring to these, we add the adjective *undirected* to the acronym of the problems.

2. NP-hardness and inapproximability

This section shows polynomial reductions from a restricted version of the well-known Exact 3-Cover to PCEC and κ-PSEC, proving that these problems are NP-hard. These reductions also imply inapproximability results for the problems. The restricted version of the Exact 3-Cover used was shown by Gonzalez [8] to be NP-complete and is defined as follows.

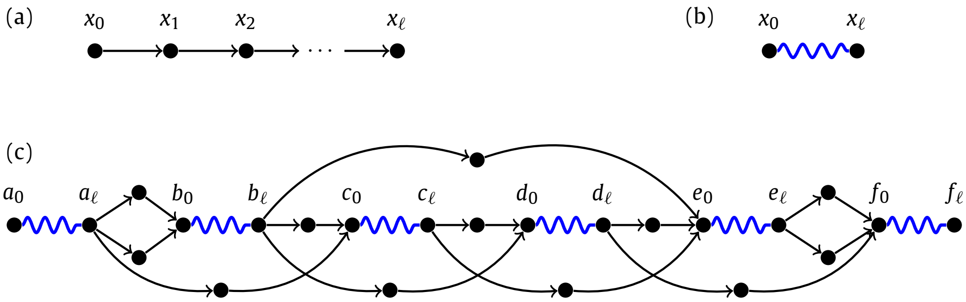

Fig. 1. (a) The element-path for an $x \in X$. (b) Representation of an element-path. (c) The digraph G for the instance $\langle X, \mathcal{T}\rangle$ of RX3C with $X = \{a, b, c, d, e, f\}$ and $\mathcal{T} = \{abc,abd,ace,bef,cdf,def\}$.

Problem 2.1. Restricted Exact 3-Cover (RX3C)
Input: a tuple $\langle X, \mathcal{T}\rangle$, where X is a set of elements and $\mathcal{T}$ is a set of triples of elements of X such that every element in X is in exactly three triples in $\mathcal{T}$.
Output: a subset T of $\mathcal{T}$ such that each element of X is contained in exactly one triple in T, or the information that such T does not exist.

The NP-hardness for PCEC comes from the following theorem, which establishes a strong inapproximability result that implies that PCEC does not belong to the APX class of problems unless P = NP, that is, there is no polynomial-time constant approximation for PCEC unless P = NP. The result holds for the particular class of acyclic digraphs with maximum degree 4. Recall that a (directed) graph G is **acyclic** if there is no (directed) cycle in G.

Theorem 2.1. PCEC *is NP-hard and, denoting by* $I = \langle G, \mathcal{P}\rangle$ *an arbitrary instance, there is no polynomial-time* $|\mathcal{P}|^{O(1)}$*-approximation for* PCEC *unless P = NP, even when restricted to acyclic digraphs with maximum degree 4.*

Proof. Let $I = \langle X, \mathcal{T}\rangle$ be an instance of RX3C. Note that $|\mathcal{T}| = |X|$ as each element in X is in exactly three triples in $\mathcal{T}$. Without loss of generality, assume that the elements of X are ordered and, for each triple $t \in \mathcal{T}$, the three elements maintain the order they have in X. Let $\ell = 3 \cdot |X|^c$ for some integer constant $c \geq 1$. An instance $I' = \langle G = \langle V, A\rangle, \mathcal{P}\rangle$ of PCEC is constructed as follows.

The set V consists of $\ell + 1$ nodes $x_0, x_1, \ldots, x_\ell$ for each element $x \in X$, and two nodes t_{xy} and t_{yz} for each triple $\langle x, y, z\rangle \in \mathcal{T}$. There is an arc in A from x_{i-1} to x_i for $i = 1, \ldots, \ell$ for each element $x \in X$ and, for each triple $\langle x, y, z\rangle \in \mathcal{T}$, there are four arcs in A: from x_ℓ to t_{xy}, from t_{xy} to y_0, from y_ℓ to t_{yz}, and from t_{yz} to z_0. The arcs in A for each element $x \in X$ form a path from x_0 to x_ℓ, referred to as an **element-path**. Fig. 1 illustrates an instance of RX3C and the corresponding digraph $G = \langle V, A\rangle$.

Because the elements of X are ordered and the three elements in every triple in $\mathcal{T}$ maintain the order they have in X, the constructed digraph $G = \langle V, A\rangle$ of I' is acyclic. Moreover, because each element is in exactly three triples, the number of incoming arcs from any node is at most three, and the number of outgoing arcs from any node is also at most three. Furthermore, the maximum degree of G is at most 4.

Now we proceed to define $\mathcal{P}$. For each triple $t = \langle x, y, z\rangle \in \mathcal{T}$, the concatenation of the element-paths for x, y, and z going through the connecting nodes, that is, the path $x_0x_1 \ldots x_\ell t_{xy} y_0 y_1 \ldots y_\ell t_{yz} z_0 z_1 \ldots z_\ell$, is called a **triple-path**. The set $\mathcal{P}$ consists of all triple-paths, therefore $|\mathcal{P}| = |\mathcal{T}| = |X|$.

Trivially, I' is a valid instance for PCEC. Below, it is proved that there exists an exact cover for instance I of RX3C if and only if there exists a solution for instance I' of PCEC whose cardinality is at most $3 \cdot |X|$.

If there exists an exact cover $T \subseteq \mathcal{T}$ for instance I of RX3C, then a solution $\mathcal{S}$ for instance I' of PCEC can be constructed as follows:

Step 1: For each triple $t \in T$, the triple-path of t is added to $\mathcal{S}$.
Step 2: For each triple $t = \langle x, y, z\rangle \in \mathcal{T} \setminus T$, the arcs $\langle x_\ell, t_{xy}\rangle$, $\langle t_{xy}, y_0\rangle$, $\langle y_\ell, t_{yz}\rangle$, and $\langle t_{yz}, z_0\rangle$ are added to $\mathcal{S}$.

Notice that $\mathcal{S}$ only contains arcs of A and triple-paths of $\mathcal{P}$, therefore $\mathcal{S} \subseteq \mathcal{P} \cup A$. Since T is an exact cover, each element $x \in X$ is covered by exactly one triple $t \in T$, and every arc of the element-path associated with x is covered by the triple-path of t (added to $\mathcal{S}$ in Step 1). Also, since the triples in T are disjoint, those triple-paths added to $\mathcal{S}$ do not share arcs. Moreover, every arc in A not covered by the triple-paths in $\mathcal{S}$ at the end of Step 1 is added to $\mathcal{S}$ in Step 2. Thus, each arc of A is covered by exactly one element of $\mathcal{S}$, implying that $\mathcal{S}$ is a feasible solution for instance I' of PCEC. Besides, $\mathcal{S}$ satisfies

$$|\mathcal{S}| = |T| + 4 \cdot |\mathcal{T} \setminus T| = 4 \cdot |\mathcal{T}| - 3 \cdot |T| = 4 \cdot |X| - |X| = 3 \cdot |X|,$$

because $3 \cdot |T| = |X|$ as T is an exact cover, and $|\mathcal{T}| = |X|$ for RX3C.

