Computing directed Steiner path covers

Frank Gurski¹ · Dominique Komander¹ · Carolin Rehs¹ · Jochen Rethmann² · Egon Wanke¹

Accepted: 8 July 2021 / Published online: 27 July 2021
© The Author(s) 2021

Abstract

In this article we consider the Directed Steiner Path Cover problem on directed co-graphs. Given a directed graph $G = (V, E)$ and a set $T \subseteq V$ of so-called terminal vertices, the problem is to find a minimum number of vertex-disjoint simple directed paths, which contain all terminal vertices and a minimum number of non-terminal vertices (Steiner vertices). The primary minimization criteria is the number of paths. We show how to compute in linear time a minimum Steiner path cover for directed co-graphs. This leads to a linear time computation of an optimal directed Steiner path on directed co-graphs, if it exists. Since the Steiner path problem generalizes the Hamiltonian path problem, our results imply the first linear time algorithm for the directed Hamiltonian path problem on directed co-graphs. We also give binary integer programs for the (directed) Hamiltonian path problem, for the (directed) Steiner path problem, and for the (directed) Steiner path cover problem. These integer programs can be used to minimize change-over times in pick-and-place machines used by companies in electronic industry.

A short part of this article has already been published Gurski et al. (2020) in the Proceedings of SOFSEM 2020.

✉ Jochen Rethmann
jochen.rethmann@hsnr.de

Frank Gurski
frank.gurski@hhu.de

Dominique Komander
dominique.komander@hhu.de

Carolin Rehs
carolin.rehs@hhu.de

Egon Wanke
egon.wanke@hhu.de

¹ Institute of Computer Science, Heinrich Heine University Düsseldorf, 40225 Düsseldorf, Germany

² Faculty of Electrical Engineering and Computer Science, Niederrhein University of Applied Sciences, 47805 Krefeld, Germany

Keywords Binary integer program · Combinatorial optimization · Directed co-graphs · Directed Steiner path cover problem · Directed Steiner path problem · Directed Hamiltonian path problem · Pick-and-place machines

1 Introduction

For the well known Steiner tree problem there are efficient algorithms on special graph classes like series-parallel graphs (Wald and Colbourn 1983), outerplanar graphs (Wald and Colbourn 1982) and graphs of bounded tree-width (Bodlaender et al. 2015; Chimani et al. 2012). Some authors consider the class Steiner tree problem (CSP), which is a generalization of the Steiner tree problem where the vertices are partitioned into classes of terminals (Reich and Widmayer 1990). The unit-weight version of CSP is linear time solvable on co-graphs (Westbrook and Yan 1995).

The Steiner path problem is a special restriction of the Steiner tree problem in which the required terminal vertices lie on one path with minimal cost. It can also be seen as a generalization of the Hamiltonian path problem, since if we choose each vertex as a terminal vertex and if we have uniform edge weights, the Steiner path problem is equal to the Hamiltonian path problem. In Abu-Affash et al. (2014) they observed the Euclidean bottleneck Steiner path problem. Moreover, in Moharana et al. (2013) there was given a linear time solution for the Steiner path problem on trees.

A Steiner tree always exists within connected graphs. Nevertheless, it is not always possible to find a Steiner path, which motivates us to look at Steiner path cover problems. This Steiner path cover problem was already considered in Cusic and Lendl (2021) on interval graphs.

In this article we investigate the directed Steiner path cover problem for which we give the following definition. Let G be a directed graph with vertex set $V(G)$ and edge set $E(G)$ and let $T \subseteq V(G)$ be a set of terminal vertices. Further, let $c : E(G) \rightarrow \mathbb{R}^{\geq 0}$ be a function that assigns a weight to each edge. A *directed Steiner path cover* for G is a set of vertex-disjoint simple directed paths P in G that consists of all terminal vertices T and possibly some additional non-terminal (Steiner) vertices of $V(G) - T$. We define the *size* of a directed Steiner path cover as the number of its paths, i.e. the size is $|P|$, while the *cost* is defined as the sum of weights of the edges used in the paths in a directed Steiner path cover of minimum size.

Name: Directed Steiner Path Cover

Instance: A directed graph G , a set of terminal vertices $T \subseteq V(G)$, and edge weights $c : E(G) \rightarrow \mathbb{R}^{\geq 0}$.

Task: Find a directed Steiner path cover P of minimum size for G that minimizes $\sum_{p \in P} \sum_{e \in p} c(e)$.

Minimizing only the sum of the weights of the edges is not reasonable since this sum is minimized if we take each terminal vertex on its own as a path of length 0. So we primarily require that the number of paths is minimal.

The directed Steiner path cover problem is NP-hard since it is a generalization of the directed Hamiltonian path problem. This gives a motivation for a restriction of the problem to special graph classes. We observe digraphs of a very natural graph

class, namely directed co-graphs, which are defined as follows. Directed co-graphs (complement reducible graphs) are generated starting with the single vertex graph by applying disjoint union, order composition and series composition (Bechet et al. 1997). There exists another characterization of directed co-graphs by the exclusion of eight forbidden induced sub-digraphs, see Crespelle and Paul (2006, Fig. 2).

Further, directed co-graphs are precisely the digraphs of directed NLC-width 1 and they are a proper subset of the digraphs of directed clique-width at most 2 (Bang-Jensen and Gutin 2018; Gurski et al. 2016). This graph class is also interesting from an algorithmic point of view. Several hard graph problems are polynomial time solvable using dynamic programming along the tree structure of the input graph, see Bang-Jensen and Maddaloni (2014), Gurski (2017), Gurski et al. (2019), Gurski et al. (2019), Gurski and Rehs (2018), Gurski et al. (2021). Additionally, there is a practical application as directed co-graphs can be used for the reconstruction of the evolutionary history of genes or species using genomic sequence data, see Hellmuth et al. (2017), Nojgaard et al. (2018).

For graphs where all edges have the same weight, the above definition of the directed Steiner path cover problem results in a minimum number of Steiner vertices. Graphs without edge weights can be considered as a special case of graphs with unit-edge weights. Since edge weights do not occur in co-graphs, we use the following problem definition.

Name: Unit-Edge-Weight Directed Steiner Path Cover

Instance: A directed graph G and a set of terminal vertices $T \subseteq V(G)$.

Task: Find a directed Steiner path cover of minimum size for G such that the number of Steiner vertices is minimal.

In the following we present how to compute the value of a directed Steiner path cover of minimum size and cost for the disjoint union, order composition and series composition of two directed co-graphs in linear time from the corresponding values of the involved directed co-graphs. For this purpose, we give a definition of a useful normal form for directed Steiner path covers in digraphs which are defined by the order composition or series composition of two directed co-graphs. Moreover, we show a linear time algorithm which builds a directed Steiner path cover of minimum size and cost for a directed co-graph.

The (unit-weight) directed Steiner tree problem is MSO_1 -definable by Ganian et al. (2014, Proposition 4.14). But this does not hold for the directed Steiner path (cover) problem, since it is a generalization of the directed Hamiltonian path problem which is not MSO_1 -definable. Thus it is not possible to obtain our solutions for the directed Steiner path (cover) problem using the fact that directed co-graphs have bounded directed clique-width (Gurski et al. 2016) and the result from Ganian et al. (2014) stating that all MSO_1 -definable digraph problems are fixed parameter tractable for the parameter directed clique-width. Since the Hamiltonian cycle problem is $\text{W}[1]$ -hard parameterized by clique-width (Fomin et al. 2010) it even follows that Hamiltonian path problem and thus the (directed) Steiner path (cover) problem are $\text{W}[1]$ -hard when parameterized by (directed) clique-width.

We also give binary integer programs for the directed Hamiltonian path problem and for the directed Steiner path (cover) problem. These integer programs can be used

to minimize change-over times in pick-and-place machines used in electronic industry. The problem of minimizing change-over times is introduced in Sect. 4.

Name: Directed Steiner Path

Instance: A directed graph G , a set $T \subset V(G)$ of terminal vertices, and a function $c : E(G) \rightarrow \mathbb{R}^{\geq 0}$ that assigns to each edge some weight.

Task: Find a directed Steiner path p in graph G that minimizes $\sum_{e \in p} c(e)$.

2 Preliminaries

We consider the directed Steiner path cover problem on directed co-graphs, so we first recall the definition of directed co-graphs and we give a normal form of directed Steiner path covers.

2.1 Directed co-graphs

The class of directed co-graphs was introduced by Bechet et al. in (1997).

Definition 1 The class of *directed co-graphs* is recursively defined as follows.

- (i) Every digraph on a single vertex $(\{v\}, \emptyset)$, denoted by $\bullet_v$, is a *directed co-graph*.
- (ii) If A, B are vertex-disjoint directed co-graphs, then
  - (a) the disjoint union $A \oplus B$, which is defined as the digraph with vertex set $V(A) \cup V(B)$ and edge set $E(A) \cup E(B)$,
  - (b) the order composition $A \oslash B$, defined by their disjoint union plus all possible edges only directed from $V(A)$ to $V(B)$, and
  - (c) the series composition $A \otimes B$, defined by their disjoint union plus all possible edges between $V(A)$ and $V(B)$ in both directions,
 are *directed co-graphs*.

An expression which uses these operations is called a *directed co-expression*. The recursive generation of a directed co-graph can be described by a tree structure, called *directed co-tree*. The vertices of the digraph are represented by the leaves of the directed co-tree, while the inner vertices of the directed co-tree correspond to the operations applied on the subgraphs of G defined by the respective subtrees. Constructing a directed co-tree for a directed co-graph is possible in linear time, see Crespelle and Paul (2006).

For the characterization of directed co-graphs by excluding eight forbidden induced sub-digraphs, see Crespelle and Paul (2006), Fig. 2 or Table 1.

In the following we give a definition of a normal form for directed Steiner path covers in directed co-graphs which are defined by the order composition or series composition of two directed co-graphs.

