



# Dynamic Programming Algorithms on Directed Cographs

Frank Gurski\*

*Institute of Computer Science, University of Düsseldorf, Germany.*

(Received: 20 November 2016; Accepted: 19 December 2016)

**Abstract** In this paper we consider directed cographs, which are defined by Bechet et al. in [3] using the disjoint union, series, and order composition, from an algorithmic point of view. Applying their recursive structure we give dynamic programming algorithms to show that for every directed cograph the size of a largest edgeless subdigraph, the size of a largest subdigraph which is a tournament, the size of a largest semicomplete subdigraph, and the size of a largest complete subdigraph can be computed in linear time. Our main results show that the hamiltonian path, hamiltonian cycle, regular subdigraph, and directed cut problem are polynomial on directed cographs.

**Keywords** directed cograph, digraph, dynamic programming

**AMS 2010 subject classifications** 05C20, 90C39, 05C85

**DOI:** 10.19139/soic.v5i1.260

## 1. Introduction

*Undirected cographs* have been introduced in the 1970s by a number of authors under different notations, such as hereditary Dacey graphs (HD graphs) in [16],  $D^*$ -graphs in [12], 2-parity graphs in [5], and complement reducible graphs (cographs) in [13]. Cographs can be characterized as the set of graphs without an induced path with four vertices [6]. From an algorithmic point of view the following definition is very useful. (a) Every single vertex is a cograph. (b) If  $G_1, \dots, G_k$  are cographs, then the disjoint union of the graphs  $G_1, \dots, G_k$  is a cograph. (c) If  $G_1, \dots, G_k$  are cographs, then the join of the graphs  $G_1, \dots, G_k$  is a cograph. By this definition every cograph can be represented by a tree structure, denoted as *cotree*. The leaves of the cotree represent the vertices of the graph and the inner nodes of the cotree correspond to the operations applied on the subexpressions defined by the subtrees. For every graph  $G$  one can decide in linear time, whether  $G$  is a cograph and in the case of a positive answer construct a cotree for  $G$ , see [7] or [11]. Using the cotree a lot of hard problems have been shown to be solvable in polynomial time when restricted to cographs. Such problems are clique, independent set, partition into independent sets (chromatic number), partition into cliques, hamiltonian cycle, isomorphism [6], and vertex disjoint paths [10]. On the other hand, the achromatic number problem is NP-complete for cographs [4].

*Directed cographs* were introduced in 1997 by Bechet et al. in [3] as follows. (a) Every single vertex is a directed cograph. (b) If  $G_1, \dots, G_k$  are directed cographs, then the disjoint union of the graphs  $G_1, \dots, G_k$  is a directed cograph. (c) If  $G_1, \dots, G_k$  are directed cographs, then the series composition of the graphs  $G_1, \dots, G_k$  is a directed cograph. (d) If  $G_1, \dots, G_k$  are directed cographs, then the order composition of the graphs  $G_1, \dots, G_k$  is a directed cograph. Also every directed cograph can be defined by a tree structure, denoted as *di-cotree*. The leaves of the di-cotree represent the vertices of the graph and the inner nodes of the di-cotree correspond to the

\*Correspondence to: Frank Gurski (Email: frank.gurski@hhu.de). Institute of Computer Science, Algorithmics for Hard Problems Group, University of Düsseldorf, 40225 Düsseldorf, Germany

operations applied on the subexpressions defined by the subtrees. Also for every directed graph  $G$  one can decide in linear time, whether  $G$  is a directed cograph and in the case of a positive answer construct a di-cotree for  $G$ , see [8]. Directed cographs can be characterized by eight forbidden induced digraphs [8]. Calculus of directed cographs were also considered in connection with pomset logic in [14]. The algorithmic use of directed cographs has been shown in [2] using their decomposability into acyclic digraphs and semicomplete graphs and also by flow techniques (cf. Section 4.11.3 in [1]).

In this paper we use the recursive structure of directed cographs to give new dynamic programming solutions for the following problems. We show that for every directed cograph a largest edgeless subdigraph, a largest subdigraph which is a tournament, a largest semicomplete subdigraph, and a largest complete subdigraph can be computed in linear time. Further we show that the hamiltonian path, hamiltonian cycle, regular subdigraph, and directed cut problem are polynomial on directed cographs. For general digraphs all these problems are NP-hard.

Our approach can be applied to a lot of further digraph problems. The aim of this paper is not to complete the list of digraph problems which can be solved by dynamic programming on directed cographs, but to show that all these problems can be handled by dynamic programming solutions using the same scheme. We are also not interested in bounding the running time which is sometimes given by a polynomial of higher degree. Such considerations are only interesting if anybody really has to implement the solutions.