For the other direction, suppose $\mathcal{S}$ is a feasible solution for instance I' of PCEC and $|\mathcal{S}| \leq 3 \cdot |X|$. Since $\mathcal{S}$ is a feasible solution, $\mathcal{S} \subseteq \mathcal{P} \cup A$, meaning that $\mathcal{S}$ may only contain triple-paths and single arcs. Therefore, if, for some element $x \in X$, the element-path associated with x is not a subpath of some triple-path in $\mathcal{S}$, then all ℓ arcs of such an element-path would be individually in $\mathcal{S}$ and $|\mathcal{S}| > \ell = 3 \cdot |X|^c \geq 3 \cdot |X|$ because $c \geq 1$, a contradiction. Consequently, for every element $x \in X$, the element-path associated with x is a subpath of some triple-path of $\mathcal{S}$. Also, each arc of G is covered by exactly one element of $\mathcal{S}$, implying that each element-path is covered by exactly one triple-path of $\mathcal{S}$. Thus, the triples associated with the triple-paths of $\mathcal{S}$ are disjoint and cover all elements of X. Hence, the set of such triples is an exact cover for instance I of RX3C.

The number of nodes in the digraph G is $(\ell+1) \cdot |X| + 2 \cdot |\mathcal{T}| = (\ell+3) \cdot |X|$ and the number of arcs is $\ell \cdot |X| + 4 \cdot |\mathcal{T}| = (\ell+4) \cdot |X|$. Also, there are $|\mathcal{T}| = |X|$ paths in $\mathcal{P}$. Thus, the space and the computation time required to construct I' from I are $O(\ell \cdot |X|) = O(|X|^{c+1})$. Furthermore, the reduction is polynomial and, since RX3C is NP-complete [8], PCEC is NP-hard.

In addition, suppose that PCEC admits a polynomial-time $|\mathcal{P}|^\theta$-approximation algorithm for some constant $\theta > 0$. Then, by setting $c = \lceil \theta + 1 \rceil$ and considering an optimal solution $\mathcal{S}^*$ for instance I' of PCEC, such an algorithm finds a solution $\mathcal{S}$ that satisfies

$$|\mathcal{S}| \leq |\mathcal{P}|^\theta \cdot \left|\mathcal{S}^*\right| = |\mathcal{T}|^\theta \cdot \left|\mathcal{S}^*\right| = |X|^\theta \cdot \left|\mathcal{S}^*\right|.$$

If there is an exact cover for instance I of RX3C, then $|\mathcal{S}^*| \leq 3 \cdot |X|$ and

$$|\mathcal{S}| \leq |X|^\theta \cdot \left|\mathcal{S}^*\right| \leq 3 \cdot |X|^{\theta+1}.$$

If there is an element $x \in X$ whose element-path is not a subpath of some triple-path of $\mathcal{S}$, then each arc of the element-path of x will be in $\mathcal{S}$ and

$$|\mathcal{S}| > 3 \cdot |X|^{\lceil \theta+1 \rceil} \geq 3 \cdot |X|^{\theta+1},$$

a contradiction.

Thus, all element-paths must be subpaths of a triple-path in $\mathcal{S}$. Hence, the triples associated with triple-paths in $\mathcal{S}$ form an exact cover for instance I of RX3C, implying that there exists a polynomial-time algorithm for RX3C. Therefore, unless P = NP, given any positive constant θ, PCEC does not admit a polynomial-time algorithm with approximation factor less than or equal to $|\mathcal{P}|^\theta$. Recall that the digraph G constructed in the reduction is acyclic. Also, note that each node in G has degree at most 4. So this result holds even for acyclic digraphs with maximum degree 4. □

For the undirected case, the previous proof is also valid. The only part that would not apply is the statement that the constructed graph is acyclic. So, an analogous result can be expressed for the undirected PCEC as follows.

Theorem 2.2. *Undirected* PCEC *is NP-hard and, denoting by $I = \langle G, \mathcal{P} \rangle$ an arbitrary instance, there is no polynomial-time $|\mathcal{P}|^{O(1)}$-approximation for undirected* PCEC *unless P = NP, even when restricted to graphs with maximum degree 4.*

Hence, in general, PCEC seems very hard, even in approximation terms. Fortunately, as we will see, κ-PSEC seems more tractable in terms of approximations. For κ-PSEC, the NP-hardness also comes from a reduction from RX3C and is given by the following theorem.

Theorem 2.3. κ-PSEC *is NP-hard on acyclic digraphs with $k = 5$ and maximum degree 6.*

Proof. Let $I = \langle X, \mathcal{T} \rangle$ be an instance of RX3C. Again, without loss of generality, assume that the elements of X are ordered and, for each triple in $\mathcal{T}$, the three elements maintain the order they have in X. An instance $I' = \langle G, \mathcal{P}, k \rangle$ of κ-PSEC can be constructed as follows.

For each element $x \in X$, the digraph G contains four nodes x_o, x_d, x_1, and x_2, with arcs from x_o, x_1, and x_2 to x_d as in Fig. 2(a). This forms the **element-gadget** for x. The arc from x_o to x_d is called an **element-arc**, and the arcs from x_1 and x_2 to x_d are called **triple-arcs**. In addition to the arcs in element-gadgets, for each triple $\langle x, y, z \rangle \in \mathcal{T}$, there are two more arcs in G, one from x_d to y_o and one from y_d to z_o. These are the **connecting-arcs**. Different triples might share some of the connecting-arcs, so that there are no parallel arcs in the digraph. Fig. 2(b) illustrates an instance of RX3C and the corresponding digraph G.

Because, for each triple in $\mathcal{T}$, the three elements maintain the order they have in X, the constructed digraph G is acyclic. Also, each node has at most three incoming arcs and at most three outgoing arcs.

Take $k = 5$ and define the set $\mathcal{P}$ of paths in G as follows. For each triple $t = \langle x, y, z \rangle \in \mathcal{T}$, there are seven paths in $\mathcal{P}$ (Fig. 3 illustrates them):

- The **covering-path of** t, that has length 5 and passes through the element-arcs of x, y, and z ($x_o x_d y_o y_d z_o z_d$).
- Six **non-covering-paths of** t: four with a triple-arc and the following connecting-arc of the triple, and two with only the triple-arcs for the last element of the triple ($x_1 x_d y_o$, $x_2 x_d y_o$, $y_1 y_d z_o$, $y_2 y_d z_o$, $z_1 z_d$, and $z_2 z_d$, for instance).

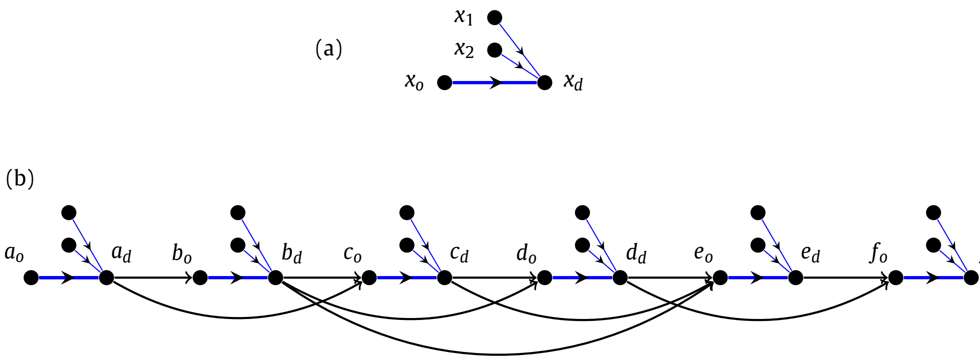

Fig. 2. (a) The element-gadget for an $x \in X$. (b) The digraph G for the instance $\langle X, \mathcal{T} \rangle$ of RX3C with $X = \{a, b, c, d, e, f\}$ and $\mathcal{T} = \{abc, abd, ace, bef, cdf, def\}$. The element-gadgets are marked in blue. (For interpretation of the colors in the figures, the reader is referred to the web version of this article.)