Table 1 The eight forbidden induced sub-digraphs for directed co-graphs (see Crespelle and Paul 2006)

2.2 Normal form for directed steiner path covers

Let G be a directed co-graph and let $T \subseteq V(G)$ be a set of terminal vertices. Further, let C be a directed Steiner path cover for G with respect to T , while $s(C)$ denotes the number of Steiner vertices in the paths of C .

Lemma 1 *Let C be a directed Steiner path cover for some directed co-graph $G = A \odot B$ or $G = A \otimes B$ with respect to a set $T \subseteq V(G)$ of terminal vertices. Then, there is a directed Steiner path cover C' with respect to T that does not contain paths p and p' satisfying one of the structures (1)-(4), such that $|C| \geq |C'|$ and $s(C) \geq s(C')$ applies. Let $q_1, \dots, q_4$ denote sub-paths which may be empty.*

1. $p = (x, q_1)$ or $p = (q_1, x)$ where $x \notin T$. Comment: No path starts or ends with a Steiner vertex.
2. $p = (q_1, u, x, v, q_2)$ where $u \in V(A)$, $v \in V(B)$, and $x \notin T$. Comment: On a path, the neighbors u, v of a Steiner vertex x are both contained in the same digraph.
3. $p = (q_1, x)$, $p' = (u, q_2)$, where $x \in V(A)$, $u \in V(B)$, $p \neq p'$. Comment: No path p ends in A , if there is a path $p' \neq p$ that starts in B .
4. $p = (\dots, x, u, v, y, \dots)$ where $u, v \notin T$. Comment: The paths contain no edge between two Steiner vertices.

If $G = A \otimes B$, then the cover C' also does not contain paths satisfying structures (5)-(8).

5. $p = (x, q_1)$, $p' = (u, q_2)$, where $x \in V(A)$, $u \in V(B)$, $p \neq p'$. Comment: All paths start in the same digraph.
6. $p = (q_1, x, y, q_2)$, $p' = (q_3, u, v, q_4)$ where $x, y \in V(A)$, $u, v \in V(B)$. Comment: The cover C' contains edges of only one of the digraphs.
7. $p = (x, q_1)$, $p' = (q_2, u, y, v, q_3)$, where $x, y \in V(A)$, $u, v \in V(B)$, and $y \notin T$. Comment: If a path starts in A then there is no Steiner vertex in A with two neighbors on the path in B .
8. $p = (x, q_1)$, $p' = (q_2, u, v, q_3)$, where $x \in V(A)$ and $u, v \in V(B)$. Comment: If a path starts in A , then no edge of B is contained in the cover.

- Proof**
1. If x is removed from p we get a cover with one Steiner vertex less than C .
  2. If x is removed from p , we get a cover with one Steiner vertex less than C .
  3. We combine the paths to only one path (q_1, x, u, q_2) and we get a cover with one path less than C .
  4. Since G is a directed co-graph, the underlying undirected graph is a co-graph such that the path cannot include a P_4 , i.e. a simple path of 4 vertices, as induced subgraph. Thus, there must be at least one additional arc. If such an additional arc would shorten the Steiner path by skipping u , v or both then we remove u or v or both and take the shortcut for getting a cover C' . Additional arcs that do not shorten the path would create a forbidden induced subgraph from Table 1 which is not possible. For details see Table 2.
  5. The new paths are q_1 and (x, u, q_2) . The cover C' is as good as C .
  6. If $p \neq p'$, then (q_1, x, v, q_4) and (q_3, u, y, q_2) are the paths in cover C' , see Fig. 1. If $p = p'$, then we have to distinguish whether $(u, v) \in q_1$, $(u, v) \in q_2$, $(x, y) \in q_3$, or $(x, y) \in q_4$. We show how to handle the first case, the other three cases are similar. Let $p = (q_3, u, v, q_5, b, a, q_6, x, y, q_2)$, where $b \in V(B)$ and $a \in V(A)$. Then the new path in cover C' is $(q_3, u, a, q_6, x, v, q_5, b, y, q_2)$. Such vertices a and b must exist because $v \in V(B)$ and $x \in V(A)$, possibly it holds $a = x$ or $b = v$. In any case cover C' is as good as C , see Fig. 2.
  7. If $p \neq p'$, then q_1 and (q_2, u, x, v, q_3) are the new paths in cover C' . If $p = p'$, i.e. $q_1 = (q'_2, u, y, v, q_3)$, where q'_2 is obtained from q_2 by removing x , then (q'_2, u, x, v, q_3) is the new path in cover C' . The cover C' is as good as C , see Fig. 3. If $p \neq p'$, then the edge (a, b) is missing in Fig. 3.
  8. If $p \neq p'$, then q_1 and (q_2, u, x, v, q_3) are the new paths in cover C' . If $p = p'$, i.e. $q_1 = (q'_2, u, v, q_3)$, where q'_2 is obtained from q_2 by removing x , then (q'_2, u, x, v, q_3) is the new path in cover C' . The cover C' is as good as C , see Fig. 4. If $p \neq p'$, then the edge (a, b) is missing in Fig. 4.

Operations 1, 2, 4 and 7 reduce the number of Steiner vertices by one, the remaining operations 3, 5 and 6 do not change the number of Steiner vertices. Therefore, operations 1, 2, 4 and 7 can only be executed at most $|V - (T_A \cup T_B)|$ times.

Operation 6 reduces the number of paths by one, the remaining operations do not increase the number of paths. Therefore operation 6 can be executed at most $\max\{|T_A|, |T_B|\}$ times.

Let us now consider those edges on a path that connect vertices of A and vertices of B . The maximum number of those edges is $|V(A)| + |V(B)| - 1$. Operation 7 can remove two such edges, operations 3 and 5 can add two such edges. Since the other operations 1, 2, 4 and 6 do not reduce the number of edges, operations 3 and 5 can be used at most $(|V(A)| + |V(B)| - 1)/2 + |V - (T_A \cup T_B)|$ times. $\square$

The hypothesis of Lemma 1 is symmetric in A and B and thus, the statement of Lemma 1 is also valid for co-graphs $G = A \otimes B$ if we switch A and B .

Definition 2 A directed Steiner path cover C for some directed co-graph $G = A \oslash B$ or $G = A \otimes B$ is said to be in *normal form* if none of the operations described in the proof of Lemma 1 is applicable.

Now we assume that a directed Steiner path cover for some directed co-graph $G = A \oslash B$ or $G = A \otimes B$ is always in normal form, since the operations of the

Table 2 The leftmost column shows a graph with underlying undirected P_4 and at least one additional arc that do not shorten the path. The other columns shows the forbidden subgraphs that are contained in the leftmost graph depending on the edges of the P_4

Graph with underlying P_4 and additional edges that do not shorten the path	None	a	b	c
	a, b	a, c	b, c	a, b, c
	D_5 {1, 2, 3} D_3 {2, 3, 4}	D_4 {1, 2, 3} D_4 {1, 2, 3}	D_4 {1, 2, 3} D_4 {1, 2, 3}	D_3 {1, 3, 4} D_3 {1, 3, 4}
	D_5 {2, 3, 4} D_2 {1, 2, 4}	D_2 {1, 2, 4} D_4 {2, 3, 4}	D_2 {1, 2, 3} D_2 {1, 2, 3}	D_4 {2, 3, 4} D_2 {1, 2, 4}
	D_5 {2, 3, 4} D_4 {2, 3, 4}	D_4 {1, 2, 3} D_4 {2, 3, 4}	D_4 {1, 2, 3} D_4 {1, 2, 3}	D_4 {2, 3, 4} D_2 {1, 2, 4}
	D_5 {1, 2, 3} D_4 {2, 3, 4}	D_4 {1, 2, 3} D_4 {2, 3, 4}	D_4 {1, 2, 3} D_4 {1, 2, 3}	D_4 {2, 3, 4} D_6 {1, 2, 3, 4}
	D_5 {2, 3, 4} D_1 {1, 3, 4}	D_3 {1, 2, 3} D_3 {1, 3, 4}	D_2 {1, 2, 3} D_3 {1, 3, 4}	D_1 {1, 2, 3} D_3 {1, 3, 4}
	D_5 {1, 2, 3} D_2 {1, 2, 4}	D_1 {2, 3, 4} D_2 {1, 2, 4}	D_1 {1, 2, 4} D_1 {1, 2, 4}	D_1 {1, 2, 4} D_2 {1, 2, 4}
	D_1 {1, 2, 3} D_3 {2, 3, 4}	D_1 {2, 3, 4} D_2 {1, 2, 4}	D_1 {1, 2, 4} D_2 {1, 2, 3}	D_1 {1, 2, 3} D_3 {1, 3, 4}

Fig. 1 Substitution in the proof of Lemma 1(6.) for $p \neq p'$

Fig. 2 Substitution in the proof of Lemma 1(6.) for $p = p'$

Fig. 3 Substitution in the proof of Lemma 1(7.)

Fig. 4 Substitution in the proof of Lemma 1(8.)

proof of Lemma 1 do not increase the number of paths or Steiner vertices of a cover. Lemma 1 implies the following theorem.

Theorem 1 For each directed co-graph $G = A \otimes B$ and set of terminal vertices $T \subseteq V(G)$ any directed Steiner path cover C in normal form with respect to T does not contain an edge of digraph A , and no path in C starts or ends in digraph A if $|T_A| \leq |T_B|$.

Proof (by contradiction) Assume, the Steiner path cover C contains an edge of digraph A . Then by Lemma 1(5), all paths starts in digraph A . By Lemma 1(4), it holds that no Steiner vertex v of $V(A)$ is contained in C , where the neighbors of v are both of digraph B . By Lemma 1 (1), (2), and (5), it holds that all vertices of $V(B)$ from C are connected with a terminal vertex of $V(A)$, thus $|T_A| > |T_B|$. ζ

Second, we have to show that no path in C starts or ends in digraph A . Assume on the contrary, that there is one path that starts in A . By Lemma 1(6), it holds that all paths start in A . Continuing as in the first case this leads to a contradiction. $\square$

Remark 1 For each directed co-graph $G = A \otimes B$ and set of terminal vertices $T \subseteq V(G)$ any directed Steiner path cover C in normal form with respect to T it holds that each path that starts in A either remains in A or it crosses over to B and remains in B . Each path that reaches a vertex of B has to stay in B since no edge from a vertex in B to a vertex in A exists.

