

Gurski, Frank; Komander, Dominique; Rehs, Carolin

**Article — Published Version**

## Solutions for subset sum problems with special digraph constraints

Mathematical Methods of Operations Research

**Provided in Cooperation with:**

Springer Nature

*Suggested Citation:* Gurski, Frank; Komander, Dominique; Rehs, Carolin (2020) : Solutions for subset sum problems with special digraph constraints, Mathematical Methods of Operations Research, ISSN 1432-5217, Springer, Berlin, Heidelberg, Vol. 92, Iss. 2, pp. 401-433, <https://doi.org/10.1007/s00186-020-00718-6>

This Version is available at:

<https://hdl.handle.net/10419/288447>

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**Terms of use:**

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*



<https://creativecommons.org/licenses/by/4.0/>



# Solutions for subset sum problems with special digraph constraints

Frank Gurski<sup>1</sup> · Dominique Komander<sup>1</sup> · Carolin Rehs<sup>1</sup>

Received: 24 June 2019 / Revised: 22 May 2020  
© The Author(s) 2020

## Abstract

The subset sum problem is one of the simplest and most fundamental NP-hard problems in combinatorial optimization. We consider two extensions of this problem: The subset sum problem with digraph constraint (SSG) and subset sum problem with weak digraph constraint (SSGW). In both problems there is given a digraph with sizes assigned to the vertices. Within SSG we want to find a subset of vertices whose total size does not exceed a given capacity and which contains a vertex if at least one of its predecessors is part of the solution. Within SSGW we want to find a subset of vertices whose total size does not exceed a given capacity and which contains a vertex if all its predecessors are part of the solution. SSG and SSGW have been introduced recently by Gourvès et al. who studied their complexity for directed acyclic graphs and oriented trees. We show that both problems are NP-hard even on oriented co-graphs and minimal series-parallel digraphs. Further, we provide pseudo-polynomial solutions for SSG and SSGW with digraph constraints given by directed co-graphs and series-parallel digraphs.

**Keywords** Subset sum problem · Digraph constraint · Directed co-graphs · Series-parallel digraphs

**Mathematics Subject Classification** 05C85 · 90C39 · 05C69

## 1 Introduction

The subset sum problem is one of the most fundamental NP-hard problems in combinatorial optimization. Within the *subset sum problem (SSP)* there is given a set  $A = \{a_1, \dots, a_n\}$  of  $n$  items. Every item  $a_j$  has a size  $s_j$  and there is a capacity  $c$ . All

---

✉ Frank Gurski  
frank.gurski@hhu.de

<sup>1</sup> Institute of Computer Science, Algorithmics for Hard Problems Group, University of Düsseldorf, 40225 Düsseldorf, Germany

values are assumed to be positive integers and  $s_j \leq c$  for every  $j \in \{1, \dots, n\}$ . The task is to choose a subset  $A'$  of  $A$ , such that the sum of the sizes of the items in  $A'$  is maximized and is at most  $c$ .

In this paper we consider the following two sum problems which additionally have given a digraph on the item set. Both problems have been introduced recently by Gourvès et al. (2018). Within the *subset sum problem with digraph constraint* (SSG) we want to find a subset of vertices whose total size does not exceed a given capacity and which contains a vertex if at least one of its predecessors is part of the solution. Within the *subset sum problem with weak digraph constraint* (SSGW) the goal is to find a subset of vertices whose total size does not exceed a given capacity and which contains a vertex if all its predecessors are part of the solution. Since SSG and SSGW generalize SSP, they are NP-hard. Both problems are integer-valued problems, which motivates to observe whether they are weakly NP-hard, i.e. the existence of pseudo-polynomial algorithms.

For related works we refer to Gourvès et al. (2018, Section 3). In Gourvès et al. (2018) it has been shown that on directed acyclic graphs (DAGs) SSG is strongly NP-hard and SSGW is even APX-hard. Further, they showed that the restriction to oriented trees allows to give a pseudo-polynomial algorithm using dynamic programming along the tree.

In this paper we consider SSG and SSGW on further special digraph classes. First, we consider co-graphs (short for complement reducible graphs), which can be generated from the single vertex graph by applying disjoint union, order composition and series composition (Bechet et al 1997). They can also be characterized by excluding eight forbidden induced subdigraphs. Directed co-graphs are exactly the digraphs of directed NLC-width<sup>1</sup> 1 and a proper subset of the digraphs of directed clique-width at most 2 (Gurski et al. 2016). Directed co-graphs are interesting from an algorithmic point of view since several hard graph problems can be solved in polynomial time by dynamic programming along the tree structure of the input graph, see Bang-Jensen and Maddaloni (2014), Gurski (2017), Gurski and Rehs (2018), Gurski et al. (2019a,b). Moreover, directed co-graphs are very useful for the reconstruction of the evolutionary history of genes or species using genomic sequence data (Hellmuth et al. 2017; Nojgaard et al. 2018).

Further, we look at SSG and SSGW on series-parallel digraphs, which are exactly the digraphs whose transitive closure equals the transitive closure of some minimal series-parallel digraph. Minimal series-parallel digraphs can be generated from the single vertex graph by applying the parallel composition and series composition (Valdes et al. 1982). Series-parallel digraphs are also interesting from an algorithmic point of view since several hard graph problems can be solved in polynomial time by dynamic programming along the tree structure of the input graph, see Monma and Sidney (1977), Steiner (1985) and Rendl (1986).

We show pseudo-polynomial solutions for SSG and SSGW on directed co-graphs and minimal series-parallel digraphs and deduce a pseudo-polynomial time solution for SSG on series-parallel digraphs. Our results are based on dynamic programming

<sup>1</sup> The abbreviation NLC results from the node label controlled embedding mechanism originally defined for graph grammars (Engelfriet and Rozenberg 1997).

**Table 1** Known running times for SSG and SSGW with digraph constraints restricted to special graph classes

	SSG		SSGW	
Transitive tournaments	$\mathcal{O}(n^2)$	Remark 2	$\mathcal{O}(n \cdot c^4 + m)$	Theorem 3
Bioriented cliques	$\mathcal{O}(n)$	Remark 3	$\mathcal{O}(n \cdot c^4 + m)$	Theorem 3
DAGs	Strongly NP-hard	Gourvès et al. (2018)	APX-hard	Gourvès et al. (2018)
Oriented trees	$\mathcal{O}(n \cdot c^3)$	Gourvès et al. (2018)	$\mathcal{O}(n \cdot c^2)$	Gourvès et al. (2018)
Directed co-graphs	$\mathcal{O}(n \cdot c^2 + m)$	Theorem 2	$\mathcal{O}(n \cdot c^4 + m)$	Theorem 3
Minimal series-parallel	$\mathcal{O}(n \cdot c^2 + m)$	Theorem 5	$\mathcal{O}(n \cdot c^4 + m)$	Theorem 7
Series-parallel	$\mathcal{O}(n \cdot c^2 + n^{2.37})$	Theorem 6	Open	

Let  $n$  be the number of vertices and  $m$  the number of directed edges of the input digraph and  $c$  be the capacity

along the tree-structure of the considered digraphs. The considered digraph classes are incomparable w.r.t. inclusion to oriented trees considered in Gourvès et al. (2018), see Fig. 7. Moreover, the digraphs of our interest allow to define dense graphs, i.e. graphs where the number of directed edges is quadratic in the number of vertices. In Table 1 we summarize the known results from Gourvès et al. (2018) and the results of this work about subset sum problems with special digraph constraints.

## 2 Preliminaries

### 2.1 Digraphs

A *directed graph* or *digraph* is a pair  $G = (V, E)$ , where  $V$  is a finite set of *vertices* and  $E \subseteq \{(u, v) \mid u, v \in V, u \neq v\}$  is a finite set of ordered pairs of distinct vertices called *arcs* or *directed edges*. For a vertex  $v \in V$ , the sets  $N_G^+(v) = \{u \in V \mid (v, u) \in E\}$  and  $N_G^-(v) = \{u \in V \mid (u, v) \in E\}$  are called the *set of all successors* and the *set of all predecessors* of  $v$  in  $G$ . The *outdegree* of  $v$ ,  $\text{outdegree}_G(v)$  for short, is the number of successors of  $v$  and the *indegree* of  $v$ ,  $\text{indegree}_G(v)$  for short, is the number of predecessors of  $v$ . We may omit indices if the graph under consideration is clear from the context. A vertex  $v \in V$  is *out-dominating* (*in-dominated*) if it is adjacent to every other vertex in  $V$  and is a source (a sink, respectively).

A digraph  $G' = (V', E')$  is a *subdigraph* of digraph  $G = (V, E)$  if  $V' \subseteq V$  and  $E' \subseteq E$ . If every arc of  $E$  with start- and end-vertex in  $V'$  is in  $E'$ , we say that  $G'$  is an *induced subdigraph* of digraph  $G$  and we write  $G' = G[V']$ .

For  $n \geq 2$  we denote by

$$\vec{P}_n = (\{v_1, \dots, v_n\}, \{(v_1, v_2), \dots, (v_{n-1}, v_n)\})$$

a *directed path* on  $n$  vertices. Vertex  $v_1$  is the *start vertex* and  $v_n$  is the *end vertex* of  $\vec{P}_n$ . For  $n \geq 2$  we denote by

$$\vec{C}_n = (\{v_1, \dots, v_n\}, \{(v_1, v_2), \dots, (v_{n-1}, v_n), (v_n, v_1)\})$$

a directed cycle on  $n$  vertices.

A directed acyclic graph (DAG for short) is a digraph without any  $\vec{C}_n$ , for  $n \geq 2$ , as subdigraph. A vertex  $v$  is *reachable* from a vertex  $u$  in  $G$ , if  $G$  contains a  $\vec{P}_n$  as a subdigraph having start vertex  $u$  and end vertex  $v$ .

A *weakly connected component* of  $G$  is a maximal subdigraph, such that the corresponding underlying graph is connected. A *strongly connected component* of  $G$  is a maximal subdigraph, in which every vertex is reachable from every other vertex.

An *out-rooted-tree* (*in-rooted-tree*) is an orientation of a tree with a distinguished root such that all arcs are directed away from (to) the root.

## 2.2 Problems

Let  $A = \{a_1, \dots, a_n\}$  be a set of  $n$  items, such that every item  $a_j$  has a size  $s_j$ . For a subset  $A'$  of  $A$  we define

$$s(A') := \sum_{a_j \in A'} s_j$$

and the *capacity constraint* by

$$s(A') \leq c. \tag{1}$$

**Name:** Subset sum (SSP)

**Instance:** A set  $A = \{a_1, \dots, a_n\}$  of  $n$  items. Every item  $a_j$  has a size  $s_j$  and there is a capacity  $c$ .

**Task:** Find a subset  $A'$  of  $A$  that maximizes  $s(A')$  subject to (1).

The parameters  $n$ ,  $s_j$ , and  $c$  are assumed to be positive integers. See Kellerer et al. (2010, Chapter 4) for a survey on the subset sum problem. In order to consider generalizations of the subset sum problem we will consider constraints for a digraph  $G = (A, E)$  with objects assigned to the vertices.

The *digraph constraint* ensures that  $A' \subseteq A$  contains a vertex  $y$ , if it contains at least one predecessor of  $y$ , i.e.

$$\forall y \in A (N^-(y) \cap A' \neq \emptyset) \Rightarrow y \in A'. \tag{2}$$

The *weak digraph constraint* ensures that  $A'$  contains a vertex  $y$ , if it contains every predecessor of  $y$ , i.e.

$$\forall y \in A (N^-(y) \subseteq A' \wedge N^-(y) \neq \emptyset) \Rightarrow y \in A'. \tag{3}$$

This allows us to state the following optimization problems given in Gourvès et al. (2018).

- Name:** Subset sum with digraph constraint (SSG)  
**Instance:** A set  $A = \{a_1, \dots, a_n\}$  of  $n$  items and a digraph  $G = (A, E)$ . Every item  $a_j$  has a size  $s_j$  and there is a capacity  $c$ .  
**Task:** Find a subset  $A'$  of  $A$  that maximizes  $s(A')$  subject to (1) and (2).
- Name:** Subset sum with weak digraph constraint (SSGW)  
**Instance:** A set  $A = \{a_1, \dots, a_n\}$  of  $n$  items and a digraph  $G = (A, E)$ . Every item  $a_j$  has a size  $s_j$  and there is a capacity  $c$ .  
**Task:** Find a subset  $A'$  of  $A$  that maximizes  $s(A')$  subject to (1) and (3).