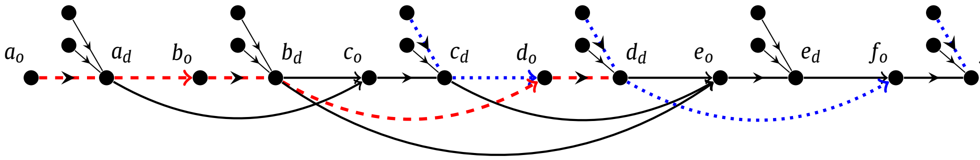

Fig. 3. The digraph for the instance $\langle X, \mathcal{T} \rangle$ of RX3C with $X = \{a, b, c, d, e, f\}$ and $\mathcal{T} = \{abc, abd, ace, bef, cdf, def\}$. In dashed red, the covering-path for the triple *abd* and, in dotted blue, three of the six non-covering-paths for the triple *cdf*.

Note that arcs that are not within element-gadgets might be in several covering- and non-covering-paths. On the other hand, triple-arcs only lie in non-covering-paths from $\mathcal{P}$. Also, all arcs in G are in some path from $\mathcal{P}$, and I' is a valid instance of κ-PSEC. Curiously, all paths in $\mathcal{P}$ have length at most 5.

Let $\mathcal{S}$ denote an arbitrary feasible solution for I'. No two triple-arcs belong to the same path of $\mathcal{P}$, hence each triple-arc lies in a different path in $\mathcal{S}$. Also, only covering-paths cover element-arcs, and each such path covers three such arcs. There are $2 \cdot |X|$ triple-arcs and $|X|$ element-arcs, therefore $|\mathcal{S}| \geq 2 \cdot |X| + |X|/3 = 7 \cdot |X|/3$.

Below, it is proved that there exists an exact cover for instance I of RX3C if and only if there exists a solution whose cardinality is exactly $7 \cdot |X|/3$ for instance I' of κ-PSEC.

If $T \subseteq \mathcal{T}$ is a solution for instance I of RX3C, then a solution $\mathcal{S}$ for instance I' of κ-PSEC can be constructed as follows:

Step 1: For each triple $t \in T$, the covering-path of t is added to $\mathcal{S}$.
Step 2: For each element $x \in X$, two paths are added to $\mathcal{S}$. There are two triples t_1, t_2 in $\mathcal{T} \setminus T$ that contain x. If x is the last element in t_1, add the path $x_1 x_d$ to $\mathcal{S}$. If x is not the last element in t_1, let y be the next element in t_1 and add the path $x_1 x_d y_o$ to $\mathcal{S}$ if the connecting-arc $x_d y_0$ is not in an element-path in T; otherwise add the path $x_1 x_d$ to $\mathcal{S}$. Do the same with the triple t_2 but using x_2 instead of x_1.

It is not hard to see that all paths included in $\mathcal{S}$ are pairwise arc-disjoint. Also, all the element-arcs are covered in Step 1 and all the triple-arcs are covered in Step 2. Moreover, the connecting-arcs are either covered in Step 1 by a covering-path, or covered in Step 2 by a non-covering-path. So $\mathcal{S}$ covers exactly once all arcs of G, and is therefore a feasible solution for κ-PSEC.

There is one path for each $t \in T$, and one path for each triple-arc. Hence, the number of paths included in $\mathcal{S}$ is exactly $|X|/3 + 2 \cdot |X| = 7 \cdot |X|/3$.

For the other direction, suppose that $\mathcal{S}$ is a solution for instance I' of κ-PSEC with $|\mathcal{S}| = 7 \cdot |X|/3$. There are $2 \cdot |X|$ paths in $\mathcal{S}$, one covering each triple-arc. All such paths are subpaths of non-covering-paths, and therefore do not cover any element-arc. Hence, there are at most $|X|/3$ paths in $\mathcal{S}$ to cover the $|X|$ element-arcs. As each covering-path covers three element-arcs, and no proper subpath of a covering-path covers three element-arcs, we conclude that $\mathcal{S}$ must contain exactly $|X|/3$ covering-paths. These paths must be disjoint, hence the triples corresponding to them define an exact cover of X for instance I of RX3C.

The instance I' has $4 \cdot |X|$ nodes, $3 \cdot |X|$ arcs within element-gadgets, plus at most two connecting-arcs per triple, for a total of at most $5 \cdot |X|$ arcs, as there are $|X|$ triples in every instance of RX3C. There are seven paths per triple in $\mathcal{P}$, for a total of $7 \cdot |X|$ paths, being $|X|$ of length 5, and $6 \cdot |X|$ of length 1 or 2. Thus, the size of I' is $O(|X|)$, which is also the computation time required to construct instance I' from I. Consequently, the described reduction from RX3C to κ-PSEC is polynomial and κ-PSEC is NP-hard. Moreover, as G is acyclic, this result is valid even for acyclic digraphs, with each node having at most three incoming arcs and at most three outgoing arcs. □

Analogously to PCEC, the result holds also for the undirected κ-PSEC.

Theorem 2.4. *Undirected* κ-PSEC *is NP-hard on graphs with* $k = 5$ *maximum degree 6.*

3. Polynomial-time algorithms for special cases

As shown before, even for acyclic digraphs, there are no polynomial-time algorithms for PCEC or κ-PSEC unless P = NP, henceforth this section considers subclasses of digraphs to propose *efficient* algorithms for these problems.

The **underlying graph** of a digraph G is the (undirected simple) graph H that results from replacing each directed arc of G with an undirected edge, that is, $V_H = V_G$ and $A_H = \{\{u, v\} : \langle u, v\rangle \in A_G \text{ or } \langle v, u\rangle \in A_G\}$. A **polytree** is an acyclic digraph whose underlying graph is a tree. In particular, the class of polytrees contains all rooted trees (i.e., an acyclic digraph with a special node called **root**, such that there exists exactly one path from the root to any other node). The following theorem gives a polynomial-time algorithm for PCEC on polytrees.

Theorem 3.1. *Any instance* $I = \langle T, \mathcal{P}\rangle$ *of* PCEC *where* T *is a polytree on* n *nodes can be solved in* $O((n + |\mathcal{P}|)^2 + n^{\frac{7}{2}} \cdot \log n)$ *time.*

Proof. PCEC on polytrees has the so called optimal substructure [9], so the dynamic programming approach will apply. The idea is as follows. Let $I = \langle T, \mathcal{P}\rangle$ be an instance of PCEC where T is a polytree. Root T at an arbitrary vertex r, regardless of the arcs orientations. Add to $\mathcal{P}$ all paths with a single arc of T so that any solution is a subset of $\mathcal{P}$.

Consider an optimal solution $\mathcal{S} \subseteq \mathcal{P}$ for I. Let Q be the path in $\mathcal{S}$ that uses r. Each path Q determines a forest $F_Q = T - A_Q$, where A_Q denotes the set of arcs of Q.