3 Algorithms for the directed Steiner path cover problem

3.1 Computing the optimal number of paths

Let G be a directed co-graph and $T \subseteq V(G)$ be a set of terminal vertices. We define $p(G, T)$ as the minimum number of paths within a Steiner path cover for G with respect to T . Further, let $s(G, T)$ be the minimum number of Steiner vertices in a directed Steiner path cover of size $p(G, T)$ with respect to T . If it is clear from the context, we do not specify set T .

Lemma 2 *Let A and B be two vertex-disjoint directed co-graphs and let $T_A \subseteq V(A)$ and $T_B \subseteq V(B)$ be two sets of terminal vertices. The following equations apply.*

1. $p(\bullet_v, \emptyset) = 0$ and $p(\bullet_v, \{v\}) = 1$
2. $p(A \oplus B, T_A \cup T_B) = p(A, T_A) + p(B, T_B)$
3. $p(A \otimes B, \emptyset) = 0$
4. $p(A \otimes B, T_A \cup T_B) = \max\{1, p(B, T_B) - |V(A)|\}$ if $1 \leq |T_B|$ and $|T_A| \leq |T_B|$
5. $p(A \otimes B, T_A \cup T_B) = \max\{1, p(A, T_A) - |V(B)|\}$ if $1 \leq |T_A|$ and $|T_A| > |T_B|$
6. $p(A \oslash B, T_A \cup T_B) = p(A, T_A)$ if $p(A) > p(B)$
7. $p(A \oslash B, T_A \cup T_B) = p(B, T_B)$ if $p(A) \leq p(B)$

Proof 1. - 3. Obviously holds.

4. We show $p(A \otimes B) \geq \max\{1, p(B) - |V(A)|\}$ by an indirect proof. We assume that in a directed Steiner path cover C for $A \otimes B$ there are less than $\max\{1, p(B) - |V(A)|\}$ paths. Removing all vertices of A from all paths in C leads to a directed Steiner path cover of size $|C| + |V(A)| < p(B)$ for B . $\nexists$

We now show that $p(A \otimes B) \leq \max\{1, p(B) - |V(A)|\}$ applies. We can use any vertex of A to combine two paths of the cover of B to one path, as the series composition of A and B creates every possible directed edge between A and B . If there exists more terminal vertices in T_A than there are paths in the cover of B , i.e. $p(B) < |T_A|$, we split paths of B and reconnect them with terminal vertices of T_A , which is always possible since $|T_A| \leq |T_B|$.

5. Similar to 4.
6. We show that $p(A \oslash B) \leq p(A)$ holds. Consider that it is possible to append any path of A by any path of B , see Lemma 1(3). As this creates no edge between B and A , we cannot extend a path of B with a path of A .

We show by an indirect proof that $p(A \oslash B) \geq p(A)$ applies. We assume that a directed Steiner path cover C for $A \oslash B$ contains less than $p(A)$ paths. By removing all vertices of B from all paths in C we get a Steiner path cover of size $|C| < p(A)$. $\nexists$

6. Similar to 6. □

3.2 Computing the optimal number of Steiner vertices

Remark 2 For two vertex-disjoint directed co-graphs A, B and two sets of terminal vertices $T_A \subseteq V(A)$, $T_B \subseteq V(B)$ it holds that $s(A \oplus B, T_A \cup T_B) = s(A, T_A) + s(B, T_B)$, as the disjoint union does not create new edges.

Remark 3 Let $G = A \oslash B$ be a directed co-graph, and let C be a directed Steiner path cover of G such that $p = (q_1, u_1, x, q_2, v_1)$ is a path in A , $p_1 = (u_2, q_3)$ and $p_2 = (v_2, q_4)$ are paths in B and all paths are vertex-disjoint paths in C , where $x \notin T$, $u_1, u_2, v_1, v_2 \in T$, and $q_1, \dots, q_4$ are sub-paths. Now, we split p at vertex x into two paths and combine them with p_1 and p_2 to obtain (q_1, u_1, u_2, q_3) and (q_2, v_1, v_2, q_4) as new paths. On this way we get a Steiner path cover without increasing the cost. If we switch A and B , we obtain (u_2, q_3, q_1, u_1) and (v_2, q_4, q_2, v_1) as new paths such that the statement holds as well.

We conclude the central lemma of our work. The proof is done by induction on the structure of the co-graph.

Lemma 3 Let G be a directed co-graph and C a directed Steiner path cover for G with respect to a set $T \subseteq V(G)$ of terminal vertices. Then, it holds that $p(G) + s(G) \leq |C| + \mathfrak{s}(C)$.

Proof (by induction) Obviously, the statement is valid for directed co-graphs with only one vertex. We now assume that the statement is valid for directed co-graphs of n vertices. Let A and B be vertex-disjoint directed co-graphs with at most n vertices each.

Disjoint union: Let $G = A \oplus B$ be a directed co-graph with more than n vertices. By Lemma 2, and Remark 2, we know that $p(A \oplus B) + s(A \oplus B) = p(A) + p(B) + s(A) + s(B)$. By the induction hypothesis, $p(A) + s(A) \leq |C_{|A}| + \mathfrak{s}(C_{|A})$ and $p(B) + s(B) \leq |C_{|B}| + \mathfrak{s}(C_{|B})$ apply, where $C_{|A}$ denotes the cover C restricted to digraph A , i.e. the cover which results if we remove all vertices of B from C . This yields to the statement of the lemma.

$$p(A \oplus B) + s(A \oplus B) \leq |C_{|A}| + \mathfrak{s}(C_{|A}) + |C_{|B}| + \mathfrak{s}(C_{|B}) = |C| + \mathfrak{s}(C)$$

Series composition: Let $G = A \otimes B$ be a directed co-graph on more than n vertices and without loss of generality it holds that $|T_A| \leq |T_B|$.

1. Let $X(A)$ denote the vertices of A used in cover C , and let D be the cover for B that results by removing the vertices of $X(A)$ from the cover C . By induction hypothesis, we get $p(B) + s(B) \leq |D| + \mathfrak{s}(D)$.
2. Let $nt(X(A))$ be the number of non-terminal vertices of $X(A)$. By Theorem 1 we get that $\mathfrak{s}(C) = \mathfrak{s}(D) + nt(X(A))$ and $|C| = |D| - |T_A| - nt(X(A))$. Consequently, we come to $|C| + \mathfrak{s}(C) = |D| + \mathfrak{s}(D) - |T_A|$.

Combining these results together we get:

$$p(B) + s(B) - |T_A| \leq |D| + \mathfrak{s}(D) - |T_A| = |C| + \mathfrak{s}(C)$$

To prove the statement of the lemma, we start with considering the case $p(B) - 1 \leq |V(A)|$. Then, we get $p(A \otimes B) = 1$. If $|T_A| \geq p(B) - 1$, then $d := |T_A| - (p(B) - 1)$ many Steiner vertices from B , are replaced by terminal vertices from A , if they are available. Otherwise, if $|T_A| < p(B) - 1$, we get $-d = (p(B) - 1) - |T_A|$ many Steiner vertices from A with which we combine the paths. Consequently, it holds that

$s(A \otimes B) \leq \max\{0, s(B) - d\}$ as the number of Steiner vertices in an optimal cover is at most the number of Steiner vertices in a certain cover. Thus, since $p(A \otimes B) = 1$ we obtain for $s(B) \geq d$:

$$\begin{aligned} p(A \otimes B) + s(A \otimes B) &\leq 1 + s(B) - d = 1 + s(B) - (|T_A| - (p(B) - 1)) \\ &= \chi + s(B) - |T_A| + p(B) - \chi \leq |C| + \mathfrak{s}(C) \end{aligned}$$

If $s(B) < d$, then all Steiner vertices of B are replaced by terminal vertices of A and as we have $|T_A| \leq |T_B|$, some of the paths of B are reconnected by the remaining terminal vertices of A . Consequently, we get $p(A \otimes B) + s(A \otimes B) = 1 \leq |C| + \mathfrak{s}(C)$.

Now, we come to the case where $p(B) - 1 > |V(A)|$, i.e. it is not possible to combine all paths in an optimal cover for B with vertices of A . By Lemma 2, we know that $p(A \otimes B) = \max\{1, p(B) - |V(A)|\}$. Consequently, for $p(A \otimes B) > 1$ it follows:

$$\begin{aligned} p(A \otimes B) + s(A \otimes B) &\leq p(B) - |V(A)| + s(B) + nt(A) \\ &= p(B) + s(B) - |T_A| \leq |C| + \mathfrak{s}(C) \end{aligned}$$

Since the non-terminal vertices of A are now used to combine paths of the cover, the non-terminal vertices of A become Steiner vertices.

Order composition: Let $G = A \oslash B$ be a directed co-graph with more than n vertices. By the induction hypothesis, we get $p(A) + s(A) \leq |C_A| + \mathfrak{s}(C_A)$ and $p(B) + s(B) \leq |C_B| + \mathfrak{s}(C_B)$.