In our problems the parameters  $n$ ,  $s_j$ , and  $c$  are assumed to be positive integers.<sup>2</sup> Further, in the defined problems a subset  $A'$  of  $A$  is called *feasible*, if it satisfies the prescribed constraints of the problem. By  $OPT(I)$  we denote the value of an optimal solution for input  $I$ .

**Observation 1** *Every feasible solution for SSG is also a feasible solution for SSGW, but not vice versa.*

**Observation 2**  $A' = \emptyset$  and  $A' = A$  for  $s(A) \leq c$  are feasible solutions for every instance of SSG and for every instance of SSGW.

In order to give equivalent characterizations for SSG and SSGW we use binary integer programs.

**Remark 1** To formulate SSG and SSGW as a binary integer program, we introduce a binary variable  $x_j \in \{0, 1\}$  for each item  $a_j \in A$ ,  $1 \leq j \leq n$ . The idea is to have  $x_j = 1$  if and only if item  $a_j \in A'$ .

- SSG corresponds to maximizing  $\sum_{j=1}^n s_j x_j$  subject to  $\sum_{j=1}^n s_j x_j \leq c$ ,  $x_i \leq x_j$  for every  $j \in \{1, \dots, n\}$  and for every  $a_i \in N^-(a_j)$ , and  $x_j \in \{0, 1\}$  for every  $j \in \{1, \dots, n\}$ .
- SSGW corresponds to maximizing  $\sum_{j=1}^n s_j x_j$  subject to  $\sum_{j=1}^n s_j x_j \leq c$ ,  $\sum_{\{i|a_i \in N^-(a_j)\}} x_i \leq x_j + \text{indegree}(a_j) - 1$  for every  $j \in \{1, \dots, n\}$ , and  $x_j \in \{0, 1\}$  for every  $j \in \{1, \dots, n\}$ .

The complexity for SSG and SSGW restricted to DAGs and oriented trees was considered in Gourvès et al. (2018).

**Theorem 1** (Gourvès et al. 2018) *On DAGs SSG is strongly NP-hard and SSGW is APX-hard.*

## 2.3 Basic results

Let  $G = (V, E)$  be a digraph and  $x \in V$ . By  $R_x$  we denote the vertices of  $V$  which are reachable from  $x$  and by  $S_x$  we denote the vertices of  $V$  which are in the same strongly connected component as  $x$ . Thus, it holds that  $\{x\} \subseteq S_x \subseteq R_x \subseteq V$ .

<sup>2</sup> The results in Gourvès et al. (2018) also consider null sizes, which are excluded in our work. All our solutions can be extended to pseudopolynomial solutions which solve SSG and SSGW using null sizes, see Sect. 5.

**Lemma 1** *Let  $A'$  be a feasible solution for SSG on a digraph  $G = (A, E)$  and  $x \in A$ . Then, it holds that  $x \in A'$  if and only if  $R_x \subseteq A'$ .*

**Lemma 2** *Let  $A'$  be a feasible solution for SSG on a digraph  $G = (A, E)$  and  $x \in A$ . Then, it holds that  $x \in A'$  if and only if  $S_x \subseteq A'$ .*

**Lemma 3** *SSG is solvable in  $\mathcal{O}(2^t \cdot (n + m))$  time on digraphs with  $n$  vertices,  $m$  arcs, and  $t$  strongly connected components.*

**Proof** By Lemma 2 for every feasible solution  $A'$  and every strongly connected component  $S$ , it either holds that  $S \subseteq A'$  or  $S \cap A' = \emptyset$ . Since all strongly connected components are vertex disjoint, we can solve SSG by verifying  $2^t$  possible feasible solutions. Verifying the capacity constraint can be done in  $\mathcal{O}(n)$  time and verifying the digraph constraint can be done in  $\mathcal{O}(n + m)$  time.  $\square$

In the condensation  $con(G)$  of a digraph  $G = (V, E)$  every strongly connected component  $C$  of  $G$  is represented by a vertex  $v_C$  and there is an arc between two vertices  $v_C$  and  $v_{C'}$  if there exist  $u \in C$  and  $v \in C'$ , such that  $(u, v) \in E$ . For every digraph  $G$  it holds that  $con(G)$  is a directed acyclic graph.

In order to solve SSG it is useful to consider the condensation of the input digraph  $G = (A, E)$ . By defining the size of a vertex  $v_C$  of  $con(G)$  by the sum of the sizes of the vertices in  $C$ , the following result has been shown in Gourvès et al. (2018, Lemma 2).

**Lemma 4** (Gourvès et al. 2018) *For a given instance of SSG on digraph  $G$ , there is a bijection between the feasible solutions (and thus the set of optimal solutions) of SSG for  $G$  and the feasible solutions (and thus the set of optimal solutions) for  $con(G)$ .*

Thus, in order to solve SSG we can restrict ourselves to DAGs by computing the condensation of the input graph in a first step. The next example shows that Lemma 4 does not hold for SSGW.

**Example 1** We consider the digraph  $G$  in Fig. 1. For SSGW with  $c = 2$  and all sizes  $s_j = 1$  we have among others  $\{a_4\}$  as a feasible solution. Since  $con(G)$  is a path of length one, formally

$$con(G) = (\{v_{\{a_1, a_2, a_3, a_4\}}, v_{\{a_5\}}\}, \{(v_{\{a_1, a_2, a_3, a_4\}}, v_{\{a_5\}})\}),$$

the only feasible solution is  $\{a_5\}$ , which implies that  $\{a_4\}$  is not a feasible solution for SSGW using  $con(G)$ .

The transitive closure  $td(G)$  of a digraph  $G$  has the same vertex set as  $G$  and for two distinct vertices  $u, v$  there is an arc  $(u, v)$  in  $td(G)$  if and only if there is a directed path from  $u$  to  $v$  in  $G$ . The transitive reduction  $tr(G)$  of a digraph  $G$  has the same vertex set as  $G$  and as few arcs of  $G$  as possible, such that  $G$  and  $tr(G)$  have the same transitive closure. The transitive closure is unique for every digraph. The transitive reduction is unique for directed acyclic graphs. However, for arbitrary digraphs the transitive reduction is not unique. The time complexity of the best known algorithm

Fig. 1 Digraph in Example 1

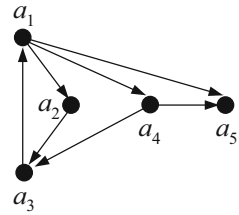
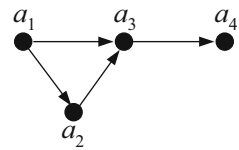


Fig. 2 Digraph in Example 2



for finding the transitive reduction of a graph is the same as the time to compute the transitive closure of a graph or to perform Boolean matrix multiplication (Aho et al. 1972). The best known algorithm to perform Boolean matrix multiplication has running time  $\mathcal{O}(n^{2.3729})$  by Le Gall (2014).

**Lemma 5** *For a given instance of SSG on a directed acyclic graph  $G$ , the set of feasible solutions and thus the set of optimal solutions of SSG for  $G$  and for  $tr(G)$  are equal.*

**Proof** Since a transitive reduction is a subdigraph of the given graph, every feasible solution  $A'$  for  $G$  is also a feasible solution for the transitive reduction  $tr(G)$ . To show the reverse direction, let  $A'$  be a feasible solution for  $tr(G)$ . By the definition of  $tr(G)$  we know that for every vertex  $v$ , every predecessor  $u$  of  $v$  in  $G$  is also a predecessor of  $v$  in  $tr(G)$  or there is a path from  $u$  to  $v$  in  $tr(G)$ . By Lemma 1 we know that  $A'$  is also a feasible solution for  $G$ .  $\square$

Thus, in order to solve SSG we can restrict ourselves to transitive reductions. The next example shows that Lemma 5 does not hold for SSGW.

**Example 2** We consider the digraph  $G$  in Fig. 2. For SSGW with  $c = 2$  and all sizes  $s_j = 1$  we have among others  $\{a_2\}$  as a feasible solution. Since  $tr(G)$  is a path, formally

$$tr(G) = (\{a_1, a_2, a_3, a_4\}, \{(a_1, a_2), (a_2, a_3), (a_3, a_4)\}),$$

$a_2$  implies by (3) that  $a_3$  and  $a_4$  must be part of the solution, which implies that  $\{a_2\}$  is not a feasible solution for SSGW using  $tr(G)$ .

In the correctness proofs of our algorithms in Sects. 3 and 4 we will use the following lemmata.

**Lemma 6** *Let  $G = (V_G, E_G)$  be a digraph and let  $H = (V_H, E_H)$  be an induced subdigraph of  $G$ . If  $A'$  is a feasible solution for SSG on  $G$ , then  $A' \cap V_H$  is a feasible solution for SSG on  $H$ .*



**Proof** If  $A'$  is a feasible solution for SSG on  $G$ , then it holds that

$$\forall y \in V_G (N_G^-(y) \cap A' \neq \emptyset) \Rightarrow y \in A'.$$

By restricting to  $y$  having no predecessors from  $V_G \setminus V_H$ , we obtain

$$\forall y \in V_G (N_G^-(y) \cap A' \cap V_H \neq \emptyset) \Rightarrow y \in A'.$$

By restricting  $y$  to  $V_H \subseteq V_G$  we obtain

$$\forall y \in V_H (N_H^-(y) \cap A' \cap V_H \neq \emptyset) \Rightarrow y \in A' \cap V_H,$$

i.e.,  $A' \cap V_H$  is a feasible solution for SSG on  $H$ . □

The reverse direction of Lemma 6 does not hold, since vertices with predecessors in  $A' \cap (V_G \setminus V_H)$  are not considered by the feasible solutions for SSG on  $H$ . By considering the induced subdigraph  $H = (\{a_2, a_3, a_4\}, \{(a_2, a_3), (a_3, a_4)\})$  of digraph  $G$  in Example 2 we observe that Lemma 6 does not hold for SSGW. Next, we give two weaker forms of Lemma 6 which also hold for SSGW.

**Lemma 7** *Let  $G = (V_G, E_G)$  be a digraph and let  $H = (V_H, E_H)$  be a weakly connected component of  $G$ . If  $A'$  is a feasible solution for SSGW on  $G$ , then  $A' \cap V_H$  is a feasible solution for SSGW on  $H$ .*

**Proof** If  $A'$  is a feasible solution for SSGW on  $G$ , then it holds that

$$\forall y \in V_G (N_G^-(y) \subseteq A' \wedge N_G^-(y) \neq \emptyset) \Rightarrow y \in A'.$$

By restricting  $y$  to  $V_H \subseteq V_G$  we obtain

$$\forall y \in V_H (N_G^-(y) \subseteq A' \wedge N_G^-(y) \neq \emptyset) \Rightarrow y \in A' \cap V_H.$$

Since  $H$  is a weakly connected component of  $G$  for all  $y \in V_H$  it holds that  $N_H^-(y) = N_G^-(y)$  such that

$$\forall y \in V_H (N_H^-(y) \subseteq A' \cap V_H \wedge N_H^-(y) \neq \emptyset) \Rightarrow y \in A' \cap V_H,$$

i.e.,  $A' \cap V_H$  is a feasible solution for SSGW on  $H$ . □

**Lemma 8** *Let  $G = (V_G, E_G)$  be a digraph and let  $H = (V_H, E_H)$  be an induced subdigraph of  $G$ , such that no non-source of  $H$  has a predecessor in  $V_G \setminus V_H$ . If  $A'$  is a feasible solution for SSGW on  $G$ , then  $A' \cap V_H$  is a feasible solution for SSGW on  $H$ .*

**Proof** If  $A'$  is a feasible solution for SSGW on  $G$ , then it holds that

$$\forall y \in V_G (N_G^-(y) \subseteq A' \wedge N_G^-(y) \neq \emptyset) \Rightarrow y \in A'.$$

By restricting  $y$  to  $V_H \subseteq V_G$  we obtain that

$$\forall y \in V_H (N_G^-(y) \subseteq A' \wedge N_G^-(y) \neq \emptyset) \Rightarrow y \in A' \cap V_H.$$

By restricting  $y$  to be a non-source of  $H$ , we obtain

$$\forall y \in V_H (N_G^-(y) \subseteq A' \wedge N_G^-(y) \neq \emptyset \wedge N_H^-(y) \neq \emptyset) \Rightarrow y \in A' \cap V_H.$$

Since no non-source of  $H$  has a predecessor in  $V_G \setminus V_H$ , we obtain

$$\forall y \in V_H (N_G^-(y) \subseteq A' \wedge N_H^-(y) \neq \emptyset) \Rightarrow y \in A' \cap V_H.$$