Because the paths in $\mathcal{S}$ are arc-disjoint, each path in $\mathcal{S} \setminus \{Q\}$ lies in one of the components of F_Q. For each non-trivial component C of F_Q, let $\mathcal{P}_C$ be the set of paths in $\mathcal{P}$ that lie in C of F_Q, and $\mathcal{S}_C$ be the set of paths in $\mathcal{S}$ that lie in C of F_Q. Note that $\mathcal{S}_C$ is an optimal solution for the instances $I_C(Q) = \langle C, \mathcal{P}_C\rangle$. Otherwise, $\mathcal{S}$ would not be an optimal solution for I. This is the key observation to the dynamic programming algorithm that follows.

For every node v, let T_v denote the subtree of T rooted at v. The idea is, for every path $Q \in \mathcal{P}$ containing r, to compute an optimal solution of the instances $I_C(Q)$ for each non-trivial component C of F_Q that lies in T_v for some child v of r. With these solutions, one can obtain an optimal solution for I as follows.

For short, a **child component** of F_Q is a component C of F_Q that lies in some T_v for a child v of r. First assume that the number of children of r is even. Construct a weighted complete undirected graph H with a vertex for each child of r in T. To define the weight of an edge xy, first define two numbers w_1 and w_2. The number w_1 is the minimum, taken over all paths $Q \in \mathcal{P}$ going through xry (or yrx), of the sum of the solution sizes for all instances $I_C(Q)$ where C is a non-trivial child component of F_Q. Consider $w_1 = \infty$ if no such path Q exists. The number w_2 is the minimum, taken over all paths Q_x and Q_y such that Q_x starts or ends at r and goes through x, and Q_y starts or ends at r and goes through y, of the sum of the solution sizes for all instances $I_C(Q)$ where C is a non-trivial child component of F_{Q_x} or a non-trivial child component of F_{Q_y}. Note that w_2 is finite because $\mathcal{P}$ contains all paths with a single arc. The weight of xy is the minimum between w_1 and w_2. If the number of children of r is odd, a similar construction can be done with the addition of a dummy vertex to H, to be matched to one of the children of r. A minimum weight perfect matching in H gives us a solution for instance I.

To solve one such instance $I_C(Q)$, the idea is similar. Let v be the root of C. Construct a weighted complete undirected graph $H_C(Q)$ with a vertex for each child of v in C. Define the weight of an edge xy in $H_C(Q)$ exactly as above. A minimum weight perfect matching in $H_C(Q)$ gives us a solution for instance $I_C(Q)$.

The number of distinct such components C rooted at v that might be generated over the whole process is at most $d(v) + 1$, where $d(v)$ is the degree of v in T: one with all children of v in T, and one with exactly a missing child of v. Thus, one can compute optimal solutions for $I_C(Q)$ in a bottom-up way, to eventually obtain an optimal solution for I. The total number of minimum weighted perfect matchings to be computed is at most $\sum_v (d(v) + 1) = O(n)$.

For a complete graph on n nodes, with edge weights bounded by n, it is possible to compute a minimum weight perfect matching, if one exists, in time $O(n^{\frac{5}{2}} \cdot \log n)$ [10]. For each path Q, the number of nodes in each graph $H_C(Q)$ is at most n, where n is the number of nodes in T. The edge weights are sizes of sets of arc-disjoint paths in the tree T, therefore they also have value less than n. Hence the overall time on minimum perfect matching computations is $O(n \cdot n^{\frac{5}{2}} \cdot \log n)$.

For each path $Q \in \mathcal{P}$, using a tree traversal algorithm, one can compute F_Q, determine its children components, and determine $\mathcal{P}_C$ for each children component C in linear time, that is, in $O(n + |\mathcal{P}|)$ time. For the latter, note that a path lies in a component if and only if both its ends lie in the component.

Recall that we added to $\mathcal{P}$ the $n - 1$ arcs of G. Hence, in terms of the original $\mathcal{P}$, the total time spent with the minimum perfect matching computations is $O((n + |\mathcal{P}|)^2 + n^{\frac{7}{2}} \cdot \log n)$, and this is the computation time to solve the instance I of PCEC. □

We can improve the computation time achieved by Theorem 3.1 for different particular subclasses of graphs. Some of them will be used to obtain approximation algorithms in the next section. Below we show how to obtain these improvements.

Rooted trees are a particular case of the graphs considered in Theorem 3.1 for which no matching computation is required, because every arc is directed away from the root. Thus, the computation time boils down to the construction of the smaller instances, which in total takes quadratic time.

Corollary 3.1. *Any instance $I = \langle T, \mathcal{P} \rangle$ of* PCEC*, where T is a rooted tree on n nodes, can be solved in* $\mathrm{O}((n + |\mathcal{P}|)^2)$ *time.*

Directed paths are rooted trees where each node, except for the last one, has exactly one child. Hence, for each path in $\mathcal{P}$, there is at most one previously computed value of interest, and there is always at most one children component, which is easily determined. Consequently, any instance $I = \langle G, \mathcal{P} \rangle$ of PCEC where G is a directed path on n nodes can be solved in $\mathrm{O}(n + |\mathcal{P}|)$ time. Building on this, for an instance where the underlying graph of G is a path $v_1 v_2 \ldots v_n$, an optimal solution can be computed by joining optimal solutions for instances on directed paths. Specifically, let $\vec{G}$ be the subgraph of G with only the existing arcs from v_i to v_{i+1} for $i = 1, \ldots, n-1$, and $\vec{\mathcal{P}}$ be the set of paths in $\mathcal{P}$ that are in $\vec{G}$. Define $\overleftarrow{G}$ and $\overleftarrow{\mathcal{P}}$ symmetrically, and consider the instances $\vec{I} = \langle \vec{G}, \vec{\mathcal{P}} \rangle$ and $\overleftarrow{I} = \langle \overleftarrow{G}, \overleftarrow{\mathcal{P}} \rangle$. Note that there is no path in $\mathcal{P}$ with arcs from $\vec{G}$ and $\overleftarrow{G}$, so $\vec{\mathcal{P}}$ and $\overleftarrow{\mathcal{P}}$ partition the set $\mathcal{P}$. The digraphs $\vec{G}$ and $\overleftarrow{G}$ are collections of arc-disjoint directed paths, implying that optimal solutions for $\vec{I}$ and $\overleftarrow{I}$ can be obtained by using the idea previously described over each directed path separately. The overall time complexity required to divide I into the directed path instances and to solve them is $\mathrm{O}(n + |\mathcal{P}|)$.