Let us first consider the case $p(A) > p(B)$. By Lemma 2 it holds $p(A \oslash B) = p(A)$. We can append any path of A by any path of B , and by Remark 3 it holds that for every path that there is more in A than in B , a Steiner vertex of B can be removed. Additionally, as an optimal cover has at most as many Steiner vertices as a concrete cover, we get $s(A \oslash B) \leq \mathfrak{s}(C_A) + \mathfrak{s}(C_B) - \min\{\mathfrak{s}(C_B), |C_A| - |C_B|\}$. If we sum up both equations we come to

$$p(A \oslash B) + s(A \oslash B) \leq p(A) + \mathfrak{s}(C_A) + \mathfrak{s}(C_B) - \min\{\mathfrak{s}(C_B), |C_A| - |C_B|\}$$

If $\mathfrak{s}(C_B) \geq |C_A| - |C_B|$ applies, and as $\mathfrak{s}(C) = \mathfrak{s}(C_A) + \mathfrak{s}(C_B)$ applies, we get

$$p(A \oslash B) + s(A \oslash B) \leq p(A) + \mathfrak{s}(C) - |C_A| + |C_B|.$$

The statement would be shown if $p(A) - |C_A| + |C_B| \leq |C|$ applied. It holds $p(A) \leq |C_A|$ since an optimal cover has at most as many paths as a concrete cover, and it holds $|C_B| \leq |C|$, since $|C| = \max\{|C_A|, |C_B|\}$ by Remark 1. We sum up these equations which leads to $p(A) + |C_B| \leq |C_A| + |C|$. This is equivalent to $p(A) - |C_A| + |C_B| \leq |C|$, such that $p(A \oslash B) + s(A \oslash B) \leq |C| + \mathfrak{s}(C)$ has been shown.

If $\mathfrak{s}(C_B) < |C_A| - |C_B|$, then we get $p(A \oslash B) + s(A \oslash B) \leq p(A) + \mathfrak{s}(C_A)$. It is left to show that $p(A) + \mathfrak{s}(C_A) \leq |C| + \mathfrak{s}(C)$ applies. Since an optimal cover has at most as many paths as a concrete cover, it holds $p(A) \leq |C_A|$. Further, we

Fig. 5 Small example that shows the contrapositive of the statement of Lemma 3 in a graph B that is not directed co-graph

know that $|C_A| \leq |C|$ since $|C| = \max\{|C_A|, |C_B|\}$ by Remark 1. Moreover, it holds $s(C_A) \leq s(C)$, as a part can at most be as big as the whole.

We can show the other case $p(A) \leq p(B)$ on a similar way. $\square$

To see why Lemma 3 is crucial for the rest of this work, consider the directed graph B of Fig. 5 that is not a directed co-graph. Terminal vertices $T_A = \{f, g\}$ and $T_B = \{a, c, e, u, w, y\}$ are shown as squares. In the left part of the figure a Steiner path cover $C_\ell = \{(a, b, c, d, e), (u, v, w, x, y)\}$ for graph B is shown with $|C_\ell| = 2$ and $s(C_\ell) = 4$ which is optimal. In the right part of the figure a Steiner path cover $C_r = \{(a, b, c, w, x, y), (e), (u)\}$ for B is shown with $|C_r| = 3$ and $s(C_r) = 2$.

The right cover can be extended to an optimal cover for $A \otimes B$ if the vertices of A are used to combine the path: $\{(u, f, a, b, c, w, x, y, g, e)\}$ is an optimal cover for $A \otimes B$ with only one path and 2 Steiner vertices. The left Steiner path cover can not be extended to an optimal cover for $A \otimes B$. For example, we can split path (a, b, c, d, e) at vertex b into two paths (a) and (c, d, e) and reconnect them by a vertex of A and get (a, f, c, d, e) . The other vertex of A must be used to combine the remaining two paths to $(a, f, c, d, e, g, u, v, w, x, y)$ which results in a cover for $A \otimes B$ that consists of one path but 3 Steiner vertices. For graph B the statement of Lemma 3 is not satisfied: $p(B) + s(B) = 2 + 4 = 6 > |C_r| + s(C_r) = 3 + 2 = 5$

In the proof of Lemma 4 we use the statement of Lemma 3 to show that optimal solutions for directed co-graphs A and B can be combined to an optimal solution for $A \odot B$ and $A \otimes B$.

Remark 4 Let G be a directed co-graph and let C be a directed Steiner path cover for G with respect to some set of terminal vertices $T \subseteq V(G)$. Then $s(C) \geq s(G)$ holds only if $|C| = p(G)$. If $|C| > p(G)$ then $s(C)$ might be smaller than $s(G)$.

Lemma 4 Let A and B be two vertex-disjoint digraphs, and let $T_A \subseteq V(A)$, $T_B \subseteq V(A)$ be sets of terminal vertices. Then the following equations applies:

1. $s(\bullet_v, \emptyset) = 0$ and $s(\bullet_v, \{v\}) = 0$
2. $s(A \oplus B, T_A \cup T_B) = s(A, T_A) + s(B, T_B)$
3. $s(A \otimes B) = \max\{0, s(B) + p(B) - p(A \otimes B) - |T_A|\}$ if $|T_A| \leq |T_B|$
4. $s(A \otimes B) = \max\{0, s(A) + p(A) - p(A \otimes B) - |T_B|\}$ if $|T_A| > |T_B|$
5. $s(A \odot B) = s(A) + s(B)$ if $p(A) = p(B)$
6. $s(A \odot B) = s(A) + s(B) - \min\{s(A), p(B) - p(A)\}$ if $p(A) < p(B)$
7. $s(A \odot B) = s(A) + s(B) - \min\{s(B), p(A) - p(B)\}$ if $p(A) > p(B)$

Proof 1. Obvious.

2. See Remark 2

3. At first we show $s(A \otimes B) \leq \max\{0, s(B) + p(B) - p(A \otimes B) - |T_A|\}$.

Lemma 3 implies that $s(A \otimes B) + p(A \otimes B) \leq s(C) + |C|$ holds for any cover C for directed co-graph $A \otimes B$ and any set of terminal vertices T . We consider the cover C for $A \otimes B$ which is obtained by an optimal cover D for B in the following way: Take the terminal vertices of A to either combine paths of D or to remove a Steiner vertex of D by replacing $v \notin T$ by a terminal vertex of A in a path as $(\dots, u, v, w, \dots) \in D$, where $u, w \in T$. If $|T_A| \geq s(B) + p(B)$, then we can combine all paths of D while all Steiner vertices can be removed by terminal vertices of A . Since $|T_A| \leq |T_B|$ holds, we can split some of the paths and reconnected them by the remaining terminal vertices of A . Consequently, we get $s(C) + |C| = 1$ and $s(A \otimes B) = 0$.

Otherwise, if $|T_A| < s(B) + p(B)$, then we have $s(C) + |C| = s(B) + p(B) - |T_A|$, and by Lemma 3, the statement follows.

$$\begin{aligned} s(A \otimes B) + p(A \otimes B) &\leq s(B) + p(B) - |T_A| = s(C) + |C| \\ \iff s(A \otimes B) &\leq s(B) + p(B) - p(A \otimes B) - |T_A| \end{aligned}$$

We prove now that $s(A \otimes B) \geq \max\{0, s(B) + p(B) - p(A \otimes B) - |T_A|\}$.

By $X(A)$ we denote the vertices of $V(A)$ that are part of the paths in an optimal cover C for $A \otimes B$. Let D be the cover for B which we get by removing the vertices of $X(A)$ from C . Then, by Theorem 1 we come to the following:

$$\begin{aligned} |X(A)| &= nt(X(A)) + |T_A| = |D| - p(A \otimes B) \\ \iff nt(X(A)) &= |D| - p(A \otimes B) - |T_A| \end{aligned}$$

Thus, we get:

$$\begin{aligned} s(A \otimes B) - nt(X(A)) &= s(D) = s(A \otimes B) - |D| + p(A \otimes B) + |T_A| \\ \iff s(A \otimes B) &= s(D) + |D| - p(A \otimes B) - |T_A| \\ \Rightarrow s(A \otimes B) &\geq s(B) + p(B) - p(A \otimes B) - |T_A| \end{aligned}$$

The implication follows, as by Lemma 3 it holds $s(D) + |D| \geq s(B) + p(B)$.

4. Can be shown similar to the previous item.

5. To show that $s(A \oslash B) \leq s(A) + s(B)$ holds, we consider optimal covers C and D for A and B . We then build a cover E for $A \oslash B$ in such a way that any path of C is appended by a path of D , see Lemma 1(3). Since $|E| = p(A \oslash B)$ applies, $s(A \oslash B) \leq s(E) = s(C) + s(D) = s(A) + s(B)$ follows. This is because an optimal cover has at most as many Steiner vertices as a concrete cover.

To show that $s(A \oslash B) \geq s(A) + s(B)$ holds, we consider an optimal cover C for $A \oslash B$. Then, we have $s(A \oslash B) = s(C|_A) + s(C|_B) \geq s(A) + s(B)$, as $|C|_A| = p(A) = p(A \oslash B) = p(B) = |C|_B|$, see Remark 4.

6. We distinguish two cases. At first, let $s(A) > p(B) - p(A)$.

To show that $s(A \oslash B) \leq s(A) + s(B) - (p(B) - p(A))$ holds, we consider optimal covers C and D for A and B . We construct a cover E for $A \oslash B$ as follows. We first split $p(B) - p(A)$ many paths of C at Steiner vertices as stated

in Remark 3. After, we combine each of the resulting paths with a path of D . On this way, it holds that $|E| = p(A \otimes B) = p(B)$ and therefore, $s(A \otimes B) \leq s(C) + s(D) - (p(B) - p(A)) = s(A) + s(B) - (p(B) - p(A))$.

Please note, a Steiner path cover C for $A \otimes B$ with $s(C|_A) > 0$ is not optimal if $|C|_A| < |C| = p(A \otimes B)$ holds. By Remark 3 a path of $C|_A$ could be splitted at a Steiner vertex and the number of Steiner vertices could be reduced.