Then it holds in  $H$  that

$$\forall y \in V_H (N_H^-(y) \subseteq A' \cap V_H \wedge N_H^-(y) \neq \emptyset) \Rightarrow y \in A' \cap V_H,$$

i.e.,  $A' \cap V_H$  is a feasible solution for SSGW on  $H$ . □

**Lemma 9** *Let  $G = (V_G, E_G)$  be a digraph such that there is a 2-partition  $(V_1, V_2)$  of  $V_G$  with  $\{(u, v) \mid u \in V_1, v \in V_2\} \subseteq E_G$ . If  $A'$  is a feasible solution for SSGW on  $G$  such that  $V_1 \subseteq A'$ , then  $A' \cap V_2$  is a feasible solution for SSGW on  $G[V_2]$ .*

**Proof** If  $A'$  is a feasible solution for SSGW on  $G$ , then it holds that

$$\forall y \in V_G (N_G^-(y) \subseteq A' \wedge N_G^-(y) \neq \emptyset) \Rightarrow y \in A'.$$

By restricting  $y$  to  $V_2 \subseteq V_G$  we obtain

$$\forall y \in V_2 (N_G^-(y) \subseteq A' \wedge N_G^-(y) \neq \emptyset) \Rightarrow y \in A' \cap V_2.$$

Since  $V_1 \subseteq A'$  it holds that

$$\forall y \in V_2 (N_G^-(y) \subseteq V_1 \cup A' \cap V_2 \wedge N_G^-(y) \neq \emptyset) \Rightarrow y \in A' \cap V_2.$$

Thus, it holds that

$$\forall y \in V_2 (N_{G[V_2]}^-(y) \subseteq A' \cap V_2 \wedge (N_{G[V_1]}^-(y) \cup N_{G[V_1]}^-(y)) \neq \emptyset) \Rightarrow y \in A' \cap V_2.$$

Since  $V_1 = N_{G[V_1]}^-(y) \neq \emptyset$  it holds that

$$\forall y \in V_2 (N_{G[V_2]}^-(y) \subseteq A' \cap V_2) \Rightarrow y \in A' \cap V_2.$$

By the properties of the logical implication it also holds that

$$\forall y \in V_2 (N_{G[V_2]}^-(y) \subseteq A' \cap V_2 \wedge N_{G[V_2]}^-(y) \neq \emptyset) \Rightarrow y \in A' \cap V_2,$$

i.e.,  $A' \cap V_2$  is a feasible solution for SSGW on  $G[V_2]$ .  $\square$

Further, we will use the following result for solutions of SSP on digraphs with sizes assigned to the vertices.

**Observation 3** *Let  $G = (V_G, E_G)$  be a digraph with sizes assigned to the vertices and let  $H = (V_H, E_H)$  be an induced subdigraph of  $G$ . If  $A' \subseteq V_G$  satisfies (1), then  $A' \cap V_H$  satisfies (1).*

### 3 SSG and SSGW on directed co-graphs

#### 3.1 Directed co-graphs

Let  $G_1 = (V_1, E_1)$  and  $G_2 = (V_2, E_2)$  be two vertex-disjoint digraphs. The following operations have already been considered by Bechet et al (1997).

- The *disjoint union* of  $G_1$  and  $G_2$ , denoted by  $G_1 \oplus G_2$ , is the digraph with vertex set  $V_1 \cup V_2$  and arc set  $E_1 \cup E_2$ .
- The *series composition* of  $G_1$  and  $G_2$ , denoted by  $G_1 \otimes G_2$ , is defined by their disjoint union plus all possible arcs between vertices of  $G_1$  and  $G_2$ .
- The *order composition* of  $G_1$  and  $G_2$ , denoted by  $G_1 \oslash G_2$ , is defined by their disjoint union plus all possible arcs from vertices of  $G_1$  to vertices of  $G_2$ .

We recall the definition of directed co-graphs from Crespelle and Paul (2006).<sup>3</sup>

**Definition 1** (Directed co-graphs, Crespelle and Paul 2006) The class of *directed co-graphs* is recursively defined as follows.

1. Every digraph with a single vertex  $(\{v\}, \emptyset)$ , denoted by  $v$ , is a *directed co-graph*.
2. If  $G_1$  and  $G_2$  are vertex-disjoint directed co-graphs, then
  - (a) the disjoint union  $G_1 \oplus G_2$ ,
  - (b) the series composition  $G_1 \otimes G_2$ , and
  - (c) the order composition  $G_1 \oslash G_2$  are *directed co-graphs*.

The class of directed co-graphs is denoted by DC.

Every expression  $X$  using the four operations of Definition 1 is called a *di-co-expression* and  $\text{digraph}(X)$  is the defined digraph.

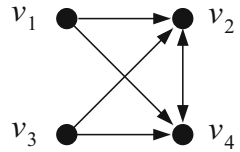
**Example 3** The di-co-expression

$$X = ((v_1 \oplus v_3) \oslash (v_2 \otimes v_4)) \quad (4)$$

defines  $\text{digraph}(X)$  shown in Fig. 3.

<sup>3</sup> In Crespelle and Paul (2006) directed co-graphs are defined by disjoint union, series composition, and order composition combining an arbitrary number of digraphs. We restrict ourselves to binary operations, which is possible since these operations are associative.

Fig. 3 Digraph in Example 3



As undirected co-graphs can be characterized by forbidding the  $P_4$ , directed co-graphs can be characterized likewise by excluding eight forbidden induced subdigraphs (Crespelle and Paul 2006). For every directed co-graph we can define a tree structure denoted as *di-co-tree*. It is a binary ordered rooted tree whose vertices are labeled by the operations of the di-co-expression.

**Definition 2** (Di-co-tree) The *di-co-tree* for some directed co-graph  $G$  is recursively defined as follows.

- The *di-co-tree*  $T$  for di-co-expression  $v$  consists of a single vertex  $r$  (the *root* of  $T$ ) labeled by  $v$ .
- The *di-co-tree*  $T$  for di-co-expression  $G_1 \oplus G_2$  consists of a copy  $T_1$  of the di-co-tree for  $G_1$ , a copy  $T_2$  of the di-co-tree for  $G_2$ , an additional vertex  $r$  (the *root* of  $T$ ) labeled by  $\oplus$  and two additional arcs from vertex  $r$  to the roots of  $T_1$  and  $T_2$ . The root of  $T_1$  is the *first child* of  $r$  and the root of  $T_2$  is the *second child* of  $r$ .
- The *di-co-tree*  $T$  for di-co-expressions  $G_1 \otimes G_2$  and  $G_1 \oslash G_2$  are defined analogously to  $G_1 \oplus G_2$ .

For every directed co-graph one can construct a di-co-tree in linear time, see Crespelle and Paul (2006). Due to their recursive structure there are problems that are hard in general but which can be solved efficiently on directed co-graphs, see Bang-Jensen and Maddaloni (2014), Gurski (2017), Gurski et al. (2019a, b); Gurski et al (2020) and Gurski and Rehs (2018).

**Observation 4** Let  $G$  be a directed co-graph and  $T$  be a di-co-tree for  $G$ . For every vertex  $u$  of  $T$  which corresponds to a series operation, the subtree rooted at  $u$  defines a strongly connected subdigraph of  $G$ . Further, for every vertex  $u$  of  $T$  representing a series operation, such that no predecessor of  $u$  corresponds to a series operation, the leaves of the subtree rooted at  $u$  correspond to a strongly connected component of  $G$ .

By omitting the series composition within Definition 1 we obtain the class of all *oriented co-graphs*. The class of oriented co-graphs is denoted by  $\mathcal{O}$ .

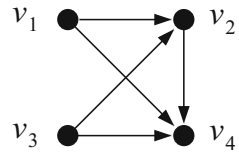
**Example 4** The di-co-expression

$$X = ((v_1 \oplus v_3) \oslash (v_2 \otimes v_4)) \tag{5}$$

defines  $\text{digraph}(X)$  shown in Fig. 4.

The class of oriented co-graphs has already been analyzed by Lawler in Lawler (1976) and Corneil et al. (1981, Section 5) using the notation of *transitive series-parallel (TSP) digraphs*. A digraph  $G = (V, E)$  is called *transitive* if for every pair

Fig. 4 Digraph in Example 4



$(u, v) \in E$  and  $(v, w) \in E$  of arcs with  $u \neq w$  the arc  $(u, w)$  also belongs to  $E$ . For oriented co-graphs the oriented chromatic number and also the graph isomorphism problem can be solved in linear time (Gurski et al. 2019b).

**Observation 5** *Every oriented co-graph is a directed co-graph and every oriented co-graph is a DAG.*

Since SSP corresponds to SSG and also to SSGW on a digraph without arcs, which is an oriented co-graph, we obtain the following result.

**Proposition 1** *SSG and SSGW are NP-hard on oriented co-graphs.*

Next, we will show pseudo-polynomial solutions for SSG and SSGW restricted to directed co-graphs. The main idea is a dynamic programming along the recursive structure of a given directed co-graph.

### 3.2 Subset sum with digraph constraint (SSG)

By Lemma 4 in order to solve SSG we can restrict ourselves to directed acyclic graphs. This can be done by replacing every strongly connected component  $S$  by a new vertex  $x_S$  whose size is the sum of the sizes of the vertices in  $S$ . In order to identify the strongly connected components of directed co-graphs using a di-co-tree we apply Observation 4. We perform a breadth first search on a di-co-tree  $T$  starting at the root and for every vertex  $u$  of  $T$  which corresponds to a series operation we substitute the subtree rooted at  $u$  by a single vertex whose size is the sum of the sizes of the vertices corresponding to the leaves of the subtree rooted at  $u$ . This does not reduce the size of the digraph or its di-co-tree in general, e.g. for oriented co-graphs we have no non-trivial strongly connected component.

We consider an instance of SSG such that  $G = (A, E)$  is a directed co-graph which is given by some di-co-expression  $X$ . For some subexpression  $X'$  of  $X$  let  $F(X', s) = 1$  if there is a solution  $A'$  in the graph defined by  $X'$  satisfying (1) and (2) such that  $s(A') = s$ , otherwise let  $F(X', s) = 0$ . We use the notation  $s(X') = \sum_{a_j \in X'} s_j$ .

**Lemma 10** *Let  $0 \leq s \leq c$ .*

1.  $F(a_j, s) = 1$  if and only if  $s = 0$  or  $s_j = s$ .  
In all other cases  $F(a_j, s) = 0$ .
2.  $F(X_1 \oplus X_2, s) = 1$ , if and only if there are some  $0 \leq s' \leq s$  and  $0 \leq s'' \leq s$  such that  $s' + s'' = s$  and  $F(X_1, s') = 1$  and  $F(X_2, s'') = 1$ .  
In all other cases  $F(X_1 \oplus X_2, s) = 0$ .
3.  $F(X_1 \otimes X_2, s) = 1$ , if and only if

- $F(X_2, s) = 1$  for  $0 \leq s \leq s(X_2)$ <sup>4</sup> or
- there is an  $s' > 0$ , such that  $s = s' + s(X_2)$  and  $F(X_1, s') = 1$ .

In all other cases  $F(X_1 \oslash X_2, s) = 0$ .

4.  $F(X_1 \otimes X_2, s) = 1$ , if and only if  $s = 0$  or  $s = s(X_1) + s(X_2)$ .  
In all other cases  $F(X_1 \otimes X_2, s) = 0$ .

**Proof** We show the correctness of the stated equivalences. Let  $0 \leq s \leq c$ .

1. The only possible solutions in  $\text{digraph}(a_j)$  are  $\emptyset$  and  $\{a_j\}$  which have size 0 and  $s_j$ , respectively.
2. If  $F(X_1 \oplus X_2, s) = 1$ , then by Lemma 6 there are  $s'$  and  $s''$  such that  $s' + s'' = s$  and solutions in  $\text{digraph}(X_1)$  and in  $\text{digraph}(X_2)$  which guarantee  $F(X_1, s') = 1$  and  $F(X_2, s'') = 1$ .  
Further, for every  $s'$  and  $s''$ , such that  $s' + s'' = s$ ,  $F(X_1, s') = 1$ , and  $F(X_2, s'') = 1$ , it holds that  $F(X_1 \oplus X_2, s) = 1$  since the operation (disjoint union) does not create new edges.