Corollary 3.2. *Any instance $I = \langle G, \mathcal{P} \rangle$ of* PCEC *where the underlying graph of G is a path on n nodes can be solved in* $\mathrm{O}(n + |\mathcal{P}|)$ *time.*

Consider an instance $I = \langle G, \mathcal{P} \rangle$ of PCEC where G is a directed cycle on n nodes. If $n < |\mathcal{P}|$, then let $I_v = \langle G, \mathcal{P}_v \rangle$ be the instance of PCEC where $\mathcal{P}_v$ is the subset of $\mathcal{P}$ with all paths that do not contain v as an internal node. (One can decide whether a path from $\mathcal{P}$ is in $\mathcal{P}_v$ in $\mathrm{O}(1)$ time, by looking at the first, second, and last of its nodes.) Each such instance I_v behaves as if G were a path, and so it can be solved as in Corollary 3.2 in $\mathrm{O}(|\mathcal{P}|)$ time, because $n < |\mathcal{P}|$. Let u be such that an optimal solution $\mathcal{S}$ for I_u has minimum size among all instances I_v. Such $\mathcal{S}$ is an optimal solution for I, and can be computed in $\mathrm{O}(n \cdot |\mathcal{P}|)$ time. On the other hand, if $|\mathcal{P}| < n$, then, for each path $Q \in \mathcal{P}$, let $I_Q = \langle G_Q, \mathcal{P}_Q \rangle$ be the instance of PCEC where G_Q is G after the removal of the internal nodes of Q, and $\mathcal{P}_Q$ is the subset of $\mathcal{P}$ with all paths that do not contain any of the removed nodes. Note that each G_Q is a path, so we can compute an optimal solution $\mathcal{S}_Q$ for each I_Q by Corollary 3.2 in $\mathrm{O}(n)$ time, because $|\mathcal{P}| < n$. Let Q be such that $\mathcal{S}_Q$ has minimum size among all $Q \in \mathcal{P}$. Thus $\mathcal{S} = \mathcal{S}_Q \cup \{Q\}$ is an optimal solution for I, and it can be computed again in $\mathrm{O}(n \cdot |\mathcal{P}|)$ time.

Furthermore, if the underlying graph of G is a cycle on n nodes, then proceeding as in Corollary 3.2, an instance $I = \langle G, \mathcal{P} \rangle$ of PCEC can be solved by joining optimal solutions for instances $\vec{I} = \langle \vec{G}, \vec{\mathcal{P}} \rangle$ and $\overleftarrow{I} = \langle \overleftarrow{G}, \overleftarrow{\mathcal{P}} \rangle$. In this case, $\vec{I}$ and $\overleftarrow{I}$ might be directed cycles or sets of disjoint directed paths, hence these instances can be solved in $\mathrm{O}(n \cdot |\mathcal{P}|)$ time.

Corollary 3.3. *Any instance $I = \langle G, \mathcal{P} \rangle$ of* PCEC *where the underlying graph of G is a cycle on n nodes can be solved in* $\mathrm{O}(n \cdot |\mathcal{P}|)$ *time.*

The previous ideas allow us to obtain a polynomial-time algorithm for another class of graphs, the so called **pseudo-rooted-trees** (that is, a rooted tree with at most an extra arc $\langle u, v \rangle$ that forms a directed cycle). For each path of $\mathcal{P} \cup \{\langle u, v \rangle\}$ containing $\langle u, v \rangle$, we solve the instance resulting from the removal of its arcs, and select the best solution. The graphs of such instances are forests of rooted trees. Hence, by applying Corollary 3.1, we derive the following.

Corollary 3.4. *Any instance $I = \langle G, \mathcal{P} \rangle$ of* PCEC *where G is a pseudo-rooted-tree on n nodes can be solved in* $\mathrm{O}(|\mathcal{P}| \cdot (n + |\mathcal{P}|)^2)$ *time.*

For κ-PSEC, the proposal is to use the above strategies after transforming the problem into PCEC. The reduction, from an instance $I = \langle G, \mathcal{P}, k \rangle$ of κ-PSEC, builds an instance $I' = \langle G, \mathcal{P}' \rangle$ of PCEC where $\mathcal{P}'$ is the set of all subpaths of length at most k of the paths in $\mathcal{P}$. If G is a directed path or cycle with n nodes, then $|\mathcal{P}'| \le k \cdot n$ because there are no more than k subpaths of length at most k starting at each node. For more general digraphs, the number of subpaths in $\mathcal{P}'$ is $\mathrm{O}(k \cdot n \cdot |\mathcal{P}|)$, because each path in $\mathcal{P}$ has $\mathrm{O}(k \cdot n)$ subpaths of length at most k. Thus, the previous strategies to solve PCEC, applied for solving κ-PSEC, lead to the following corollary.

Corollary 3.5. *Let $I = \langle G, \mathcal{P}, k \rangle$ be an instance of* κ-PSEC*.*

(a) *If G is a polytree on n nodes, then* κ-PSEC *on I can be solved in* $\mathrm{O}((k \cdot n \cdot |\mathcal{P}|)^2 + n^{\frac{7}{2}} \cdot \log n))$ *time.*
(b) *If G is a rooted tree on n nodes, then* κ-PSEC *on I can be solved in* $\mathrm{O}((k \cdot n \cdot |\mathcal{P}|)^2)$ *time.*
(c) *If the underlying graph of G is a path on n nodes, then* κ-PSEC *on I can be solved in* $\mathrm{O}(k \cdot n)$ *time.*
(d) *If the underlying graph of G is a cycle on n nodes, then* κ-PSEC *on I can be solved in* $\mathrm{O}(k \cdot n^2)$ *time.*
(e) *If G is a pseudo-rooted-tree on n nodes, then* κ-PSEC *on I can be solved in* $\mathrm{O}((k \cdot n \cdot |\mathcal{P}|)^3)$ *time.*

If the orientation of the arcs is removed, then the above findings remain the same for trees, paths, and cycles, hence the results below follow for the undirected variants of the problems.

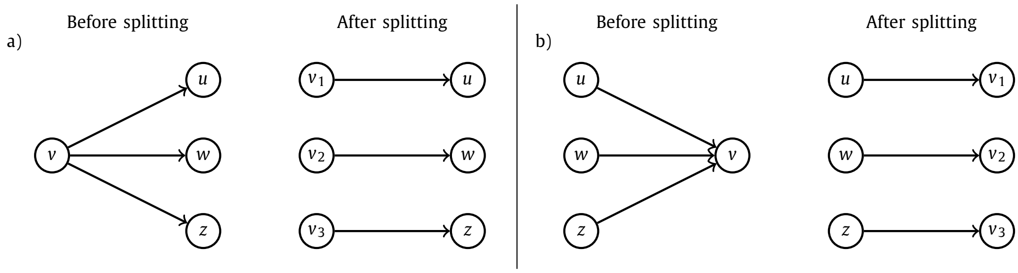

Fig. 4. a) Splitting of node v with three outgoing arcs; b) Splitting of node v with three incoming arcs.

Remark 3.1. Let $I = \langle G, \mathcal{P}, k\rangle$ be an instance of the undirected PCEC.

(a) If G is a tree on n nodes, then the undirected PCEC on I can be solved in $O((n + |\mathcal{P}|)^2 + n^{\frac{7}{2}} \cdot \log n))$ time.
(b) If G is a path on n nodes, then the undirected PCEC on I can be solved in $O(n + |\mathcal{P}|)$ time.
(c) If G is a cycle on n nodes, then the undirected PCEC on I can be solved in $O(n \cdot |\mathcal{P}|)$ time.

Remark 3.2. Let $I = \langle G, \mathcal{P}, k\rangle$ be an instance of the undirected κ-PSEC.

(a) If G is a tree on n nodes, then the undirected κ-PSEC on I can be solved in $O((k \cdot n \cdot |\mathcal{P}|)^2 + n^{\frac{7}{2}} \cdot \log n)$ time.
(b) If G is a path on n nodes, then the undirected κ-PSEC on I can be solved in $O(k \cdot n)$ time.
(c) If G is a cycle on n nodes, then the undirected κ-PSEC on I can be solved in $O(k \cdot n^2)$ time.