To show $s(A \otimes B) \geq s(A) + s(B) - (p(B) - p(A))$, we look at an optimal cover C for $A \otimes B$. Thus, we have $s(A \otimes B) = s(C) = s(C|_A) + s(C|_B)$. Further, by the previous note it holds $|C| = p(A \otimes B) = p(B) = |C|_A|$. By Lemma 3 we come to $s(C|_A) + |C|_A| \geq s(A) + p(A)$. Summing up these equations, we get $s(A \otimes B) + p(A \otimes B) = s(C|_A) + |C|_A| + s(C|_B)$. Finally, we come to:

$$\begin{aligned} s(A \otimes B) &= s(C|_A) + |C|_A| - p(A \otimes B) + s(C|_B) \\ &\geq s(A) + p(A) - p(B) + s(C|_B) \geq s(A) + p(A) - p(B) + s(B) \end{aligned}$$

The last step holds since $p(B) = |C|_B|$ and by Remark 4.

Now we consider the case in which $s(A) \leq p(B) - p(A)$. To show that $s(A \otimes B) \leq s(B)$ applies, we consider optimal covers C and D for A and B . We then build a cover E for $A \otimes B$ such that we first split as many paths of C at Steiner vertices as possible in a way described in Remark 3. After this, all Steiner vertices of C have been removed and we combine each of the resulting paths with a path of D . Consequently, we get $|E| = p(A \otimes B) = p(B)$ and thus, $s(A \otimes B) \leq s(E) = s(B)$.

To show that $s(A \otimes B) \geq s(B)$ applies, we consider an optimal cover C for $A \otimes B$. By the above note we know that $s(C|_A) = 0$, otherwise, C would not be optimal.

Then, by $|C|_B| = p(B)$ and by Remark 4 it follows $s(A \otimes B) = s(C|_B) \geq s(B)$.

7. Can be shown similar to the previous item. $\square$

A directed co-tree can be computed in linear time from the input directed co-graph (Crespelle and Paul 2006). Combining this with Lemma 2 and 4, we come to the following result.

Theorem 2 *The value of a directed Steiner path cover of minimum cost for a directed co-graph can be computed in linear time with respect to the size of the directed co-expression.*

Lemma 3 allows us to minimize the following additional cost function.

Corollary 1 *The value of a directed Steiner path cover C for a directed co-graph G such that $|C| + s(C)$ is minimal can be computed in linear time with respect to the size of the directed co-expression.*

3.3 Computing an optimal directed Steiner path cover

Now, we give an algorithm to compute an optimal directed Steiner path cover for a directed co-graph. The function SERIESCOMP, see Algorithm 1, returns an optimal

Steiner path cover for a co-graph $G = A \otimes B$, if we have some optimal covers C_A for A and C_B for B given as parameter. Let $T_A \subseteq V(A)$ and $T_B \subseteq V(B)$ be the sets of terminal vertices, and let further $|T_A| \leq |T_B|$. The function $\text{APPEND}(p, u)$ used in Algorithm 1 in lines 4, 5, and 7 appends path p by vertex u . The function $\text{COMBINE}(p, p', u)$ used in lines 14 and 20, combines path p and p' by vertex u . The function $\text{REPLACE}(p, v, u)$ used in line 26 removes the vertex v from path p and replaces it by vertex u . The function $\text{INSERT}(p, v, u)$ used in line 29 inserts vertex u between vertex v and its successor in path p .

Algorithm 1 SERIESCOMP(set A , Cover C_A , set B , Cover C_B)

```

1: if  $|T_A| = |T_B|$  or  $|T_A| = |T_B| - 1$  then ▷ results in only one path
2:   let  $T'_A := T_A$ ; let  $T'_B := T_B$ ; let  $p := ()$ ;
3:   while  $T'_A \neq \emptyset$  do
4:     let  $b \in T'_B$ ;  $T'_B := T'_B - \{b\}$ ; APPEND( $p, b$ );
5:     let  $a \in T'_A$ ;  $T'_A := T'_A - \{a\}$ ; APPEND( $p, a$ );
6:   if  $T'_B \neq \emptyset$  then
7:     let  $b \in T'_B$ ;  $T'_B := T'_B - \{b\}$ ; APPEND( $p, b$ )
8:   return  $\{p\}$ 
9: let  $T'_A := T_A$ ; let  $C'_B := C_B$ ;
10: let  $p \in C'_B$ ;  $C'_B := C'_B - \{p\}$ ;
11: while  $T'_A \neq \emptyset$  and  $C'_B \neq \emptyset$  do ▷ combine 2 paths by a terminal vertex of A
12:   let  $a \in T'_A$ ;  $T'_A := T'_A - \{a\}$ ;
13:   let  $p' \in C'_B$ ;  $C'_B := C'_B - \{p'\}$ ;
14:   COMBINE( $p, p', a$ );
15: if  $T'_A = \emptyset$  then
16:   let  $U := V(A) - T_A$ ; ▷  $U$  is the set of non-terminal vertices of A
17:   while  $U \neq \emptyset$  and  $C'_B \neq \emptyset$  do ▷ combine 2 paths by a non-terminal vertex of A
18:     let  $u \in U$ ;  $U := U - \{u\}$ ;
19:     let  $p' \in C'_B$ ;  $C'_B := C'_B - \{p'\}$ ;
20:     COMBINE( $p, p', u$ )
21:   return  $\{p\} \cup C'_B$ 
22: if  $C'_B = \emptyset$  then
23:   while  $T'_A \neq \emptyset$  do
24:     let  $a \in T'_A$ ;  $T'_A := T'_A - \{a\}$ ;
25:     if  $\exists$  Steiner vertex  $v$  in path  $p$  then
26:       REPLACE( $p, v, a$ ) ▷ remove a Steiner vertex from B
27:     else
28:       let  $\{u, v\}$  be an edge between two terminal vertices of  $C_B$ 
29:       INSERT( $p, v, a$ ) ▷ add terminal vertices of A to the path
30:   return  $\{p\}$ 

```

Similar to function SERIESCOMP we introduce a function ORDERCOMP, see algorithm 2, which returns an optimal directed Steiner path cover for some directed co-graph $G = A \otimes B$, if some optimal covers for A and B are given as parameter. The function $\text{CONCAT}(p, p')$ used in line 16 extends path p by adding path p' at its end.

Let G be a directed co-graph represented by its binary directed co-tree $\mathcal{T}(G)$. The function DIRECTEDSTEINERPATHCOVER, see algorithm 3, recursively computes

Algorithm 2 ORDERCOMP(set A , Cover C_A , set B , Cover C_B)

```

1: let  $C'_A := C_A$ ; let  $C'_B := C_B$ ;
2: if  $|C'_A| < |C'_B|$  then
3:   loop min $\{|C'_A|, |C'_B| - |C'_A|\}$ -times
4:     let  $r \in C'_A$  that contains a Steiner vertex
5:      $C'_A := C'_A - \{r\}$ 
6:      $C'_A := C'_A \cup \{r_1, r_2\}$  where  $r_1, r_2$  result from  $r$  by splitting  $r$  at a Steiner vertex
7: if  $|C'_A| > |C'_B|$  then
8:   loop min $\{|C'_B|, |C'_A| - |C'_B|\}$ -times
9:     let  $r \in C'_B$  that contains a Steiner vertex
10:     $C'_B := C'_B - \{r\}$ 
11:     $C'_B := C'_B \cup \{r_1, r_2\}$  where  $r_1, r_2$  result from  $r$  by splitting  $r$  at a Steiner vertex
12:  $C := \emptyset$ 
13: loop min $\{|C'_A|, |C'_B|\}$ -times
14:   let  $a \in C'_A$ ;  $C'_A := C'_A - \{a\}$ ;
15:   let  $b \in C'_B$ ;  $C'_B := C'_B - \{b\}$ ;
16:   CONCAT( $a, b$ )
17:    $C := C \cup \{a\}$ 
18: while  $C'_A \neq \emptyset$  do
19:   let  $a \in C'_A$ ;  $C'_A := C'_A - \{a\}$ ;  $C := C \cup \{a\}$ 
20: while  $C'_B \neq \emptyset$  do
21:   let  $b \in C'_B$ ;  $C'_B := C'_B - \{b\}$ ;  $C := C \cup \{b\}$ 
22: return  $C$ 

```

---

a directed Steiner path cover of minimum cost of the subgraph of G induced by the vertices of $\mathcal{T}(G)$. For the series composition we assume that the left subtree $\mathcal{T}_{x_\ell}$ of any vertex x of $\mathcal{T}(G)$ contains no more terminal vertices than its right subtree $\mathcal{T}_{x_r}$. Otherwise we only had to swap the children of the vertex x .

Algorithm 3 DIRECTEDSTEINERPATHCOVER(Co-Tree $\mathcal{T}$, Vertex x)

```

if  $x$  is the only vertex of  $\mathcal{T}$  then
  if  $x$  is a terminal vertex of  $G$  then
    return  $\{(x)\}$ 
  return  $\emptyset$ 
else
   $C_A := \text{DIRECTEDSTEINERPATHCOVER}(\mathcal{T}_{x_\ell}, x_\ell)$ 
   $C_B := \text{DIRECTEDSTEINERPATHCOVER}(\mathcal{T}_{x_r}, x_r)$ 
  if  $x$  corresponds to a disjoint union of  $\mathcal{T}$  then
    return  $C_A \cup C_B$ 
  if  $x$  corresponds to a series composition of  $\mathcal{T}$  then
    return SERIESCOMP( $V(\mathcal{T}_{x_\ell}), C_A, V(\mathcal{T}_{x_r}), C_B$ )
  if  $x$  corresponds to an order composition of  $\mathcal{T}$  then
    return ORDERCOMP( $V(\mathcal{T}_{x_\ell}), C_A, V(\mathcal{T}_{x_r}), C_B$ )

```

By algorithm DIRECTEDSTEINERPATHCOVER we obtain the following result.