3. If  $F(X_1 \oslash X_2, s) = 1$ , then we distinguish two cases. If the solution of size  $s$  in  $\text{digraph}(X_1 \oslash X_2)$  contains no vertices of  $\text{digraph}(X_1)$ , then by Lemma 6 there is a solution in  $\text{digraph}(X_2)$  which guarantees  $F(X_2, s) = 1$ .  
If the solution  $A'$  of size  $s$  in  $\text{digraph}(X_1 \oslash X_2)$  contains at least one vertex of  $\text{digraph}(X_1)$ , then by (2) solution  $A'$  has to contain all vertices of  $\text{digraph}(X_2)$  and by Lemma 6 there is a solution in  $\text{digraph}(X_1)$  which guarantees  $F(X_1, s - s(X_2)) = 1$ .

Further, for every  $0 \leq s \leq s(X_2)$  where  $F(X_2, s) = 1$  we have  $F(X_1 \oslash X_2, s) = 1$  since the solutions from  $\text{digraph}(X_2)$  do not contain any predecessors of vertices from  $\text{digraph}(X_1)$  in  $\text{digraph}(X_1 \oslash X_2)$ .

Also for every  $1 \leq s' \leq s(X_1)$  where  $F(X_1, s') = 1$  for  $s = s' + s(X_2)$  we have  $F(X_1 \oslash X_2, s) = 1$  since every solution in  $\text{digraph}(X_1)$  has to be extended by  $X_2$  since at least one predecessor of  $\text{digraph}(X_2)$  is part of the solution and thus, all vertices of  $\text{digraph}(X_2)$  have to belong to the solution.

4. If  $F(X_1 \otimes X_2, s) = 1$ , then we distinguish two cases. If the solution of size  $s$  is empty, then  $s = 0$ .  
Otherwise,  $s = s(X_1) + s(X_2)$  since  $\text{digraph}(X_1 \otimes X_2)$  is strongly connected and thus, all vertices of  $\text{digraph}(X_1)$  and all vertices of  $\text{digraph}(X_2)$  have to be part of the solution.  
Further, if  $s = 0$  or  $s = s(X_1) + s(X_2)$ , it holds that  $F(X_1 \otimes X_2, s) = 1$  since the empty and the complete vertex set both satisfy (2).

□

**Corollary 1** *There is a solution with sum  $s$  for an instance of SSG such that  $G$  is a directed co-graph which is given by some di-co-expression  $X$  if and only if  $F(X, s) = 1$ . Therefore,  $OPT(I) = \max\{s \mid F(X, s) = 1\}$ .*

**Theorem 2** *SSG can be solved in directed co-graphs with  $n$  vertices and  $m$  arcs in  $\mathcal{O}(n \cdot c^2 + m)$  time and  $\mathcal{O}(n \cdot c)$  space.*

<sup>4</sup> The value  $s = 0$  is for choosing an empty solution in  $\text{digraph}(X_1 \oslash X_2)$ .

**Proof** Let  $G = (A, E)$  be a directed co-graph and  $T$  be a di-co-tree for  $G$  with root  $r$ . For some vertex  $u$  of  $T$  we denote by  $T_u$  the subtree rooted at  $u$  and  $X_u$  the co-expression defined by  $T_u$ . In order to solve the SSG problem for an instance  $I$  on graph  $G$ , we traverse di-co-tree  $T$  in a bottom-up order. For every vertex  $u$  of  $T$  and  $0 \leq s \leq c$  we compute  $F(X_u, s)$  following the rules given in Lemma 10. By Corollary 1 we can solve our problem by  $F(X_r, s) = F(X, s)$ .

A di-co-tree  $T$  can be computed in  $\mathcal{O}(n + m)$  time from a directed co-graph with  $n$  vertices and  $m$  arcs, see Crespelle and Paul (2006). All  $s(X_i)$  can be precomputed in  $\mathcal{O}(n)$  time. Our rules given in Lemma 10 show the following running times.

- For every  $a_j \in A$  and every  $0 \leq s \leq c$  value  $F(a_j, s)$  is computable in  $\mathcal{O}(1)$  time.
- For every  $0 \leq s \leq c$ , every  $F(X_1 \oplus X_2, s)$  and every  $F(X_1 \otimes X_2, s)$  can be computed in  $\mathcal{O}(c)$  time from  $F(X_1, s')$  and  $F(X_2, s'')$ .
- For every  $0 \leq s \leq c$ , every  $F(X_1 \otimes X_2, s)$  can be computed in  $\mathcal{O}(1)$  time from  $s(X_1)$  and  $s(X_2)$ .

Since we have  $n$  leaves and  $n - 1$  inner vertices in  $T$ , the running time is in  $\mathcal{O}(nc^2 + m)$ .  $\square$

A *tournament* is a digraph  $G = (A, E)$  where for each two different vertices  $u, v \in A$  it holds that exactly one of the two pairs  $(u, v)$  and  $(v, u)$  belongs to  $E$ . The class of transitive tournaments is denoted by TT. Transitive tournaments are characterized in several ways, see Gould (2012, Chapter 9).

**Lemma 11** (Gould 2012) *For every digraph  $G$  the following statements are equivalent.*

1.  $G$  is a transitive tournament.
2.  $G$  is an acyclic tournament.
3.  $G$  is a tournament with exactly one Hamiltonian path.
4.  $G$  is a tournament and every vertex in  $G$  has a different outdegree, i.e.  $\{\text{outdegree}(v) \mid v \in V\} = \{0, \dots, |V| - 1\}$ .
5.  $G$  can be constructed from the one-vertex graph by repeatedly adding an out-dominating vertex.
6.  $G$  can be constructed from the one-vertex graph repeatedly adding an in-dominated vertex.

In Gourvès et al. (2018, Lemma 4) it is shown that SSG is polynomial on acyclic tournaments without stating a running time. Since acyclic tournaments, and equivalently transitive tournaments, are a subclass of oriented co-graphs, we reconsider the following result.

**Remark 2** Every transitive tournament  $G$  can be defined from a single vertex graph  $v_1$  by repeatedly adding a vertex of maximum indegree and outdegree 0, i.e. an in-dominated vertex  $v_2, \dots, v_n$  (cf. Lemma 11). This order can be defined in  $\mathcal{O}(n^2)$  time from  $G$ . The feasible solutions w.r.t. the digraph constraint (2) are  $\emptyset$  and for  $1 \leq k \leq n$  the set  $\{v_i \mid k \leq i \leq n\}$ . This leads to at most  $n + 1$  possible solutions for SSG for which we have to check the capacity constraint (1) and among those satisfying (1) we select one set with largest sum of sizes. Thus, SSG is solvable in  $\mathcal{O}(n^2)$  time on transitive tournaments with  $n$  vertices.

A *bioriented clique* is a digraph  $G = (A, E)$  where for each two different vertices  $u, v \in A$  it holds that both of the two pairs  $(u, v)$  and  $(v, u)$  belong to  $E$ . The class of bioriented cliques is denoted by BC.

**Remark 3** Within a bioriented clique  $G = (A, E)$  the whole vertex set is a strongly connected component. By Lemma 2 the only possible solutions are  $A$  and  $\emptyset$ . Thus, SSG is solvable in  $\mathcal{O}(n)$  time on bioriented cliques with  $n$  vertices.

### 3.3 Subset sum with weak digraph constraint (SSGW)

Next, we consider SSGW on directed co-graphs. In order to get useful informations about the sources within a solution, we use an extended data structure. We consider an instance of SSGW such that  $G = (A, E)$  is a directed co-graph which is given by some di-co-expression  $X$ . For some subexpression  $X'$  of  $X$  let  $H(X', s, s') = 1$  if there is a solution  $A'$  in the graph defined by  $X'$  satisfying (1) and (3) such that  $s(A') = s$  and the sum of sizes of the sources in  $A'$  is  $s'$ , otherwise let  $H(X', s, s') = 0$ . We denote by  $o(X)$  the sum of the sizes of all sources in  $\text{digraph}(X)$ .

**Remark 4** A remarkable difference between SSGW and SSG w.r.t. co-graph operations is the following. When considering  $X_1 \otimes X_2$  we can combine solutions  $A_1$  of  $X_1$  satisfying (1) and (3) which do not contain all items of  $X_1$  with solutions  $A_2$  of  $X_2$  satisfying only (1) to obtain solution  $A_1 \cup A_2$  of  $X_1 \otimes X_2$  satisfying (1) and (3), if  $s(A_1) + s(A_2) \leq c$ . Furthermore, within  $X_1 \otimes X_2$  we can combine solutions  $A_1$  of  $X_1$  satisfying (1) which do not contain all items and solutions  $A_2$  of  $X_2$  satisfying (1) which do not contain all items to obtain solution  $A_1 \cup A_2$  of  $X_1 \otimes X_2$  satisfying (1) and (3), if  $s(A_1) + s(A_2) \leq c$ .

Thus, in order to solve SSGW on a directed co-graph  $G$ , we use solutions for SSP on subexpressions for  $G$ . We consider an instance of SSP such that  $G = (A, E)$  is a directed co-graph which is given by some di-co-expression  $X$ . For a subexpression  $X'$  of  $X$  let  $H'(X', s) = 1$  if there is a solution  $A'$  in the digraph defined by  $X'$  satisfying (1) such that  $s(A') = s$ , otherwise let  $H'(X', s) = 0$ .

**Lemma 12** Let  $0 \leq s \leq c$ .

1.  $H'(a_j, s) = 1$  if and only if  $s = 0$  or  $s = s_j$ .  
In all other cases  $H'(a_j, s) = 0$ .
2.  $H'(X_1 \oplus X_2, s) = 1$ , if and only if there are some  $0 \leq s' \leq s$  and  $0 \leq s'' \leq s$  such that  $s' + s'' = s$  and  $H'(X_1, s') = 1$  and  $H'(X_2, s'') = 1$ .  
In all other cases  $H'(X_1 \oplus X_2, s) = 0$ .
3.  $H'(X_1 \otimes X_2, s) = H'(X_1 \oplus X_2, s)$
4.  $H'(X_1 \otimes X_2, s) = H'(X_1 \oplus X_2, s)$

**Proof** We show the correctness of the stated equivalences. Let  $0 \leq s \leq c$ .

1. The only possible solutions in  $\text{digraph}(a_j)$  are  $\emptyset$  and  $\{a_j\}$  which have size 0 and  $s_j$ , respectively.



2. If  $H'(X_1 \oplus X_2, s) = 1$ , then by Observation 3 there are  $s'$  and  $s''$  such that  $s' + s'' = s$  and solutions in  $\text{digraph}(X_1)$  and in  $\text{digraph}(X_2)$  which guarantee  $H'(X_1, s') = 1$  and  $H'(X_2, s'') = 1$ .  
Further, for every  $s'$  and  $s''$ , such that  $s' + s'' = s$ ,  $H'(X_1, s') = 1$ , and  $H'(X_2, s'') = 1$ , we can combine these two solutions into one solution of size  $s$  in  $\text{digraph}(X_1 \oplus X_2)$ . Thus, it holds that  $H'(X_1 \oplus X_2, s) = 1$ .
3. Since the arcs are irrelevant for the capacity constraint (1), it holds that  $H'(X_1 \otimes X_2, s) = H'(X_1 \oplus X_2, s)$ .
4. Since the arcs are irrelevant for the capacity constraint (1), it holds that  $H'(X_1 \otimes X_2, s) = H'(X_1 \oplus X_2, s)$ .

□

This allows us to compute the values  $H(X', s, s')$  as follows.

**Lemma 13** *Let  $0 \leq s, s' \leq c$ .*

1.  $H(a_j, s, s') = 1$  if and only if  $s = s' = 0$  or  $s_j = s = s'$ .  
*In all other cases  $H(a_j, s, s') = 0$ .*
2.  $H(X_1 \oplus X_2, s, s') = 1$ , if and only if there are  $0 \leq s_1 \leq s, 0 \leq s_2 \leq s, 0 \leq s'_1 \leq s', 0 \leq s'_2 \leq s'$ , such that  $s_1 + s_2 = s, s'_1 + s'_2 = s', H(X_1, s_1, s'_1) = 1$ , and  $H(X_2, s_2, s'_2) = 1$ .  
*In all other cases  $H(X_1 \oplus X_2, s, s') = 0$ .*
3.  $H(X_1 \otimes X_2, s, s') = 1$ , if and only if
  - $H(X_1, s, s') = 1$  for  $1 \leq s < s(X_1)$  or
  - $H'(X_2, s) = 1$  for  $0 \leq s \leq s(X_2)$ <sup>5</sup> and  $s' = 0$  or
  - there are  $1 \leq s_2 \leq s(X_2)$ , such that  $s(X_1) + s_2 = s, o(X_1) = s'$ , and  $H(X_2, s_2, o(X_2)) = 1$ , or
  - $s = s(X_1) + s(X_2)$  and  $s' = o(X_1)$ , or
  - there are  $0 \leq s_1 < s(X_1), 0 \leq s_2 \leq s(X_2)$ , such that  $s_1 + s_2 = s, H(X_1, s_1, s') = 1$ , and  $H'(X_2, s_2) = 1$ .