4. Approximation algorithms for special cases of κ-PSEC

The strong inapproximability result given by Theorem 2.1 for PCEC inhibits the search for efficient algorithms even to approximate the problem. However, for some special cases of κ-PSEC, we show that there are polynomial-time approximation algorithms.

4.1. Graphs with maximum degree at most 3

This section considers instances of κ-PSEC in which any node of the digraph has at most three neighbors. The idea to find approximate solutions for these instances is to transform the digraph into a case that can be efficiently solved (as the ones shown in Section 3), proving that the solution obtained from such a transformation achieves some approximation ratio. The following theorem shows the first result in that direction.

Theorem 4.1. *There exists a $\frac{3}{2}$-approximation for* κ-PSEC *on instances $I = \langle G, \mathcal{P}, k\rangle$ where G has n nodes and $\Delta(G) \leq 3$, that runs in* $O((k \cdot n \cdot |\mathcal{P}|)^3)$ *time.*

Proof. Consider an instance $I = \langle G = \langle V, A\rangle, \mathcal{S}, k\rangle$ of κ-PSEC where $\Delta(G) \leq 3$. The approximation idea is to split some of the nodes with degree at least two, in order to obtain a new graph where all nodes with degree three have two incoming and one outgoing arcs, or all these nodes have one incoming and two outgoing arcs. Such a graph will consist of components that are pseudo-rooted-trees and the corresponding instance can be efficiently solved by Corollary 3.4. The splitting will be done in such a way that will guarantee that an optimal solution for the resulting instance is not much larger than the original optimal value. This will be done by in fact producing two smaller instances, one of which will have this property. So by solving both and returning the smallest of the two obtained solutions, we will achieve the claimed approximation ratio.

First, any node $v \in V$ with three outgoing arcs or with three incoming arcs is split into three new nodes v_1, v_2, and v_3, each one with only one of the arcs. Fig. 4 shows this splitting process. The same is done with any node with two outgoing arcs and no incoming arc, or with two incoming arcs and no outgoing arc.

After repeatedly applying the above splitting process to G, a new graph G' results, where no node with degree at least two has only outgoing arcs or only incoming arcs. Also, a new set of paths $\mathcal{P}'$ is obtained by replacing each split node v by the corresponding node v_1, v_2, or (possibly) v_3.

If $\mathcal{S}$ is a feasible solution for instance I of κ-PSEC, then a feasible solution $\mathcal{S}'$ for instance $I' = \left\langle G', \mathcal{P}', k\right\rangle$ of κ-PSEC is obtained by replacing each node v by its corresponding v_1, v_2, or (possibly) v_3 in the paths of $\mathcal{S}$, and $|\mathcal{S}| = |\mathcal{S}'|$. The other direction (obtaining $\mathcal{S}$ from $\mathcal{S}'$) is also valid. In particular, optimal solutions for I and I' have the same size.

The digraph G' may still contain nodes with degree three, having two outgoing (incoming) arcs and one incoming (outgoing) arc. The splitting of some of these nodes will generate two new digraphs G_1 and G_2.

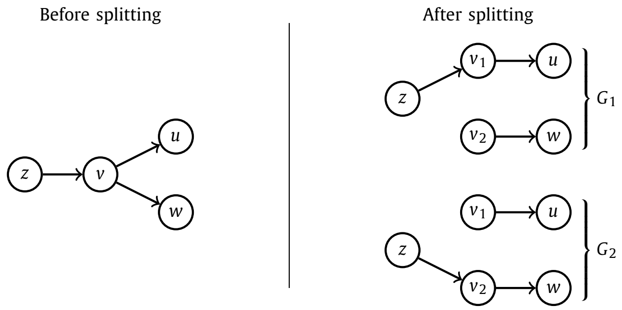

Fig. 5. Splitting of node v with one incoming arc and two outgoing arcs.

Without loss of generality, assume the number t^+ of nodes having two outgoing arcs and one incoming arc is at most as large as the number t^- of nodes with two incoming arcs and one outgoing arc, that is, $t^+ \leq t^-$. (The analysis if $t^- < t^+$ is analogous.) Each node v of G' with two outgoing arcs and one incoming arc is split into two new nodes v_1 and v_2, each one with only one of the outgoing arcs of v. The digraph G_1 is the copy of G' after the splitting of v in which v_1 also receives the incoming arc of v, while, in G_2, v_2 is the node receiving such an arc. Fig. 5 illustrates this splitting method.

After splitting each node of G' with two outgoing arcs and one incoming arc, any component of G_1 and G_2 will contain at most one cycle. Now, we define two new sets of paths $\mathcal{P}_1$ and $\mathcal{P}_2$ corresponding to G_1 and G_2. For each path $Q \in \mathcal{P}'$, if Q does not contain any split node, then Q is added to $\mathcal{P}_1$ and to $\mathcal{P}_2$. Otherwise, suppose v is as in Fig. 5 and v is in a path $Q \in \mathcal{P}'$. If Q ends at v, then the corresponding path in $\mathcal{P}_1$ has v_1 in the place of v, and the corresponding path in $\mathcal{P}_2$ has v_2 in the place of v (Fig. 6(a)). If Q starts at v and goes through $\langle v, u\rangle$, then substitute v by v_1 in the corresponding paths in $\mathcal{P}_1$ and $\mathcal{P}_2$ (Fig. 6(b)). If Q starts at v and goes through $\langle v, w\rangle$, then substitute v by v_2 in the corresponding paths in $\mathcal{P}_1$ and $\mathcal{P}_2$ (Fig. 6(c)). Finally, if v is an internal node in a path $Q \in \mathcal{P}'$, then Q uses arc $\langle z, v\rangle$ and proceeds through either arc $\langle v, u\rangle$ or arc $\langle v, w\rangle$. If Q proceeds through $\langle v, u\rangle$, then substitute v by v_1 in the corresponding path in $\mathcal{P}_1$, and split the corresponding path in $\mathcal{P}_2$ at v, substituting v by v_2 in the part that ends at v, and by v_1 in the part that starts at v (Fig. 6(d)). If Q proceeds through $\langle v, w\rangle$, then do the opposite: split the corresponding path in $\mathcal{P}_1$ at v, substituting v by v_1 in the part that ends at v, and by v_2 in the part that starts at v, and substitute v by v_2 in the corresponding path in $\mathcal{P}_2$ (Fig. 6(e)).

From any solution $\mathcal{S}'$ for instance I', one can construct solutions $\mathcal{S}_1$ and $\mathcal{S}_2$ for instances $I_1 = \langle G_1, \mathcal{P}_1, k\rangle$ and $I_2 = \langle G_2, \mathcal{P}_2, k\rangle$ of κ-PSEC as follows. For every path $Q \in \mathcal{S}'$, include in $\mathcal{S}_1$ and $\mathcal{S}_2$ the paths from $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively, corresponding to Q. Observe that each path $Q \in \mathcal{S}'$ is only split at internal nodes, and if Q is split at a node v for $\mathcal{S}_1$, then Q is not split at v for $\mathcal{S}_2$. Moreover, because the split nodes have two outgoing arcs and one incoming arc, each split node cannot be an internal node in more than one path in $\mathcal{S}'$. Consequently, $|\mathcal{S}_1| + |\mathcal{S}_2| \leq 2 \cdot \left|\mathcal{S}'\right| + t^+$. If $V^3_{G'}$ denotes the set of nodes in G' with degree three, then $t^+ + t^- = \left|V^3_{G'}\right|$ and $t^+ \leq \left|V^3_{G'}\right|/2$, because $t^+ \leq t^-$. Hence, $|\mathcal{S}_1| + |\mathcal{S}_2| \leq 2 \cdot \left|\mathcal{S}'\right| + \left|V^3_{G'}\right|/2$.