Theorem 3 *A directed Steiner path cover of minimum cost for a directed co-graph can be computed in linear time with respect to the size of the directed co-expression.*

Proof The correctness follows by Lemma 2 and 4. The running time can be achieved by storing the paths using double-linked, linear lists, where the paths that contain Steiner vertices are stored in one set and the paths that contain no Steiner vertices are stored in another set. The lists each have a pointer to the first and last element, which are terminal vertices, and they have a pointer to the first and last Steiner vertices. Additionally, we store the number of terminal and Steiner vertices for each list. Each of the operations in Algorithm DIRECTEDSTEINERPATHCOVER can be done in constant time. $\square$

4 Hamiltonian path problem

Our motivation to study the Hamiltonian Path problem comes from the problem of minimizing change-over times in pick-and-place machines in electronic industry. Before we study this problem, we recall the definition of Hamiltonian Path problem.

Name: Shortest Directed Hamiltonian Path

Instance: A directed graph G and edge weights $c : E(G) \rightarrow \mathbb{R}^{\geq 0}$.

Task: Find a directed Hamiltonian path P in G that minimizes $\sum_{e \in P} c(e)$.

Pick-and-place machines have been studied for many years, see Crama et al. (2002). Often, the focus of planning problems in printed circuit board assembly is on optimizing the throughput of these pick-and-place machines to produce many printed circuit boards (short PCB) a day. We are interested in minimizing change-over times for refilling the machines.

Mid-sized companies in the electronic industry often have to produce different PCBs on one day. Each of these PCBs has to be produced in small quantities. The SMD components like capacitors, resistors, or integrated circuits are usually positioned on the boards by pick-and-place machines and are soldered in a re-flow oven. Different types of components are fed in to the machine by feeders or trays. After one type of boards has been assembled, the pick-and-place machine has to be reassembled with other types of SMD components. The time for this refilling of components is often limited by the time the last PCBs of the previous series stay in the re-flow oven. The change-over must not take too long to avoid an unnecessary downtime of the machine. Therefore, PCBs must be processed in an order that only a few component groups have to be replaced. We show how a graph model can be used to describe the problem, and we give an integer program to solve the problem.

Assume that a company produces different PCBs, and the available pick-and-place machine can be equipped with at most k different types of SMD components, since no more trays are available. At one day, n different PCBs have to be produced, that together contain m different types of SMD components $t_1, \dots, t_m$. Each board b_i has to be equipped with a set $T_i \subset \{t_1, \dots, t_m\}$ of component types. We represent each board by an m -tuple such that the i -th entry is one if and only if type t_i is needed to produce the board. The number of ones in such a tuple is at most k . For example, $b_i = (0, 1, 0, 1, 0)$ means that components of type 2 and 4 must be used, types 1, 3, and 5 are not used for producing this board.

Fig. 6 Graph modeling 4 boards. The dashed lines show a Hamiltonian path that defines an order in which the change-over times sum up to 9

In our graph model, each board is represented by a vertex, and two vertices are connected by an edge. To change over from a board u to a board v those types have to be removed from the machine, that are used by u but not used by v . Those types, that are not used by u but are used by v have to be inserted in to the machine. Thus, the cost of change over is described by the hamming distance of u and v .

Example 1 Given $n = 4$ jobs $b_1 = (1, 0, 0, 0)$, $b_2 = (0, 1, 0, 1)$, $b_3 = (1, 0, 1, 0)$, and $b_4 = (1, 1, 1, 0)$. The resulting graph contains the artificial vertex $b_0 = (0, 0, 0, 0)$ to represent the initial state of the machine, where all trays are empty. It is shown in Fig. 6 together with the edge costs. To minimize the overall change-over time is to produce the boards in the order $b_1 = (1, 0, 0, 0)$, $b_3 = (1, 0, 1, 0)$, $b_4 = (1, 1, 1, 0)$, and $b_2 = (0, 1, 0, 1)$ which sum up to 6.

A shortest Hamiltonian path that starts at the artificial vertex representing the initial state of the machine minimizes the overall change-over time. In the next section we give a mixed integer program to solve the problem.

4.1 Integer program

In the following we assume that a directed graph $G = (V, E, c)$ is given, where V denotes the set of vertices, E denotes the set of edges, and c denotes a function that assigns the cost $c(e)$ to edge e . We represent an undirected graph G' by a directed graph G in such a way that each edge $e = \{u, v\}$ of G' is represented by two anti-parallel edges $e_1 = (u, v)$, $e_2 = (v, u)$ in G , and the weight of the new edges are $c(e_1) = c(e_2) = c(e)$.

Our integer program is based on the following idea. Choose a vertex as the first vertex on the path. This is typically an artificial vertex that represents the empty machine at the beginning of a day. The order of vertices on the path implies for two vertices u and v following immediately on the path, that the cost of edge (u, v) represents the change-over time.

Our integer program uses binary variables $x[i, j]$, $i, j \in [n]$, such that $x[i, j]$ is equal to one, if and only if edge (i, j) is on the Hamiltonian path. We use integer valued variables $p[i]$, $i \in [n]$, such that $p[i] = \ell$, if and only if vertex i is the ℓ -th vertex on the Hamiltonian path.

The first constraints ensure that the path is simple, i.e., each vertex is contained at most once in the path. The path leaves each vertex i at most once, and the path enters

each vertex j at most once, which can be stated as follows.

$$\sum_{(i,j) \in E} x[i, j] \leq 1 \quad \forall i \in V \quad \text{and} \quad \sum_{(i,j) \in E} x[i, j] \leq 1 \quad \forall j \in V$$

The following constraints prevent loops over anti-parallel edges. If there are anti-parallel edges then choose at most one of them. This is ensured by the constraints:

$$x[i, j] \leq 1 - x[j, i] \quad \forall (i, j) \in E : (j, i) \in E$$

The length (number of edges) of each Hamiltonian path is $n - 1$:

$$\sum_{(i,j) \in E} x[i, j] = n - 1$$

With just this constraint on the number of edges, we may accept solutions that are not connected. It is still possible to have cycles and a path instead of one path only. The next constraints are used to avoid cycles. If edge (i, j) belongs to the path then $p[j] - p[i] = 1$ must hold true:

$$\begin{aligned} -n + (n + 1) \cdot x[i, j] &\leq p[j] - p[i] \quad \forall (i, j) \in E \\ \text{and} \quad p[j] - p[i] &\leq n - (n - 1) \cdot x[i, j] \quad \forall (i, j) \in E \end{aligned}$$

The objective is to minimize the sum of the change-over times, i.e. to make the path length as small as possible, so we have to minimize the objective function $\sum_{(i,j) \in E} c[i, j] \cdot x[i, j]$.

Alternatively, if we want to minimize the largest change-over time, we have to introduce another integer valued variable z and the following constraints:

$$c[i, j] \cdot x[i, j] \leq z \quad \forall (i, j) \in E$$

The objective function then is to minimize the largest change-over time z .

4.2 Experimental results

We ran the mixed integer program on CPLEX 12.8.0 using default settings on some standard PC with Intel i7, 2.80 GHz CPU and 16 GB RAM running Ubuntu 18.10 as operating system.

Running times (in seconds) to solve the Hamiltonian Path problem on randomly generated tournaments with randomly generated edge weights are shown in Table 3. The running times are average values over 10 runs. Bold numbers represent times when the time limit of 1800 seconds, which is equal to 30 minutes, was reached at least once and at most twice. For the calculation of the average and standard deviation, we assumed that the run times of the aborted jobs were 1800 seconds. It is well-known that any tournament on a finite number of vertices contains a Hamiltonian path (Rédei 1934).

Table 3 Running times and standard deviation of MIP to calculate Hamiltonian Path on randomly generated tournaments

n	100	200	300	400	500
Time (s)	2.1	22.1	69.3	519	783
SD	0.7	19.1	81.6	741	885

Running times (in seconds) to solve the Hamiltonian Path problem on complete bipartite graphs $K_{n,m}$ with unit edge weights for different values of n and m are shown in Table 4. The complete bipartite graph $K_{n,m}$ contains a Hamiltonian path if and only if $|n - m| \leq 1$. The absence of such a path can be detected very quickly by the MIP as can be seen in the columns $m = n + 2$. All complete bipartite graphs are co-graphs so these results show that our algorithm with running time linear in the input size would be very much faster than the integer program. Running times of more than 30 minutes are indicated by a bar —.

5 Steiner path problem

5.1 Integer program

We assume that a directed graph $G = (V, E, c)$ is given, where V denotes the set of vertices, E denotes the set of edges, and c denotes a function that assigns the cost $c(e)$ to edge e . Further, we have a set $T \subseteq V$ of terminal vertices. We represent an undirected graph G' by a directed graph G in such a way that each edge $e = \{u, v\}$ of G' is represented by two anti-parallel edges $e_1 = (u, v)$, $e_2 = (v, u)$ in G , and the weight of the new edges are $c(e_1) = c(e_2) = c(e)$.