*In all other cases  $H(X_1 \otimes X_2, s, s') = 0$ .*

4.  $H(X_1 \otimes X_2, s, 0) = 1$ , if and only if
  - $H'(X_1, s) = 1$  for  $1 \leq s < s(X_1)$  or
  - $H'(X_2, s) = 1$  for  $0 \leq s < s(X_2)$ <sup>6</sup> or
  - there are  $1 \leq s_2 \leq s(X_2)$ , such that  $s(X_1) + s_2 = s$ , and  $H(X_2, s_2, o(X_2)) = 1$ , or
  - there are  $1 \leq s_1 \leq s(X_1)$ , such that  $s_1 + s(X_2) = s$ , and  $H(X_1, s_1, o(X_1)) = 1$ , or
  - $s = s(X_1) + s(X_2)$ , or
  - there exist  $1 \leq s_1 < s(X_1)$  and  $1 \leq s_2 < s(X_2)$  such that  $s_1 + s_2 = s, H'(X_1, s_1) = 1$ , and  $H'(X_2, s_2) = 1$ .

*In all other cases  $H(X_1 \otimes X_2, s, s') = 0$ .*

<sup>5</sup> The value  $s = 0$  is for choosing an empty solution in  $\text{digraph}(X_1 \otimes X_2)$ .

<sup>6</sup> The value  $s = 0$  is for choosing an empty solution in  $\text{digraph}(X_1 \otimes X_2)$ .

**Proof** We show the correctness of the stated equivalences. Let  $0 \leq s, s' \leq c$ .

1. The only possible solutions in  $\text{digraph}(a_j)$  are  $\emptyset$  and  $\{a_j\}$  which have size 0 and  $s_j$ , respectively. Further, a single vertex is a source.
2. If  $H(X_1 \oplus X_2, s, s') = 1$ , then by Lemma 7 there are  $s_1, s_2$  and  $s'_1, s'_2$  such that  $s_1 + s_2 = s, s'_1 + s'_2 = s'$  and solutions in  $\text{digraph}(X_1)$  and in  $\text{digraph}(X_2)$  which guarantee  $H(X_1, s_1, s'_1) = 1$  and  $H(X_2, s_2, s'_2) = 1$ .  
Further, for every  $0 \leq s_1 \leq s, 0 \leq s_2 \leq s, 0 \leq s'_1 \leq s', 0 \leq s'_2 \leq s'$ , such that  $s_1 + s_2 = s, s'_1 + s'_2 = s', H(X_1, s_1, s'_1) = 1$ , and  $H(X_2, s_2, s'_2) = 1$ , it holds that  $H(X_1 \oplus X_2, s, s') = 1$  since the operation (disjoint union) does not create new edges.
3. If  $H(X_1 \otimes X_2, s, s') = 1$ , then we distinguish the following cases. If the solution of size  $s$  in  $\text{digraph}(X_1 \otimes X_2)$  is a non-empty proper subset of the vertices of  $\text{digraph}(X_1)$ , then by Lemma 8 there is a solution in  $\text{digraph}(X_1)$  which guarantees  $H(X_1, s, s') = 1$ .

Next, assume that the solution  $A'$  of size  $s$  in  $\text{digraph}(X_1 \otimes X_2)$  contains no vertices of  $\text{digraph}(X_1)$ . Since every solution satisfying constraints (1) and (3) is also a solution which satisfies only (1), we have  $H'(X_1 \otimes X_2, s) = 1$ . And since  $A'$  contains no vertices of  $\text{digraph}(X_1)$ , Observation 3 implies that there is a solution in  $\text{digraph}(X_2)$  which guarantees  $H'(X_2, s) = 1$ .

If the solution  $A'$  of size  $s$  in  $\text{digraph}(X_1 \otimes X_2)$  contains all vertices of  $\text{digraph}(X_1)$ , then the order composition and the weak digraph constraint (3) imply that the set  $A'$  can be extended by every solution of  $\text{digraph}(X_2)$  which includes all sources of  $\text{digraph}(X_2)$ . Thus, by Lemma 9, there is a solution in  $\text{digraph}(X_2)$ , which guarantees  $H(X_2, s - s(X_1), o(X_2)) = 1$ .

Further, if the solution  $A'$  of size  $s$  in  $\text{digraph}(X_1 \otimes X_2)$  contains all vertices of  $\text{digraph}(X_1)$ , it is also possible to extend  $A'$  by all vertices of  $\text{digraph}(X_2)$  and thus  $s = s(X_1) + s(X_2)$ .

Finally, if the solution  $A'$  of size  $s$  in  $\text{digraph}(X_1 \otimes X_2)$  contains some but not all vertices of  $\text{digraph}(X_1)$  and possibly vertices of  $\text{digraph}(X_2)$ , then by Lemma 8 and Observation 3 there are  $s_1$  and  $s_2$  such that  $s_1 + s_2 = s$  and solutions in  $\text{digraph}(X_1)$  and in  $\text{digraph}(X_2)$  which guarantee  $H(X_1, s_1, s') = 1$  and  $H'(X_2, s_2) = 1$ .

The solutions of size  $1 \leq s < s(X_1)$  from  $\text{digraph}(X_1)$  remain feasible for  $\text{digraph}(X_1 \otimes X_2)$ .

Every subset  $A'$  of size  $0 \leq s \leq s(X_2)$  from  $\text{digraph}(X_2)$  which satisfies (1) leads to a solution  $A'$  of size  $s$  satisfying (1) and (3) in  $\text{digraph}(X_1 \otimes X_2)$  since every vertex of  $\text{digraph}(X_2)$  gets a predecessor in  $\text{digraph}(X_1)$ , which is not in  $A'$ .

Further, the set of all vertices of  $\text{digraph}(X_1)$  extended by every solution of  $\text{digraph}(X_2)$  of size  $s_2$  which includes all sources of  $\text{digraph}(X_2)$  leads to a feasible solution for  $\text{digraph}(X_1 \otimes X_2)$  of size  $s(X_1) + s_2$ . The size of the sources has to be updated to  $o(X_1)$ , since the sources of  $\text{digraph}(X_1)$  are the sources of  $\text{digraph}(X_1 \otimes X_2)$ .

Moreover, the complete vertex set of  $\text{digraph}(X_1 \otimes X_2)$  is obviously a feasible SSGW solution if it fulfills the capacity constraint.

Furthermore, by Remark 4 we can combine SSGW solutions of size  $s_1 < s(X_1)$  of digraph( $X_1$ ) and SSP solutions of size  $s_2$  of digraph( $X_2$ ) to a SSGW solution of size  $s_1 + s_2$  of digraph( $X_1 \otimes X_2$ ).

4. First, we want to mention that  $H(X_1 \otimes X_2, s, s') = 1$  is only possible for  $s' = 0$ , since digraph( $X_1 \otimes X_2$ ) has no sources. If  $H(X_1 \otimes X_2, s, 0) = 1$ , we distinguish the following cases. Assume the solution of size  $s$  in digraph( $X_1 \otimes X_2$ ) is a proper and non-empty subset of the vertices of digraph( $X_1$ ). Since  $H(X_1 \otimes X_2, s, 0) = 1$ , it holds that  $H'(X_1 \otimes X_2, s) = 1$ . And since  $A'$  contains only vertices of digraph( $X_1$ ), Observation 3 implies that there is a solution in digraph( $X_1$ ) which guarantees  $H'(X_1, s) = 1$ .

If the solution of size  $s$  in digraph( $X_1 \otimes X_2$ ) is a proper subset of the vertices of digraph( $X_2$ ), then by the same arguments as for digraph( $X_1$ ) there is a solution in digraph( $X_2$ ) which guarantees  $H'(X_2, s) = 1$ .

If the solution  $A'$  of size  $s$  in digraph( $X_1 \otimes X_2$ ) contains all vertices of digraph( $X_1$ ), then the series composition and the weak digraph constraint (3) imply that the set  $A'$  can be extended by all solutions of digraph( $X_2$ ) which include all sources of digraph( $X_2$ ). Thus, by Lemma 9, there is a solution in digraph( $X_2$ ), which guarantees  $H(X_2, s - s(X_1), o(X_2))$ .

If the solution  $A'$  of size  $s$  in digraph( $X_1 \otimes X_2$ ) contains all vertices of digraph( $X_2$ ), then by the same arguments as for digraph( $X_1$ ) there is a solution in digraph( $X_1$ ), which guarantees  $H(X_1, s - s(X_2), o(X_1))$ .

If the solution  $A'$  of size  $s$  in digraph( $X_1 \otimes X_2$ ) contains all vertices of digraph( $X_1$ ) or all vertices of digraph( $X_2$ ), then by (3) solution  $A'$  can be extended by all vertices of digraph( $X_2$ ) or all vertices of digraph( $X_1$ ), respectively, and thus  $s = s(X_1) + s(X_2)$ .

Finally, if the solution  $A'$  of size  $s$  in digraph( $X_1 \otimes X_2$ ) contains some but not all vertices of digraph( $X_1$ ) and some but not all vertices of digraph( $X_2$ ), then by Observation 3 there are  $s_1$  and  $s_2$  such that  $s_1 + s_2 = s$  and solutions in digraph( $X_1$ ) and in digraph( $X_2$ ) which guarantee  $H'(X_1, s_1) = 1$  and  $H'(X_2, s_2) = 1$ .

Every subset  $A'$  of size  $1 \leq s < s(X_1)$  from digraph( $X_1$ ) which satisfies (1) leads to a solution  $A'$  of size  $s$  satisfying constraints (1) and (3) in digraph( $X_1 \otimes X_2$ ) since every vertex of digraph( $X_1$ ) gets a predecessor in digraph( $X_2$ ), which is not in  $A'$ .

In the same way every subset  $A'$  of size  $0 \leq s < s(X_2)$  from digraph( $X_2$ ) which satisfies (1) leads to a solution  $A'$  of size  $s$  satisfying (1) and (3) in digraph( $X_1 \otimes X_2$ ) since every vertex of digraph( $X_2$ ) gets a predecessor in digraph( $X_1$ ), which is not in  $A'$ .

Further, all the set of all vertices of digraph( $X_1$ ) extended by every solution of digraph( $X_2$ ) of size  $s_2$  which includes all sources of digraph( $X_2$ ) leads to a feasible solution for digraph( $X_1 \otimes X_2$ ) of size  $s(X_1) + s_2$  and the set of all vertices of digraph( $X_2$ ) extended by every solution of digraph( $X_1$ ) of size  $s_1$  which includes all sources of digraph( $X_1$ ) leads to a feasible solution for digraph( $X_1 \otimes X_2$ ) of size  $s_1 + s(X_2)$ .

Moreover, the complete vertex set of digraph( $X_1 \otimes X_2$ ) is obviously a feasible SSGW solution.

Furthermore, by Remark 4, we can combine SSP solutions of size  $s_1 < s(X_1)$  and SSP solutions of size  $s_2 < s(X_2)$  to a SSGW solution of size  $s_1 + s_2$ .

□

In order to solve the SSGW problem we traverse di-co-tree  $T$  in a bottom-up order and perform the following computations depending on the type of operation.

**Corollary 2** *There is a solution with sum  $s$  for some instance of SSGW such that  $G$  is a directed co-graph which is given by some di-co-expression  $X$  if and only if  $H(X, s, s') = 1$ . Therefore,  $OPT(I) = \max\{s \mid H(X, s, s') = 1\}$ .*

The next result can be obtained by similar arguments as given within the proof of Theorem 2.

**Theorem 3** *SSGW can be solved in directed co-graphs with  $n$  vertices and  $m$  arcs in  $\mathcal{O}(n \cdot c^4 + m)$  time and  $\mathcal{O}(n \cdot c^2)$  space.*

## 4 SSG and SSGW on series-parallel digraphs

### 4.1 Series-parallel digraphs

We recall the definitions from Bang-Jensen and Gutin (2018) which are based on Valdes et al. (1982). First, we introduce two operations for two vertex-disjoint digraphs  $G_1 = (V_1, E_1)$  and  $G_2 = (V_2, E_2)$ . Let  $O_1$  be the set of vertices of outdegree 0 (set of sinks) in  $G_1$  and  $I_2$  be the set of vertices of indegree 0 (set of sources) in  $G_2$ .