Let $\mathcal{S}^*$, $\mathcal{S}'^*$, $\mathcal{S}^*_1$, and $\mathcal{S}^*_2$ be optimal solutions for instances I, I', I_1, and I_2 of κ-PSEC, respectively. Recall that $|\mathcal{S}^*| = \left|\mathcal{S}'^*\right|$. Also, if $\bar{\mathcal{S}}_1$ and $\bar{\mathcal{S}}_2$ are the solutions for I_1 and I_2 constructed from $\mathcal{S}'^*$, then

$$|\mathcal{S}^*_1| + |\mathcal{S}^*_2| \;\leq\; |\bar{\mathcal{S}}_1| + |\bar{\mathcal{S}}_2| \;\leq\; 2 \cdot |\mathcal{S}'^*| + \frac{\left|V^3_{G'}\right|}{2} \;=\; 2 \cdot |\mathcal{S}^*| + \frac{\left|V^3_{G'}\right|}{2} \;\leq\; 3 \cdot |\mathcal{S}^*|.$$

The last inequality comes from the fact that $\left|V^3_{G'}\right| \leq 2 \cdot \left|\mathcal{S}'^*\right|$, because every node in $V^3_{G'}$ has degree three in G' and must be the first or last node of at least one path in $\mathcal{S}'^*$. Furthermore, because the minimum between two values is less than or equal to their average value, it follows that

$$\min\left\{\left|\mathcal{S}^*_1\right|, \left|\mathcal{S}^*_2\right|\right\} \;\leq\; \frac{3}{2} \cdot \left|\mathcal{S}^*\right|.$$

Notice that, by replacing the new nodes with the corresponding original ones in each path of $\mathcal{S}^*_1$ and $\mathcal{S}^*_2$, new feasible solutions for κ-PSEC with instance I are obtained, and the cardinality of such solutions is the same as the cardinality of $\mathcal{S}^*_1$ and $\mathcal{S}^*_2$. Hence, the minimum between an optimal solution of κ-PSEC with I_1 and I_2 gives a $\frac{3}{2}$-approximation for κ-PSEC with instance I.

This approximation can be performed in $O((k \cdot n \cdot |\mathcal{P}|)^3)$ computation time, where $n = |V|$. Transforming I into I' takes $O(|\mathcal{P}| + n)$ time, because at most n nodes need to be split into three nodes each, and only ends of paths of $\mathcal{P}$ need to be adjusted. The construction of I_1 and I_2 produces new graphs with the same number of arcs and at most twice as many nodes as G', while each path $Q \in \mathcal{P}'$ can be split into $O(|Q|) = O(n)$ subpaths, being $O(n \cdot \left|\mathcal{P}'\right|) = O(n \cdot |\mathcal{P}|)$ the time complexity for this construction. When $t^+ \leq t^-$, all nodes in G_1 and G_2 with degree greater than two have two outgoing arcs and one incoming arc, and there is no node with two incoming arcs. So each component of G_1 and G_2 is a pseudo-rooted-tree. Therefore, by Corollary 3.5, instances I_1 and I_2 of κ-PSEC can be optimally solved in $O((k \cdot n \cdot |\mathcal{P}|)^3)$ time. The case in which $t^- < t^+$ is symmetric, and can also be solved in $O((k \cdot n \cdot |\mathcal{P}|)^3)$ time. □

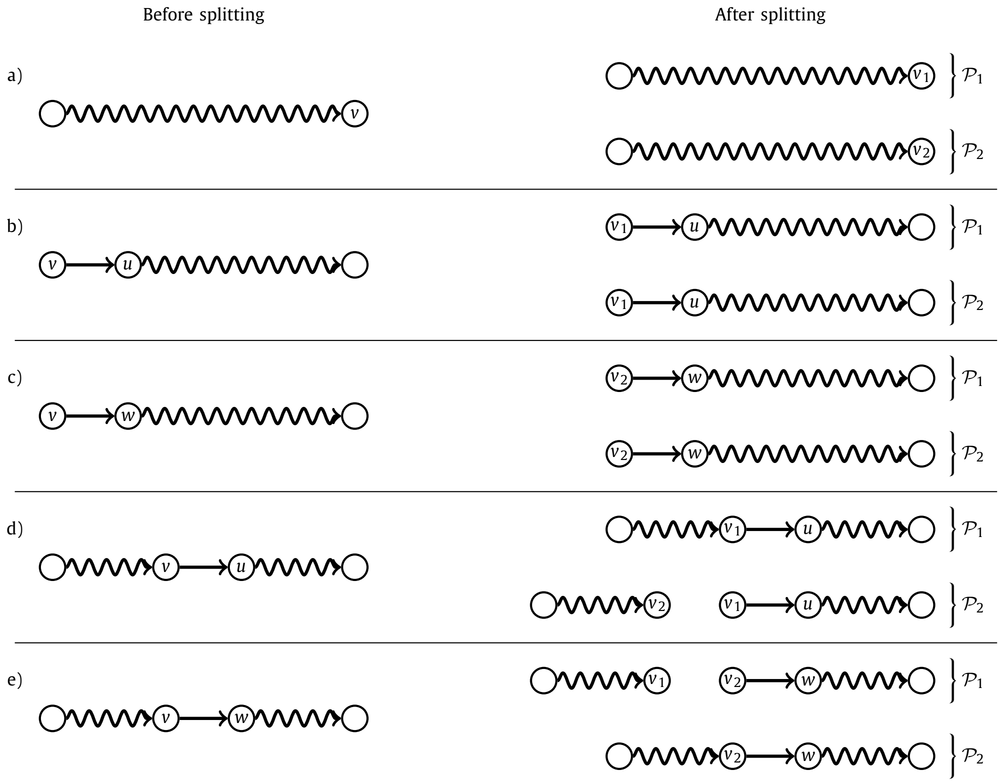

Fig. 6. Splitting of a path Q that contains the split node v with one incoming arc and two outgoing arcs. The cases are: a) Q ends at v; b) Q starts at v and goes through $\langle v, u\rangle$; c) Q starts at v and goes through $\langle v, w\rangle$; d) Q contains $\langle v, u\rangle$; and e) Q contains $\langle v, w\rangle$.

4.2. Approximation algorithms based on maximum matchings

This section proposes approximating κ-PSEC by solving a maximum matching problem on the arcs. The main idea is to cover the arcs of the graph by as many paths with exactly two arcs as possible. This approach allows to obtain approximation ratios depending on the value of k and also on the maximum degree of graphs where all nodes have odd degree.

We start by noting that κ-PSEC remains NP-hard if all nodes in the graph have odd degree. Indeed, one can reduce the general case of κ-PSEC to this one. For this, it is enough to add a new node adjacent to each even degree node, and to include one path made of each added arc. (This reduction however does not preserve approximation ratio.)

Also, any feasible solution for I has at least $|A|/k$ and at most $|A|$ paths (one path for each arc of A). Hence, any feasible solution is already a k-approximation on the optimal value. Next we deduce an improvement of almost a half over this trivial approximation factor.