In addition to the variables used in the MIP of Sect. 4.1 we use additional binary variables $y[i]$, $i \in [n]$, such that $y[i]$ is equal to one, if and only if vertex i is contained in the path. These variables are only used to simplify the formulation of the following constraints 1 and 2. As in the last section, our integer program uses binary variables $x[i, j]$, $i, j \in [n]$, such that $x[i, j]$ is equal to one, if and only if edge (i, j) is on the Steiner path. We use integer valued variables $p[i]$, $i \in [n]$, such that $p[i] = \ell$, if and only if vertex i is the ℓ -th vertex on the Steiner path. Each terminal vertex i has to be contained in the path.

$$p[i] \geq 1 \quad \forall i \in T$$

If $x[i, j] = 1$ then i and j lie on the path, i.e. if $x[i, j] = 1$ then $p[i] \neq 0$ and $p[j] \neq 0$ which can be stated by the following constraints.

$$x[i, j] \leq p[i] \quad \forall (i, j) \in E \quad \text{and} \quad x[i, j] \leq p[j] \quad \forall (i, j) \in E$$

If there are anti-parallel edges then choose at most one of them. This is ensured by the constraints:

$$x[i, j] \leq 1 - x[j, i] \quad \forall (i, j) \in E : (j, i) \in E$$

Table 4 Running times of MIP to calculate Hamiltonian Path on complete bipartite graphs $K_{n,m}$ for different values of n and m

m	$n = 50$					$n = 100$					$n = 150$					$n = 200$					$n = 250$				
	50	51	52	100	101	102	150	151	152	200	201	202	250	251	252										
Time (s)	1.7	3.2	0.1	16.7	10.9	0.1	131	366	0.2	–	370	0.3	–	1774	0.5										

It holds $y[i] = 1$ if and only if $p[i] > 0$, i.e. $y[i] = 1$ if and only if vertex i is on the path.

$$y[i] \leq p[i] \quad \forall i \in V \quad \text{and} \quad p[i] \leq n \cdot y[i] \quad \forall i \in V$$

The path has to contain one edge less than vertices.

$$\sum_{(i,j) \in E} x[i, j] = \sum_{i \in V} y[i] - 1 \quad (1)$$

If $p[i] \neq 0$ then there has to be a $x[i, j] = 1$ or $x[j, i] = 1$.

$$p[i] \leq n \cdot \left(\sum_{(i,j) \in E} x[i, j] + \sum_{(j,i) \in E} x[j, i] \right) \quad \forall i \in V$$

For each vertex $i \in V - T$ contained in the path we have one in-going and one out-going edge.

$$\sum_{(i,j) \in E} x[i, j] = y[i] \quad \forall i \in V - T \quad \text{and} \quad \sum_{(i,j) \in E} x[j, i] = y[i] \quad \forall i \in V - T \quad (2)$$

The path leaves each vertex $i \in T$ at most once and it enters each vertex $j \in T$ at most once.

$$\sum_{(i,j) \in E} x[i, j] \leq 1 \quad \forall i \in T \quad \text{and} \quad \sum_{(i,j) \in E} x[j, i] \leq 1 \quad \forall j \in T$$

Finally, to avoid cycles, if edge (i, j) belongs to the path then $p[j] - p[i] = 1$ must hold true:

$$\begin{aligned} & -n + (n+1) \cdot x[i, j] \leq p[j] - p[i] \quad \forall (i, j) \in E \\ \text{and} \quad & p[j] - p[i] \leq n - (n-1) \cdot x[i, j] \quad \forall (i, j) \in E \end{aligned}$$

The objective is to minimize the sum of edge-weights $\sum_{(i,j) \in E} c[i, j] \cdot x[i, j]$. For graphs with unit edge weights this objective function results in a Steiner path with least number of Steiner vertices.

5.2 Experimental results

Running times (in seconds) to solve Steiner Path problems on complete bipartite graphs $K_{n,3n}$ for different values of n and t many randomly chosen terminal vertices are shown in Table 5. As long as at most n vertices from the second set are selected as terminal vertices, a Steiner path exists. The absence of such a path can be detected very quickly by the MIP as can be seen in the columns $t = 2n$.

Table 5 Running times and standard deviation to solve Steiner path problems on complete bipartite graphs $K_{n,3n}$ for different values of n and t

	$n = 25$				$n = 50$				$n = 75$				$n = 100$				$n = 125$			
	12	25	50	75	100	125	150	175	200	225	250	275	300	325	350	375	400	425	450	475
t	0.9	1.2	0.1	7.2	15.5	0.2	29.7	139	0.4	189	945	0.7	200	62	125	250	1.1	581	564	0.0
SD	0.0	0.2	0.0	0.5	5.0	0.0	1.4	128	0.0	152	644	0.0	644	581	564	0.0	0.0	581	564	0.0

Fig. 7 Graph extended by a source and a sink to compute an optimal Steiner Path Cover

The running times (in seconds) are average values over 10 runs, bold numbers represent times when the time limit of 1800 seconds, which is equal to 30 minutes, was reached at least once and at most twice. If the time limit was exceeded in more than half of the measurements, we marked this with a bar —. When calculating the average and the standard deviation, we assumed 1800s for the measurements that were stopped due to timeout.

6 Steiner path cover problem

6.1 Integer program

To simplify the objective function and the IP we modify the given directed graph $G = (V, E, c)$ in such a way, that a new vertex s as source and another new vertex $t \neq s$ as sink is added. $T \subseteq V$ is the set of terminal vertices with $T \neq \emptyset$. Vertex s gets edges to each terminal vertex and each terminal vertex gets an edge to the sink t as shown in Fig. 7.

The additional edges get a weight of $M := |V| \cdot |E|$ if the graph has no edge weights, otherwise $M := |V| \cdot C$ where $C := \max_{e \in E} \{c(e)\}$ denotes the maximum edge weight of the graph. More formally, we get graph $G' = (V', E', c')$ such that $V' := V \cup \{s, t\}$, $E' := E \cup \{(s, v) \mid v \in T\} \cup \{(v, t) \mid v \in T\}$, and $c'(e) = c(e)$ for each $e \in E$ and $c'(e) = M$ if $e = (s, v)$ or $e = (v, t)$ for some $v \in T$. If graph G has no edge weights then $c'(e) = 1$ for each $e \in E$, if G is an undirected graph we replace each edge $\{u, v\} \in E$ by two anti-parallel edges (u, v) and (v, u) , the weight of them are taken from the corresponding undirected edges.

As in the last section, our integer program uses binary variables $y[i]$, $i \in [n]$, such that $y[i]$ is equal to one, if and only if vertex i is contained in one of the paths. Binary variables $x[i, j]$, $i, j \in [n]$, are used such that $x[i, j]$ is equal to one, if and only if edge (i, j) is on the Steiner path cover. We use integer valued variables $p[i]$, $i \in [n]$, such that $p[i] = \ell$, if and only if vertex i is the ℓ -th vertex on one of the paths of the Steiner path cover.

The number of paths in the cover is equal to $\sum_{(s,v) \in E'} x[s, v]$, and since there is at least one path in a Steiner Path Cover for G if $T \neq \emptyset$, the following conditions must

hold true:

$$\sum_{(s,v) \in E'} x[s, v] > 0 \quad \text{and} \quad \sum_{(v,t) \in E'} x[v, t] > 0$$

Because we have inserted artificial vertices, the path leaves each terminal vertex i exactly once, and the path enters each terminal vertex j exactly once, which can be stated as follows.

$$\sum_{(i,j) \in E'} x[i, j] = 1 \quad \forall i \in T \quad \text{and} \quad \sum_{(i,j) \in E'} x[i, j] = 1 \quad \forall j \in T$$

Since non-terminal vertices need not be contained in the path, the path leaves each non-terminal vertex i at most once, and the path enters each non-terminal vertex j at most once, which can be stated as follows.

$$\sum_{(i,j) \in E} x[i, j] \leq 1 \quad \forall i \in V - T \quad \text{and} \quad \sum_{(i,j) \in E} x[i, j] \leq 1 \quad \forall j \in V - T$$

The previous two conditions also ensure that paths are vertex-disjoint. If a path contains a non-terminal vertex j then there has to be an edge that enters j and an edge that leaves j , which can be stated as:

$$\sum_{(i,j) \in E} x[i, j] = \sum_{(j,k) \in E} x[j, k] \quad \forall j \in V - T$$

If there are anti-parallel edges then choose at most one of them. This is ensured by the constraint:

$$x[i, j] \leq 1 - x[j, i] \quad \forall (i, j) \in E : (j, i) \in E$$

Each terminal vertex i has to be contained in one of the paths.

$$p[i] \geq 1 \quad \forall i \in T$$

If $x[i, j] = 1$ then i and j lie on one of the paths, i.e. if $x[i, j] = 1$ then $p[i] \neq 0$ and $p[j] \neq 0$ which can be stated by the following constraints.

$$x[i, j] \leq p[i] \quad \forall (i, j) \in E \quad \text{and} \quad x[i, j] \leq p[j] \quad \forall (i, j) \in E$$

If $p[i] \neq 0$ then there has to be a $x[i, j] = 1$ or $x[j, i] = 1$, which is stated by the following constraints.

$$p[i] \leq n \cdot \left(\sum_{(i,j) \in E} x[i, j] + \sum_{(j,i) \in E} x[j, i] \right) \quad \forall i \in V$$

Table 6 Running times and standard deviation to solve Steiner path cover problems on complete bipartite graphs $K_{n,3n}$ for different values of n and t

	$n = 25$						$n = 50$						$n = 75$						$n = 100$						$n = 125$					
	12	25	50	75	100	125	25	50	75	100	125	150	37	75	112	150	207	276	307	375	412	450	500	550	618	687	756	825	894	963
t	12	25	50	75	100	125	25	50	75	100	125	150	37	75	112	150	207	276	307	375	412	450	500	550	618	687	756	825	894	963
Time (s)	0.9	1.4	1.1	7.4	9.1	20.9	7.4	9.1	25.7	47.7	66.1	20.9	25.7	47.7	66.1	20.9	207	165.1	55.8	307	–	–	–	–	–	–	–	–	–	–
SD	0.2	0.3	0.1	1.9	2.1	2.6	2.5	2.1	4.5	27.1	66.1	2.6	4.5	27.1	66.1	2.6	165.1	6.1	276	618	58.6	–	–	–	–	–	–	–	–	–

Table 7 Running times and standard deviation to solve Steiner path cover problems on randomly generated co-graphs with n vertices and t many randomly chosen terminal vertices

t	$n = 200$			$n = 300$			$n = 400$			$n = 500$		
	$n/5$	$n/2$	$4n/5$	$n/5$	$n/2$	$4n/5$	$n/5$	$n/2$	$4n/5$	$n/5$	$n/2$	$4n/5$
Time (s)	9.3	13.4	19.7	21.8	72.9	337	46.1	191	413	121	524	708
SD	8.7	12.0	4.5	4.9	42.3	560	9.1	84.4	481	22.9	498	600

Finally, to avoid cycles, if edge (i, j) belongs to one of the paths then $p[j] - p[i] = 1$ must hold true:

$$\begin{aligned} -n + (n + 1) \cdot x[i, j] &\leq p[j] - p[i] \quad \forall (i, j) \in E \\ \text{and} \quad p[j] - p[i] &\leq n - (n - 1) \cdot x[i, j] \quad \forall (i, j) \in E \end{aligned}$$

Minimizing an objective function like $\sum_{v \in T} x[s, v]$ would only minimize the number of paths. If there are two different path covers of minimum size, we have to select that one with the least number of Steiner vertices. By our definition of the costs of the additional edges from s to terminal vertices, we can choose the same objective function as above, when we also take care of the additional edges, see the right part in Fig. 7. Each path would add at least $2 \cdot M$ to the value of the objective function. The objective is to minimize the sum of edge-weights $\sum_{(i,j) \in E'} c[i, j] \cdot x[i, j]$. For graphs with unit edge weights this objective function results in a Steiner path cover P of minimum size, with least number of Steiner vertices $\sum_{p \in P} \sum_{e \in p} c(e)$.