- The *parallel composition* of  $G_1$  and  $G_2$ , denoted by  $G_1 \cup G_2$ , is the digraph with vertex set  $V_1 \cup V_2$  and arc set  $E_1 \cup E_2$ .
- The *series composition* of  $G_1$  and  $G_2$ , denoted by  $G_1 \times G_2$  is the digraph with vertex set  $V_1 \cup V_2$  and arc set  $E_1 \cup E_2 \cup (O_1 \times I_2)$ .

**Definition 3** (Minimal series-parallel digraphs) The class of *minimal series-parallel digraphs*, *msp-digraphs* for short, is recursively defined as follows.

1. Every digraph on a single vertex  $(\{v\}, \emptyset)$ , denoted by  $v$ , is a *minimal series-parallel digraph*.
2. If  $G_1$  and  $G_2$  are vertex-disjoint minimal series-parallel digraphs, then
  - (a) the parallel composition  $G_1 \cup G_2$  and
  - (b) then series composition  $G_1 \times G_2$  are *minimal series-parallel digraphs*.

The class of minimal series-parallel digraphs is denoted by MSP.

Every expression  $X$  using these three operations is called an *msp-expression* and  $\text{digraph}(X)$  the defined digraph.

**Example 5** 1. The msp-expression

$$X = ((v_1 \cup v_2) \times (v_3 \cup v_4)) \quad (6)$$

defines  $\text{digraph}(X)$  shown in Fig. 5.

Fig. 5 Digraph in Example 5(1.)

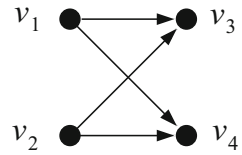
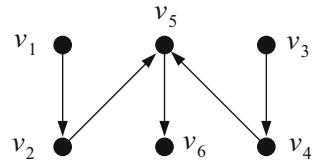


Fig. 6 Digraph in Example 5(2.)



2. The msp-expression

$$X = (((v_1 \times v_2) \cup (v_3 \times v_4)) \times (v_5 \times v_6)) \tag{7}$$

defines digraph(X) shown in Fig. 6.

For every minimal series-parallel digraph we can define a tree structure, denoted as *msp-tree*.<sup>7</sup> The leaves of the msp-tree represent the vertices of the graph and the inner vertices of the msp-tree correspond to the operations applied on the subexpressions defined by the subtrees. For every minimal series-parallel digraph one can construct a msp-tree in linear time, see Valdes et al. (1982).

**Observation 6** Every in- or out-rooted tree is a minimal series-parallel digraph.

**Lemma 14** Let  $G = (V, E)$  be a minimal series-parallel digraph. Then, for every vertex  $x \in V$  there is a sink  $x_s$  of  $G$ , such that there is a directed path from  $x$  to  $x_s$  in  $G$  and there is a source  $x_o$  of  $G$ , such that there is a path from  $x_o$  to  $x$  in  $G$ .

**Proof** Can be shown by induction on the recursive definition of minimal series-parallel digraphs. □

**Lemma 15** Let  $G = (V, E)$  be a minimal series-parallel digraph. Then, every non-empty feasible solution of SSG contains a sink of  $G$ .

**Proof** If a feasible solution  $A'$  contains some  $x \in V$ , then Lemma 14 implies that there is a sink  $x_s$  of  $G$ , such that there is a path from  $x$  to  $x_s$  in  $G$  which implies by (2) that  $x_s \in A'$ . □

**Definition 4** (Series-parallel digraphs) Series-parallel digraphs are exactly the digraphs whose transitive closure equals the transitive closure of some minimal series-parallel digraph.

The class of series-parallel digraphs is denoted by SPD.

<sup>7</sup> In Valdes et al. (1982) the tree-structure for an msp-digraphs is denoted as binary decomposition tree.

**Theorem 4** (Valdes et al. 1982) *An acyclic digraph is series-parallel, if and only if its transitive closure is  $N$ -free, where  $N = (\{u, v, w, x\}, \{(v, w), (u, w), (u, x)\})$ .*

In order to define series-parallel partial order digraphs by series-parallel partial orders, we introduce two operations. Let  $(X_1, \leq)$  and  $(X_2, \leq)$  be two partially ordered sets over a set  $X$ , such that  $X_1 \subseteq X$ ,  $X_2 \subseteq X$ , and  $X_1 \cap X_2 = \emptyset$ .

- The *series composition* of  $(X_1, \leq)$  and  $(X_2, \leq)$  is the order with the following properties. If  $x$  and  $y$  are of the same set, then their order does not change. If  $x \in X_1$  and  $y \in X_2$ , then it holds that  $x \leq y$ .
- The *parallel composition* of  $(X_1, \leq)$  and  $(X_2, \leq)$  is the order with the following properties. Elements  $x$  and  $y$  are comparable if and only if they are both comparable in  $X_1$  or both comparable in  $X_2$  and they keep their corresponding order.

**Definition 5** (Series-parallel partial order) The class of *series-parallel partial orders* over a set  $X$  is recursively defined as follows.

1. Every single element  $(\{x\}, \emptyset)$ ,  $x \in X$ , is a *series-parallel partial order*.
2. If  $(X_1, \leq)$  and  $(X_2, \leq)$  are series-parallel partial orders over set  $X$ , such that  $X_1 \subseteq X$ ,  $X_2 \subseteq X$ , and  $X_1 \cap X_2 = \emptyset$ , then
  - (a) the series composition of  $(X_1, \leq)$  and  $(X_2, \leq)$  and
  - (b) the parallel composition of  $(X_1, \leq)$  and  $(X_2, \leq)$  are *series-parallel partial orders*.

**Example 6** The following partially ordered sets are series-parallel partial orders over set  $\{x_1, x_2, x_3, x_4\}$ .

- The parallel composition of  $(\{x_1\}, \emptyset)$  and  $(\{x_3\}, \emptyset)$  leads to the series-parallel partial order  $(\{x_1, x_3\}, \emptyset)$ .
- The series composition of  $(\{x_2\}, \emptyset)$  and  $(\{x_4\}, \emptyset)$  leads to the series-parallel partial order  $(\{x_2, x_4\}, \{(x_2, x_4)\})$ .
- The series composition of  $(\{x_1, x_3\}, \emptyset)$  and  $(\{x_2, x_4\}, \{(x_2, x_4)\})$  leads to the series-parallel partial order  $(\{x_1, x_2, x_3, x_4\}, \{(x_2, x_4), (x_1, x_2), (x_1, x_4), (x_3, x_2), (x_3, x_4)\})$ .

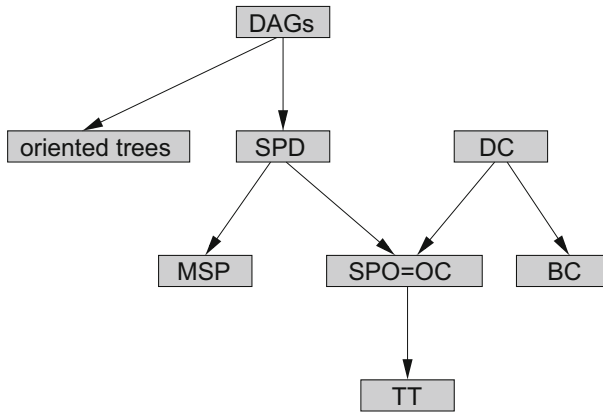
**Definition 6** (Series-parallel partial order digraphs) A *series-parallel partial order digraph*  $G = (V, E)$  is a digraph, where  $(V, \leq)$  is a series-parallel partial order and  $(x, y) \in E$  if and only if  $x \neq y$  and  $x \leq y$ .

The class of series-parallel partial order digraphs is denoted by SPO.

**Example 7** The series-parallel partial orders given in Example 6 show that the digraph shown in Fig. 4 is a series-parallel partial order digraph.

Comparing the definitions of the order composition of oriented co-graphs with the series composition of series-parallel partial order digraphs and the disjoint union composition of oriented co-graphs with the parallel composition of series-parallel partial order digraphs, see Examples 4 and 6, we obtain the following result.

**Observation 7** *The sets OC and SPO are equal.*



**Fig. 7** The figure shows the inclusions of special graph classes. A directed edge from class  $A$  to class  $B$  indicates that  $B \subseteq A$ . Two classes  $A$  and  $B$  are incomparable, if there is neither a directed path from  $A$  to  $B$ , nor a directed path from  $B$  to  $A$

In Fig. 7 we summarize the relation of directed co-graphs, series-parallel digraphs and related graph classes. The directed edges represent the existing relations between the graph classes, which follow by their definitions. For the relations to further graph classes we refer to Bang-Jensen and Gutin (2018, Figure 11.1).

Since SSP corresponds to SSG and also to SSGW on a digraph without arcs, which is a minimal series-parallel digraph, we obtain the following result.

**Proposition 2** *SSG and SSGW are NP-hard on minimal series-parallel digraph.*

Next, we will show pseudo-polynomial solutions for SSG and SSGW restricted to (minimal) series-parallel digraphs. The main idea is a dynamic programming along the recursive structure of a given (minimal) series-parallel digraph.

### 4.2 Subset sum with digraph constraint (SSG)

We consider an instance of SSG such that  $G = (A, E)$  is a minimal series-parallel digraph which is given by some msp-expression  $X$ . For some subexpression  $X'$  of  $X$  let  $F(X', s) = 1$  if there is a solution  $A'$  in the graph defined by  $X'$  satisfying (1) and (2) such that  $s(A') = s$ , otherwise let  $F(X', s) = 0$ . We use the notation  $s(X') = \sum_{a_j \in X'} s_j$ .

**Lemma 16** *Let  $0 \leq s \leq c$ .*

1.  $F(a_j, s) = 1$  if and only if  $s = 0$  or  $s_j = s$ .  
In all other cases  $F(a_j, s) = 0$ .
2.  $F(X_1 \cup X_2, s) = 1$ , if and only if there are some  $0 \leq s' \leq s$  and  $0 \leq s'' \leq s$  such that  $s' + s'' = s$  and  $F(X_1, s') = 1$  and  $F(X_2, s'') = 1$ .  
In all other cases  $F(X_1 \cup X_2, s) = 0$ .
3.  $F(X_1 \times X_2, s) = 1$ , if and only if

- $F(X_2, s) = 1$  for  $0 \leq s \leq s(X_2)$ <sup>8</sup> or
- there is some  $1 \leq s' \leq s(X_1)$  such that  $s = s' + s(X_2)$  and  $F(X_1, s') = 1$ .

In all other cases  $F(X_1 \times X_2, s) = 0$ .

**Proof** We show the correctness of the stated equivalences. Let  $0 \leq s \leq c$ .

1. The only possible solutions in  $\text{digraph}(a_j)$  are  $\emptyset$  and  $\{a_j\}$  which have size 0 and  $s_j$ , respectively.
2. If  $F(X_1 \cup X_2, s) = 1$ , then by Lemma 6 there are  $s'$  and  $s''$  such that  $s' + s'' = s$  and solutions in  $\text{digraph}(X_1)$  and in  $\text{digraph}(X_2)$  which guarantee  $F(X_1, s') = 1$  and  $F(X_2, s'') = 1$ .

Further, for every  $s'$  and  $s''$ , such that  $s' + s'' = s$ ,  $F(X_1, s') = 1$ , and  $F(X_2, s'') = 1$ , it holds that  $F(X_1 \cup X_2, s) = 1$  since the parallel composition creates no additional arcs.

3. If  $F(X_1 \times X_2, s) = 1$ , then we distinguish two cases. If the solution of size  $s$  in  $\text{digraph}(X_1 \times X_2)$  contains no vertex of  $\text{digraph}(X_1)$ , then by Lemma 6 there is a solution in  $\text{digraph}(X_2)$  which guarantees  $F(X_2, s) = 1$ .

Otherwise, the solution  $A'$  of size  $s$  in  $\text{digraph}(X_1 \times X_2)$  contains at least one vertex of  $\text{digraph}(X_1)$ . By the definition of the series composition and the digraph constraint (2) every solution from  $\text{digraph}(X_1)$  which contains a sink has to be extended by every vertex of  $X_2$  which is reachable by a source from  $\text{digraph}(X_2)$ . Since by Lemma 15 every non-empty feasible solution of SSG contains a sink, every solution from  $\text{digraph}(X_1)$  has to be extended by every vertex of  $X_2$  which is reachable by a source from  $\text{digraph}(X_2)$ . By Lemma 14 every solution from  $\text{digraph}(X_1)$  has to be extended by all vertices of  $\text{digraph}(X_2)$ . Thus, by Lemma 6 there is a solution in  $\text{digraph}(X_1)$  which guarantees  $F(X_1, s - s(X_2)) = 1$ .