Given an instance $I = \langle G = \langle V, A\rangle, \mathcal{P}, k\rangle$ of κ-PSEC, consider an undirected graph $G' = \langle V', A'\rangle$ whose nodes are the arcs of G (i.e., $V' = A$), and there is an edge between each pair of arcs that are consecutive in some path of $\mathcal{P}$. If M^* is a maximum matching in G', then each edge in M^* defines a path covering exactly two arcs of G so that such paths do not have common arcs and they are subpaths of some path in $\mathcal{P}$. Moreover, a feasible solution $\mathcal{S}$ for instance I of κ-PSEC can be constructed from these paths by adding each arc of G not covered by these length-2 paths corresponding to edges in M^*. Hence

$$|\mathcal{S}| = |A| - |M^*|.$$

If $\mathcal{S}^*$ is an optimal solution for instance I of κ-PSEC, then there exists a matching M such that

$$|M| = \sum_{P\in\mathcal{S}^*} \left\lfloor \frac{|A_P|}{2} \right\rfloor \geq \sum_{P\in\mathcal{S}^*} \frac{|A_P|-1}{2} = \frac{|A|-|\mathcal{S}^*|}{2}.$$

Furthermore, because $|M^*| \geq |M|$, it follows that

$$|\mathcal{S}| = |A| - |M^*| \leq |A| - |M| \leq \frac{|A|+|\mathcal{S}^*|}{2}.$$

Observe that $|\mathcal{S}^*| \geq |A|/k$, because each path in $\mathcal{S}^*$ cannot have more than k arcs. Consequently, $|\mathcal{S}| \leq ((k+1)/2)\cdot|\mathcal{S}^*|$ and, because a maximum matching in G' can be found in $O(|A'|\cdot\sqrt{|V'|})$ time [11], the following theorem holds.

Theorem 4.2. *There exists a* $\left(\frac{k+1}{2}\right)$*-approximation for* κ-PSEC *that runs in* $O(|\mathcal{P}| \cdot n \cdot \sqrt{m})$ *time on instances* $I = \langle G, \mathcal{P}, k \rangle$*, where G has n nodes and m arcs.*

Theorem 2.3 showed that κ-PSEC is NP-hard even for instances with fixed $k = 5$, henceforth the above result gives a 3-approximation algorithm for such instances.

Another result can be obtained if each node of G has odd degree. In such case, every node must be the first or last node in a path of $\mathcal{S}^*$, implying that $|\mathcal{S}^*| \geq |V|/2$. Because $\Delta(G) \cdot |V| \geq 2 \cdot |A|$, we derive that $|A| \leq |\mathcal{S}^*| \cdot \Delta(G)$. Thus, the solution $\mathcal{S}$ obtained from a matching as before is such that

$$|\mathcal{S}| \leq \frac{|A| + |\mathcal{S}^*|}{2} \leq \frac{\Delta(G) + 1}{2} \cdot |\mathcal{S}^*|,$$

achieving the result below.

Theorem 4.3. *There exists a* $\left(\frac{\Delta+1}{2}\right)$*-approximation for* κ-PSEC *that runs in* $O(|\mathcal{P}| \cdot n \cdot \sqrt{m})$ *time on instances* $I = \langle G, \mathcal{P}, k \rangle$ *where G has n nodes, m arcs, maximum degree* Δ*, and all nodes have odd degree.*

5. Final comments and future directions

This work introduced PCEC and formalized κ-PSEC as a combinatorial graph covering problem. These problems find practical applicability in network design and network monitoring systems, since their solution may help to obtain a granular visibility of a network in real-time while avoiding its service degradation.

The complexity of PCEC and κ-PSEC were studied and the first proofs on their hardness were presented. Both problems were shown to be NP-hard and, particularly, PCEC is very difficult to approximate, even when the network is an acyclic digraph. Still on the approximability, we proposed different approximation algorithms for particular cases of κ-PSEC.

Based on dynamic programming techniques, polynomial-time algorithms were presented to efficiently solve special cases of both problems. Those cases considered common network layouts such as paths, cycles, trees, and pseudotrees.

Those algorithms allowed to efficiently solve κ-PSEC when the maximum degree of the graph is at most 2. On the other hand, we proved that κ-PSEC is NP-hard when the maximum degree in the graph is 6, being still unknown if the problem remains NP-hard when the maximum degree of the graph is 3, 4 or 5. For the case of maximum degree 3, we proposed a polynomial-time $\frac{3}{2}$-approximation algorithm, while the other approximation algorithms for κ-PSEC achieve a factor that depends on the parameter k, or apply to graphs where every node has odd degree.

Many questions remain open regarding PCEC, κ-PSEC, and related problems. Would it be possible to obtain approximation ratios or stronger inapproximability results for other particular cases of κ-PSEC? In particular, is κ-PSEC APX-hard? Are there other layouts that admit polynomial-time algorithms? Would the problems remain hard if we consider other classes of graphs to cover the arcs? Future works will attempt to answer some of these questions.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We would like to thank the anonymous reviewers for taking the time and effort necessary to review the manuscript. We sincerely appreciate all valuable comments and suggestions, which helped us to improve the quality of the manuscript.

This study was funded in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - Finance Code 001, and by Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq (Proc. 423833/2018-9 and 310979/2020-0), Brazil.

References

[1] R. Diestel, Graph Theory, 5th edition, Graduate Texts in Mathematics, vol. 173, Springer-Verlag, Heidelberg, 2016/2017.
[2] M. Brandizi, N. Kurbatova, U. Sarkans, P. Rocca-Serra, GRAPH2TAB, a library to convert experimental workflow graphs into tabular formats, Bioinformatics 28 (12) (2012) 1665–1667.
[3] V. Jeyakumar, M. Alizadeh, Y. Geng, C. Kim, D. Mazières, Millions of little minions: using packets for low latency network programming and visibility, Comput. Commun. Rev. 44 (4) (2014) 3–14.
[4] J.A. Marques, M.C. Luizelli, R.I. Tavares da Costa Filho, L.P. Gaspary, An optimization-based approach for efficient network monitoring using in-band network telemetry, J. Internet Serv. Appl. 10 (1) (2019).
[5] Q. Zhang, V. Liu, H. Zeng, A. Krishnamurthy, High-resolution measurement of data center microbursts, in: Proceedings of the 2017 Internet Measurement Conference (IMC), Association for Computing Machinery, New York, NY, USA, 2017, pp. 78–85.
[6] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, 1993.
[7] S.V. Ravelo, Minimum constellation covers: hardness, approximability and polynomial cases, J. Comb. Optim. 41 (3) (2021) 603–624.

[8] T.F. Gonzalez, Clustering to minimize the maximum intercluster distance, Theor. Comput. Sci. 38 (1985) 293–306, article 8.
[9] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, 3rd edition, MIT Press, 2009.
[10] R. Duan, S. Pettie, H.-H. Su, Scaling algorithms for weighted matching in general graphs, ACM Trans. Algorithms 14 (1) (2018) 8.
[11] S. Micali, V.V. Vazirani, An O($\sqrt{|V|} \cdot |E|$) algorithm for finding maximum matching in general graphs, in: 21st Annual Symposium on Foundations of Computer Science (FOCS 1980), 1980, pp. 17–27.