6.2 Experimental results

Running times (in seconds) to solve Steiner Path Cover problems on complete bipartite graphs $K_{n,3n}$ for different values of n and t many randomly chosen terminal vertices are shown in Table 6. The running times are average values over 10 runs, bold numbers represent times when the time limit was reached at least once and at most twice. If the time limit was exceeded in more than half of the measurements, we marked this with a bar —. When calculating the average and the standard deviation, we assumed 1800s for the measurements that were stopped due to timeout.

Table 7 shows running times to solve Steiner Path Cover problems on randomly generated co-graphs with n vertices and t many randomly chosen terminal vertices.

The co-graphs will be generated recursively as shown in Algorithm 4. The probability p is set to be $1/3$ in our experiments.

7 Conclusion

In this paper we considered the directed Steiner path cover problem for directed co-graphs. We could show a linear time solution for computing the minimum number of paths within a directed Steiner path cover and the minimum number of Steiner vertices in such a directed Steiner path cover in directed co-graphs. The results allowed us to

Algorithm 4 RANDOM- CO- GRAPH(int t , int n)

```

choose  $t$  distinct vertices at random
call OTIMES(1,  $n$ )
function OTIMES(int  $l$ , int  $r$ )
  if  $r - l = 1$  then
    create edge  $(l, r)$  and return
   $m := \text{rand}(l, r)$  ▷ choose random number between  $l$  and  $r$ 
  call OPLUS( $l, m$ ) with probability  $p$ , otherwise call OTIMES( $l, m$ )
  call OPLUS( $m+1, r$ ) with probability  $p$ , otherwise call OTIMES( $m+1, r$ )
  create edges  $(i, j)$  for all vertices  $l \leq i \leq m$ , and  $m+1 \leq j \leq r$ 
function OPLUS(int  $l$ , int  $r$ )
  if  $r - l = 1$  then
    return
   $m := \text{rand}(l, r)$ 
  call OPLUS( $l, m$ ) with probability  $p$ , otherwise call OTIMES( $l, m$ )
  call OPLUS( $m+1, r$ ) with probability  $p$ , otherwise call OTIMES( $m+1, r$ )

```

---

give an algorithm which constructs a directed Steiner path cover of minimum cost for a directed co-graph in linear time. This leads to a linear time computation of an optimal directed Steiner path, if it exists, for directed co-graphs.

Undirected co-graphs can be generated starting with the single vertex graph by disjoint union and join operations, see Corneil et al. (1981). For an undirected co-graph G , we can solve the Steiner path cover problem in linear time by the following transformation. We replace every edge $\{u, v\}$ of G by two directed edges (u, v) and (v, u) and apply our solution for directed co-graphs. This reproves our result of Gurski et al. (2020).

The directed Hamiltonian path problem can be solved by an XP-algorithm w.r.t. the parameter directed clique-width (Ganian et al. 2013). Since directed co-graphs have directed clique-width at most two (Gurski et al. 2016) a polynomial time solution for the directed Hamiltonian path problem follows. Such an algorithm is also given in Gurski (2017). A directed Hamiltonian path exists if and only if we have $T = V(G)$ and $p(G) = 1$. Thus, our results lead to the first linear time algorithm for the directed Hamiltonian path problem on directed co-graphs, which is a generalization of the known results for undirected co-graphs of Lin et al. (1995).

In our future work we want to investigate whether these results can be carried over to other graph classes such as chordal graphs, interval graphs, or distance-hereditary graphs.

Funding Open Access funding enabled and organized by Projekt DEAL. This work was funded in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 388221852.

Declarations

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

Availability of data and materials The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abu-Affash AK, Carmi P, Katz MJ, Segal M (2014) The Euclidean bottleneck Steiner path problem and other applications of (α, β) -pair decomposition. *Discrete Comput Geom* 51(1):1–23
- Bang-Jensen J, Maddaloni A (2014) Arc-disjoint paths in decomposable digraphs. *J Graph Theory* 77:89–110
- Bang-Jensen J, Gutin G (eds) (2018) *Classes of directed graphs*. Springer, Berlin
- Bechet D, de Groote P, Retoré C (1997) A complete axiomatization of the inclusion of series-parallel partial orders. In: *Rewriting techniques and applications*, volume 1232 of LNCS, pp 230–240. Springer 1997
- Bodlaender HL, Cygan M, Kratsch S, Nederlof J (2015) Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Inf Comput* 243:86–111
- Chimani M, Mutzel P, Zey B (2012) Improved Steiner tree algorithms for bounded treewidth. *J Discrete Algorithms* 16:67–78
- Corneil DG, Lerchs H, Stewart-Burlingham L (1981) Complement reducible graphs. *Discrete Appl Math* 3:163–174
- Crama Y, van de Klundert J, Spieksma FCR (2002) Production planning problems in printed circuit board assembly. *Discrete Appl Math* 123(1–3):339–361
- Crespelle C, Paul C (2006) Fully dynamic recognition algorithm and certificate for directed cographs. *Discrete Appl Math* 154(12):1722–1741
- Custic A, Lendl S (2021) The Steiner cycle and path cover problem on interval graphs. *J Comb Optim* (To appear)
- Fomin FV, Golovach PA, Lokshtanov D, Saurabh S (2010) Intractability of clique-width parameterizations. *SIAM J Comput* 39(5):1941–1956
- Ganian R, Hliněný P, Obdržálek J (2013) A unified approach to polynomial algorithms on graphs of bounded (bi-)rank-width. *Eur J Comb* 34(3):680–701
- Ganian R, Hliněný P, Kneis J, Langer A, Obdržálek J, Rossmanith P (2014) Digraph width measures in parameterized algorithmics. *Discrete Appl Math* 168:88–107
- Gurski F (2017) Dynamic programming algorithms on directed cographs. *Stat Optim Inf Comput* 5:35–44
- Gurski F, Wanke E, Yilmaz E (2016) Directed NLC-width. *Theoret Comput Sci* 616:1–17
- Gurski F, Komander D, Rehs C (2019) Oriented coloring on recursively defined digraphs. *Algorithms* 12(4):87
- Gurski F, Komander D, Rehs C (2021) How to compute digraph width measures on directed co-graphs. *Theoret Comput Sci* 855:161–185
- Gurski F, Hoffmann S, Komander D, Rehs C, Rethmann J, Wanke E (2020) Computing Directed Steiner path covers for directed co-graphs. In: *Proceedings of the conference on current trends in theory and practice of computer science (SOFSEM)*, volume 12011 of LNCS, pp 556–565. Springer
- Gurski F, Hoffmann S, Komander D, Rehs C, Rethmann J, Wanke E (2020) Exact solutions for the Steiner path problem on special graph classes. In: *Operations research proceedings (OR 2019)*, selected papers, pp 331–338. Springer
- Gurski F, Komander D, Rehs C (2019) Computing digraph width measures on directed co-graphs. In: *Proceedings of international symposium on fundamentals of computation theory (FCT)*, volume 11651 of LNCS, pp 292–305. Springer
- Gurski F, Rehs C (2018) Directed path-width and directed tree-width of directed co-graphs. In: *Proceedings of the international conference on computing and combinatorics (COCOON)*, volume 10976 of LNCS, pp 255–267. Springer

- Hellmuth M, Stadler PF, Wieseke N (2017) The mathematics of xenology: di-cographs, symbolic ultrametrics, 2-structures and tree-representable systems of binary relations. *J Math Biol* 75(1):199–237
- Lin R, Olariu S, Pruesse G (1995) An optimal path cover algorithm for cographs. *Comput Math Appl* 30:75–83
- Moharana SS, Joshi A, Vijay S (2013) Steiner path for trees. *Int J Comput Appl* 76(5):11–14
- Nojgaard N, El-Mabrouk N, Merkle D, Wieseke N, Hellmuth M (2018) Partial homology relations–satisfiability in terms of di-cographs. In: *Proceedings of international computing and combinatorics conference (COCOON)*, volume 10976 of LNCS, pp 403–415. Springer
- Rédei L (1934) Ein kombinatorischer Satz. *Acta Litteraria Szeged* 7:39–43
- Reich G, Widmayer P (1990) Beyond Steiner’s problem: a VLSI oriented generalization. In: *Proceedings of graph-theoretical concepts in computer science (WG)*, volume 411 of LNCS, pp 196–210. Springer
- Wald JA, Colbourn CJ (1982) Steiner trees in outerplanar graphs. In: *Thirteenth southeastern conference on combinatorics, graph theory, and computing*, pp 15–22
- Wald JA, Colbourn CJ (1983) Steiner trees, partial 2-trees, and minimum IFI networks. *Networks* 13:159–167
- Westbrook J, Yan D (1995) Approximation algorithms for the class Steiner tree problem. Research Report

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