Further, for every  $0 \leq s \leq s(X_2)$  where  $F(X_2, s) = 1$  we have  $F(X_1 \times X_2, s) = 1$  since the solutions from  $\text{digraph}(X_2)$  do not contain any predecessors of vertices from  $\text{digraph}(X_1)$  in  $\text{digraph}(X_1 \times X_2)$ .

For every  $1 \leq s' \leq s(X_1)$  where  $F(X_1, s') = 1$  the definition of the series composition and the digraph constraint (2) imply that for  $s = s' + s(X_2)$  it holds that  $F(X_1 \times X_2, s) = 1$  for reasons given above. □

**Corollary 3** *There is a solution with sum  $s$  for some instance of SSG such that  $G$  is a minimal series-parallel digraph which is given by some msp-expression  $X$  if and only if  $F(X, s) = 1$ . Therefore,  $OPT(I) = \max\{s \mid F(X, s) = 1\}$ .*

**Theorem 5** *SSG can be solved in minimal series-parallel digraphs with  $n$  vertices and  $m$  arcs in  $\mathcal{O}(n \cdot c^2 + m)$  time and  $\mathcal{O}(n \cdot c)$  space.*

**Proof** Let  $G = (V, E)$  be a minimal series-parallel digraph and  $T$  be an msp-tree for  $G$  with root  $r$ . For some vertex  $u$  of  $T$  we denote by  $T_u$  the subtree rooted at  $u$  and  $X_u$  the msp-expression defined by  $T_u$ . In order to solve the SSG problem for an instance  $I$  graph  $G$ , we traverse msp-tree  $T$  in a bottom-up order. For every vertex  $u$  of  $T$  and  $0 \leq s \leq c$  we compute  $F(X_u, s)$  following the rules given in Lemma 16. By Corollary 3 we can solve our problem by  $F(X_r, s) = F(X, s)$ .

<sup>8</sup> The value  $s = 0$  is for choosing an empty solution in  $\text{digraph}(X_1 \times X_2)$ .



An msp-tree  $T$  can be computed in  $\mathcal{O}(n + m)$  time from a minimal series-parallel digraph with  $n$  vertices and  $m$  arcs, see Valdes et al. (1982). All  $s(X_i)$  can be pre-computed in  $\mathcal{O}(n)$  time. Our rules given in Lemma 16 show the following running times.

- For every  $a_j \in V$  and every  $0 \leq s \leq c$  value  $F(a_j, s)$  is computable in  $\mathcal{O}(1)$  time.
- For every  $0 \leq s \leq c$ , every  $F(X_1 \cup X_2, s)$  can be computed in  $\mathcal{O}(c)$  time from  $F(X_1, s')$  and  $F(X_2, s'')$ .
- For every  $0 \leq s \leq c$ , every  $F(X_1 \times X_2, s)$  can be computed in  $\mathcal{O}(1)$  time from  $F(X_1, s')$ ,  $F(X_2, s'')$ , and  $s(X_2)$ .

Since we have  $n$  leaves and  $n - 1$  inner vertices in  $T$ , the running time is in  $\mathcal{O}(nc^2 + m)$ . □

**Theorem 6** *SSG can be solved in series-parallel digraphs with  $n$  vertices and  $m$  arcs in  $\mathcal{O}(n \cdot c^2 + n^{2.3729})$  time and  $\mathcal{O}(n \cdot c)$  space.*

**Proof** Let  $G$  be some series-parallel digraph. By Lemma 5 we can use the transitive reduction of  $G$ , which can be computed in  $\mathcal{O}(n^{2.3729})$  time by Le Gall (2014). □

### 4.3 Subset sum with weak digraph constraint (SSGW)

Next, we consider SSGW on minimal series-parallel digraph. In order to get useful informations about the sinks within a solution, we use an extended data structure. We consider an instance of SSGW such that  $G = (A, E)$  is a minimal series-parallel digraph which is given by some msp-expression  $X$ . For some subexpression  $X'$  of  $X$  let  $H(X', s, s') = 1$  if there is a solution  $A'$  in the graph defined by  $X'$  satisfying (1) and (3) such that  $s(A') = s$  and the sum of sizes of the sinks in  $A'$  is  $s'$ , otherwise let  $H(X', s, s') = 0$ . We denote by  $i(X)$  the sum of the sizes of all sinks in digraph( $X$ ).

**Lemma 17** *Let  $0 \leq s, s' \leq c$ .*

1.  $H(a_j, s, s') = 1$  if and only if  $s = s' = 0$  or  $s_j = s = s'$ .  
*In all other cases  $H(a_j, s, s') = 0$ .*
2.  $H(X_1 \cup X_2, s, s') = 1$ , if and only if there are  $0 \leq s_1 \leq s, 0 \leq s_2 \leq s, 0 \leq s'_1 \leq s', 0 \leq s'_2 \leq s'$ , such that  $s_1 + s_2 = s, s'_1 + s'_2 = s', H(X_1, s_1, s'_1) = 1$ , and  $H(X_2, s_2, s'_2) = 1$ .  
*In all other cases  $H(X_1 \cup X_2, s, s') = 0$ .*
3.  $H(X_1 \times X_2, s, s') = 1$ , if and only if
  - $0 \leq s \leq s(X_2)$ <sup>9</sup> and  $0 \leq s' \leq s(X_2)$ , such that  $H(X_2, s, s') = 1$  or
  - there are  $1 \leq s_1 \leq s(X_1)$  and  $1 \leq s'_1 < i(X_1)$ , such that  $s_1 = s, 0 = s'$ , and  $H(X_1, s_1, s'_1) = 1$ , or
  - there are  $1 \leq s_1 \leq s(X_1)$ , such that  $s_1 + s(X_2) = s, i(X_2) = s'$ , and  $H(X_1, s_1, i(X_1)) = 1$ , or
  - there are  $1 \leq s_1 \leq s(X_1), 1 \leq s'_1 < i(X_1), 1 \leq s_2 \leq s(X_2)$ , and  $1 \leq s'_2 \leq s(X_2)$ , such that  $s_1 + s_2 = s, s'_2 = s', H(X_1, s_1, s'_1) = 1$ , and  $H(X_2, s_2, s'_2) = 1$ .

<sup>9</sup> The value  $s = s' = 0$  is for choosing an empty solution in digraph( $X_1 \times X_2$ ). The values  $s > s' = 0$  are for choosing a solution without sinks in digraph( $X_1 \times X_2$ ).

In all other cases  $H(X_1 \times X_2, s, s') = 0$ .

**Proof** We show the correctness of the stated equivalences. Let  $0 \leq s, s' \leq c$ .

1. The only possible solutions in  $\text{digraph}(a_j)$  are  $\emptyset$  and  $\{a_j\}$  which have size 0 and  $s_j$ , respectively. Further, a single vertex corresponds to a sink.
2. If  $H(X_1 \cup X_2, s, s') = 1$ , then by Lemma 7 there are  $s_1, s_2$  and  $s'_1, s'_2$  such that  $s_1 + s_2 = s, s'_1 + s'_2 = s'$  and solutions in  $\text{digraph}(X_1)$  and in  $\text{digraph}(X_2)$  which guarantee  $H(X_1, s_1, s'_1) = 1$  and  $H(X_2, s_2, s'_2) = 1$ .  
Further, for every  $0 \leq s_1 \leq s, 0 \leq s_2 \leq s, 0 \leq s'_1 \leq s', 0 \leq s'_2 \leq s'$ , such that  $s_1 + s_2 = s, s'_1 + s'_2 = s', H(X_1, s_1, s'_1) = 1$ , and  $H(X_2, s_2, s'_2) = 1$ , it holds that  $H(X_1 \cup X_2, s, s') = 1$  since we do not create any new edges by the parallel composition.
3. If  $H(X_1 \times X_2, s, s') = 1$ , then we distinguish four cases. If the solution of size  $s$  and sink size  $s'$  in  $\text{digraph}(X_1 \times X_2)$  contains no vertices of  $\text{digraph}(X_1)$ , then by Lemma 8 there is a solution in  $\text{digraph}(X_2)$  which guarantees  $H(X_2, s, s') = 1$ .  
If the solution of size  $s$  and sink size  $s'$  in  $\text{digraph}(X_1 \times X_2)$  contains only vertices of  $\text{digraph}(X_1)$  but not all sinks of  $\text{digraph}(X_1)$ , then by Lemma 8 there is a solution in  $\text{digraph}(X_1)$  which guarantees  $H(X_1, s, s') = 1$ .  
If the solution  $A'$  of size  $s$  and sink size  $s'$  in  $\text{digraph}(X_1 \times X_2)$  contains all sinks of  $\text{digraph}(X_1)$ , the series composition and the weak digraph constraint (3) imply that the set  $A'$  has to be extended by all sources of  $\text{digraph}(X_2)$ . After ignoring the sources of  $\text{digraph}(X_2)$  (because the graph is acyclic), there must exist new sources, which have to be contained in  $A'$ , since all their predecessors were sources in the original graph and so on. Thus, set  $A'$  contains all vertices of  $X_2$  and by Lemma 8 there is a solution in  $\text{digraph}(X_1)$  which guarantees  $H(X_1, s - s(X_2), i(X_1)) = 1$ .  
If the solution  $A'$  of size  $s$  and sink size  $s'$  in  $\text{digraph}(X_1 \times X_2)$  contains vertices of  $\text{digraph}(X_1)$  but not all sinks of  $\text{digraph}(X_1)$  and vertices of  $\text{digraph}(X_2)$ , then by Lemma 8 there are  $s_1, s'_1$  and  $s_2, s'_2$  such that  $s_1 + s_2 = s, s'_2 = s'$  and solutions in  $\text{digraph}(X_1)$  and in  $\text{digraph}(X_2)$  which guarantee  $H(X_1, s_1, s'_1) = 1$  and  $H(X_2, s_2, s'_2) = 1$ .  
Further, the solutions of size  $0 \leq s \leq s(X_2)$  from  $\text{digraph}(X_2)$  remain feasible in  $\text{digraph}(X_1 \times X_2)$  since the solutions from  $\text{digraph}(X_2)$  do not contain any predecessors of vertices from  $\text{digraph}(X_1)$  in  $\text{digraph}(X_1 \times X_2)$ .  
The solutions from  $\text{digraph}(X_1)$  which do not contain all sinks of  $X_1$ , i.e.  $1 \leq s'_1 < i(X_1)$  remain feasible in  $\text{digraph}(X_1 \times X_2)$ , but the sizes of sinks have to be changed to 0 since these sinks are no longer sinks in the  $\text{digraph}(X_1 \times X_2)$ .  
Next we consider solutions  $A'$  from  $\text{digraph}(X_1)$  which contain all sinks of  $\text{digraph}(X_1)$ , i.e.  $s' = i(X_1)$ . As mentioned above, the series composition and the weak digraph constraint (3) imply that the set  $A'$  has to be extended by all vertices of  $X_2$ . The sizes of sinks have to be changed to  $i(X_2)$ , since all sinks of  $X_2$  are also sinks in the  $\text{digraph}(X_1 \times X_2)$ .  
Further, we can combine solutions of size  $1 \leq s_1 \leq s(X_1)$  from  $\text{digraph}(X_1)$ , which do not contain all sinks of  $X_1$ , i.e.  $1 \leq s'_1 < i(X_1)$ , and solutions of size  $1 \leq s_2 \leq s(X_2)$  from  $\text{digraph}(X_2)$ , to a solution of size  $s_1 + s_2$  and sizes of sinks  $s'_2$  in  $\text{digraph}(X_1 \times X_2)$ .  $\square$

**Corollary 4** *There is a solution with sum  $s$  for some instance of SSGW such that  $G$  is a minimal series-parallel digraph which is given by some msp-expression  $X$  if and only if  $H(X, s, s') = 1$ . Therefore,  $OPT(I) = \max\{s \mid H(X, s, s') = 1\}$ .*

**Theorem 7** *SSGW can be solved in minimal series-parallel digraphs with  $n$  vertices and  $m$  arcs in  $\mathcal{O}(n \cdot c^4 + m)$  time and  $\mathcal{O}(n \cdot c^2)$  space.*

**Proof** Let  $G = (V, E)$  be a minimal series-parallel digraph and  $T$  be an msp-tree for  $G$  with root  $r$ . For some vertex  $u$  of  $T$  we denote by  $T_u$  the subtree rooted at  $u$  and  $X_u$  the msp-expression defined by  $T_u$ . In order to solve the SSGW problem for an instance  $I$  graph  $G$ , we traverse msp-tree  $T$  in a bottom-up order. For every vertex  $u$  of  $T$  and  $0 \leq s, s' \leq c$  we compute  $H(X_u, s, s')$  following the rules given in Lemma 17. By Corollary 4 we can solve our problem by  $H(X_r, s, s') = H(X, s, s')$ .

An msp-tree  $T$  can be computed in  $\mathcal{O}(n + m)$  time from a minimal series-parallel digraph with  $n$  vertices and  $m$  arcs, see Valdes et al. (1982). All  $s(X_i)$  and all  $i(X_i)$  can be precomputed in  $\mathcal{O}(n)$  time. Our rules given in Lemma 17 show the following running times.

- For every  $a_j \in A$  and every  $0 \leq s, s' \leq c$  value  $H(a_j, s, s')$  is computable in  $\mathcal{O}(1)$  time.
- For every  $0 \leq s, s' \leq c$ , every  $H(X_1 \cup X_2, s, s')$  can be computed in  $\mathcal{O}(c^2)$  time from  $H(X_1, s_1, s'_1)$  and  $H(X_2, s_2, s'_2)$ .
- For every  $0 \leq s, s' \leq c$ , every  $H(X_1 \times X_2, s, s')$  can be computed in  $\mathcal{O}(c^2)$  time from  $H(X_1, s_1, s'_1)$ ,  $H(X_2, s_2, s'_2)$ , and  $i(X_1)$ .

Since we have  $n$  leaves and  $n - 1$  inner vertices in  $T$ , the running time is in  $\mathcal{O}(nc^4 + m)$ .  $\square$

## 5 Conclusions and outlook

The presented methods allow us to solve SSG and SSGW with digraph constraints given by directed co-graphs and (minimal) series-parallel digraphs in pseudo-polynomial time.

In contrast to Gourvès et al. (2018) we did not consider null sizes. This allows us to verify whether a solution consists of all vertices or contains all sinks of a subgraph by using the sum of the sizes of the corresponding items. SSG and SSGW using null sizes can also be solved in pseudo-polynomial time on directed co-graphs and (minimal) series-parallel digraphs by additional counting the number of vertices or sinks within a SSGW solution.

For future work it could be interesting to find a solution for SSGW for series-parallel digraphs in general. Example 2 shows that Lemma 5 and the recursive structure of minimal series-parallel digraphs cannot be used in this case.

It remains to analyze whether the shown results also hold for other graph classes. Therefore one could consider edge series-parallel digraphs from Valdes et al. (1982). Further, it remains to look at more general graph classes, such as graphs of bounded directed clique-width. Directed clique-width measures the difficulty of decomposing a graph into a special tree-structure and was defined by Courcelle and Olariu in Courcelle and Olariu (2000). An alternative parameter is directed tree-width defined in

Johnson et al. (2001). Since in the directed case bounded directed tree-width does not imply bounded directed clique-width, solutions for subset sum problems with digraph constraints of bounded directed tree-width are interesting as well.

Furthermore, it could be useful to consider related problems. These include the two minimization problems which are introduced in Gourvès et al. (2018) by adding a maximality constraint to SSG and SSGW. Moreover, a generalization of the results for SSG to the partially ordered knapsack problem (Johnson and Niemi 1983; Kellerer and Pferschy 2004) is still open.

**Acknowledgements** The work of the second and third author was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 388221852.

**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/>.

## A Examples

### A.1 SSG on directed co-graphs

**Example 8** We consider the SSG instance  $I$  with  $n = 4$  items using digraph( $X$ ) shown in Fig. 3 and defined by the expression in (4),  $c = 7$ , and the following sizes.

$$\begin{array}{c|cccc} j & 1 & 2 & 3 & 4 \\ \hline s_j & 1 & 2 & 2 & 3 \end{array}$$

The rules given in Lemma 10 lead to the values in Table 2. Thus, the optimal solution is  $\{a_2, a_3, a_4\}$  with  $OPT(I) = 7$ .

**Table 2** Table for Example 8

		$F(X', s)$							
$X'$	$s$	0	1	2	3	4	5	6	7
$v_1$		1	1	0	0	0	0	0	0
$v_2$		1	0	1	0	0	0	0	0
$v_3$		1	0	1	0	0	0	0	0
$v_4$		1	0	0	1	0	0	0	0
$v_1 \oplus v_3$		1	1	1	1	0	0	0	0
$v_2 \otimes v_4$		1	0	0	0	0	1	0	0
$(v_1 \oplus v_3) \otimes (v_2 \otimes v_4)$		1	0	0	0	0	1	1	1

**Table 3** Table for Example 10

		$H'(X', s)$							
$X'$	$s$	0	1	2	3	4	5	6	7
$v_1$		1	1	0	0	0	0	0	0
$v_2$		1	0	1	0	0	0	0	0
$v_3$		1	0	1	0	0	0	0	0
$v_4$		1	0	0	1	0	0	0	0
$v_1 \oplus v_3$		1	1	1	1	0	0	0	0
$v_2 \otimes v_4$		1	0	1	1	0	1	0	0
$(v_1 \oplus v_3) \otimes (v_2 \otimes v_4)$		1	1	1	1	1	1	1	1

### A.2 SSG on transitive tournaments

**Example 9** We consider the SSG instance  $I$  with  $n = 4$  items using the transitive tournament digraph( $X$ ) defined by expression

$$X = (((v_1 \otimes v_2) \otimes v_3) \otimes v_4), \tag{8}$$

$c = 7$ , and the following sizes.

$$\begin{array}{c|cccc} j & 1 & 2 & 3 & 4 \\ \hline s_j & 1 & 2 & 2 & 3 \end{array}$$

By Remark 2 for this instance the feasible solutions of SSG w.r.t. the digraph constraint (2) are  $\emptyset, \{a_4\}, \{a_3, a_4\}, \{a_2, a_3, a_4\}$ , and  $\{a_1, a_2, a_3, a_4\}$ . Among these only  $\emptyset, \{a_4\}, \{a_3, a_4\}$ , and  $\{a_2, a_3, a_4\}$  satisfy the capacity constraint (1). Thus, the optimal solution is  $\{a_2, a_3, a_4\}$  with  $OPT(I) = 7$ .

### A.3 SSGW on directed co-graphs

**Example 10** We consider the SSGW instance  $I$  with  $n = 4$  items using digraph( $X$ ) shown in Fig. 3 and defined by the expression in (4),  $c = 7$ , and the following sizes.

$$\begin{array}{c|cccc} j & 1 & 2 & 3 & 4 \\ \hline s_j & 1 & 2 & 2 & 3 \end{array}$$

The rules given in Lemma 12 lead to the values in Table 3 and the rules given in Lemma 13 lead to the values in Table 4. Thus, the optimal solution is  $\{a_2, a_3, a_4\}$  with  $OPT(I) = 7$ .



**Table 5** Table for Example 11

$X'$	$s$	$F(X', s)$							
		0	1	2	3	4	5	6	7
$v_1$		1	0	1	0	0	0	0	0
$v_2$		1	1	0	0	0	0	0	0
$v_3$		1	0	0	0	1	0	0	0
$v_4$		1	0	0	1	0	0	0	0
$v_5$		1	0	1	0	0	0	0	0
$v_6$		1	0	0	1	0	0	0	0
$v_1 \times v_2$		1	1	0	1	0	0	0	0
$v_3 \times v_4$		1	0	0	1	0	0	0	1
$v_5 \times v_6$		1	0	0	1	0	1	0	0
$(v_1 \times v_2) \cup (v_3 \times v_4)$		1	1	0	1	1	0	1	1
$((v_1 \times v_2) \cup (v_3 \times v_4)) \times (v_5 \times v_6)$		1	0	0	1	0	1	1	0

### A.4 SSG on series-parallel digraphs

**Example 11** We consider the SSG instance  $I$  with  $n = 6$  items using digraph( $X$ ) defined by the expression in (7),  $c = 7$ , and the following sizes.

$$\frac{j}{s_j} \mid \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 1 & 4 & 3 & 2 & 3 \end{array}$$

The rules given in Lemma 16 lead to the values in Table 5. Thus, the optimal solution is  $\{a_2, a_5, a_6\}$  with  $OPT(I) = 6$ .

### A.5 SSGW on series-parallel digraphs

**Example 12** We consider the SSGW instance  $I$  with  $n = 6$  items using digraph( $X$ ) defined by the expression in (7),  $c = 7$ , and the following sizes.

$$\frac{j}{s_j} \mid \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 1 & 4 & 3 & 2 & 3 \end{array}$$

The rules given in Lemma 17 lead to the values in Table 6. Thus, the optimal solution is  $\{v_3, v_4\}$  with  $OPT(I) = 7$ .





## References

- Aho AV, Garey MR, Ullman JD (1972) The transitive reduction of a directed graph. *SIAM J Comput* 1(2):131–137
- Bang-Jensen J, Gutin G (eds) (2018) *Classes of directed graphs*. Springer, Berlin
- Bang-Jensen J, Maddaloni A (2014) Arc-disjoint paths in decomposable digraphs. *J Graph Theory* 77:89–110
- Bechet D, de Groote P, Retoré C (1997) A complete axiomatisation of the inclusion of series-parallel partial orders. *Rewriting techniques and applications*, volume 1232 of LNCS. Springer, Berlin, pp 230–240
- Cornel DG, Lerchs H, Stewart-Burlingham L (1981) Complement reducible graphs. *Discrete Appl Math* 3:163–174
- Courcelle B, Olariu S (2000) Upper bounds to the clique width of graphs. *Discrete Appl Math* 101:77–114
- Crespelle C, Paul C (2006) Fully dynamic recognition algorithm and certificate for directed cographs. *Discrete Appl Math* 154(12):1722–1741
- Engelfriet J, Rozenberg G (1997) Node replacement graph grammars. *Handbook of grammars and computing by graph transformation*. World Scientific, Singapore, pp 1–94
- Gould R (2012) *Graph theory*. Dover Publications Inc, New York
- Gourvès L, Monnot J, Tlilane L (2018) Subset sum problems with digraph constraints. *J Comb Optim* 36(3):937–964
- 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. *Theor Comput Sci* 616:1–17
- Gurski F, Komander D, Rehs C (2019a) Computing digraph width measures on directed co-graphs. In: *Proceedings of international symposium on fundamentals of computation theory (FCT)*, vol 11651 of LNCS. Springer, Berlin, pp 292–305
- Gurski F, Komander D, Rehs C (2019) Oriented coloring on recursively defined digraphs. *Algorithms* 12(4):87
- 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)*, vol 12011 of LNCS. Springer, Berlin, pp 556–565
- 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)*, vol 10976 LNCS. Springer, Berlin, pp 255–267
- 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
- Johnson DS, Niemi KA (1983) On knapsacks, partitions, and a new dynamic programming technique for trees. *Math Oper Res* 8(1):1–14
- Johnson T, Robertson N, Seymour PD, Thomas R (2001) Directed tree-width. *J Comb Theory Ser B* 82:138–155
- Kellerer H, Pferschy U (2004) A new fully polynomial time approximation scheme for the knapsack problem. *J Comb Optim* 8:5–11
- Kellerer H, Pferschy U, Pisinger D (2010) *Knapsack problems*. Springer, Berlin
- Lawler EL (1976) Graphical algorithms and their complexity. *Math Centre Tracts* 81:3–32
- Le Gall F (2014) Powers of tensors and fast matrix multiplication. In: *Proceedings of the international symposium on symbolic and algebraic computation (ISSAC)*. ACM, pp 296–303
- Monma CL, Sidney JB (1977) A general algorithm for optimal job sequencing with series-parallel constraints. *Math Oper Res* 4:215–224
- 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, Springer, New York pp 403–415
- Rendl F (1986) Quadratic assignment problems on series-parallel digraphs. *Z Oper Res Ser A-B* 30(3):A161–A173
- Steiner G (1985) A compact labeling scheme for series-parallel graphs. *Discrete Appl Math* 11(3):281–297

Valdes J, Tarjan RE, Lawler EL (1982) The recognition of series-parallel digraphs. *SIAM J Comput* 11:298–313

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