## 2. Preliminaries

### 2.1. Digraphs

A *directed graph* or *digraph* is a pair  $G = (V, A)$ , where  $V$  is a finite set of *vertices* and  $A \subseteq \{(u, v) \mid u, v \in V, u \neq v\}$  is a finite set of ordered pairs of distinct<sup>†</sup> vertices called *directed edges* or *arcs*. For a vertex  $v \in V$ , the sets  $N_G^+(v) = \{u \in V \mid (v, u) \in A\}$  and  $N_G^-(v) = \{u \in V \mid (u, v) \in A\}$  are called the *set of all out-neighbours* and the *set of all in-neighbours* of  $v$ . The *outdegree* of  $v$ ,  $\text{out-deg}_G(v)$  for short, is the number of out-neighbours of  $v$  and the *indegree* of  $v$ ,  $\text{in-deg}_G(v)$  for short, is the number of in-neighbours of  $v$  in  $G$ . A digraph  $G' = (V', A')$  is a *subdigraph* of digraph  $G = (V, A)$  if  $V' \subseteq V$  and  $A' \subseteq A$ . If every arc of  $A$  with both end vertices in  $V'$  is in  $A'$ , we say that  $G'$  is an *induced subdigraph* of digraph  $G$  and we write  $G' = G[V']$ . For some directed graph  $G = (V, E)$  its complement digraph is defined by  $\text{co-}G = (V, \{(u, v) \mid (u, v) \notin E, u, v \in V, u \neq v\})$ . For some given digraph  $G = (V, E)$ , we define its underlying undirected graph by ignoring the directions of the edges, i.e.  $u(G) = (V, \{\{u, v\} \mid (u, v) \in E, u, v \in V\})$ .

### 2.2. Directed Cographs

Let  $G_1 = (V_1, E_1), \dots, G_k = (V_k, E_k)$  be  $k$  vertex-disjoint directed graphs.

- The *disjoint union* of  $G_1, \dots, G_k$ , denoted by  $G_1 \oplus \dots \oplus G_k$ , is the graph with vertex set  $V_1 \cup \dots \cup V_k$  and arc set  $E_1 \cup \dots \cup E_k$ .
- The *series composition* of  $G_1, \dots, G_k$ , denoted by  $G_1 \otimes \dots \otimes G_k$ , is defined by their disjoint union plus all possible arcs between vertices of  $G_i$  and  $G_j$  for all  $1 \leq i, j \leq k, i \neq j$ .
- The *order composition* of  $G_1, \dots, G_k$ , denoted by  $G_1 \circ \dots \circ G_k$ , is defined by their disjoint union plus all possible arcs between vertices of  $G_i$  and  $G_j$  for all  $1 \leq i < j \leq k$ .

The class of directed cographs has been defined recursively in [3].

- (i) Every digraph on a single vertex  $(\{v\}, \emptyset)$ , denoted by  $\bullet$ , is a directed cograph.
- (ii) If  $G_1, \dots, G_k$  are directed cographs, then the disjoint union  $G_1 \oplus \dots \oplus G_k$  is a directed cograph.

<sup>†</sup>Thus we do not consider directed graphs with loops.

- (iii) If  $G_1, \dots, G_k$  are directed cographs, then the series composition  $G_1 \otimes \dots \otimes G_k$  is a directed cograph.
- (iv) If  $G_1, \dots, G_k$  are directed cographs, then the order composition  $G_1 \circ \dots \circ G_k$  is a directed cograph.

Obviously for every directed cograph we can define a tree structure, denoted as *di-cotree*. The leaves of the di-cotree represent the vertices of the graph and the inner nodes of the di-cotree correspond to the operations applied on the subexpressions defined by the subtrees. For every directed cograph one can construct a di-cotree in linear time, see [8].





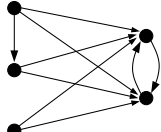
Since the disjoint union  $\oplus$ , the series composition  $\otimes$ , and the order composition  $\circ$  are associative, we can recursively transform every di-cotree  $T$  into a binary di-cotree  $T'$  for  $G$ .

*Lemma 1*

Every cotree  $T$  can be transformed into an equivalent binary di-cotree  $T'$ , such that every inner vertex in  $T'$  has exactly two sons.

*Example 1 (Directed cograph)*

Table I shows the definition of a directed cograph and its di-cotree.

cograph expression $X_i$	digraph defined by $X_i$
$X_1 = \bullet$	
$X_2 = X_1 \circ X_1$	
$X_3 = X_2 \oplus X_1$	
$X_4 = X_1 \otimes X_1$	
$X_5 = X_3 \circ X_4$	

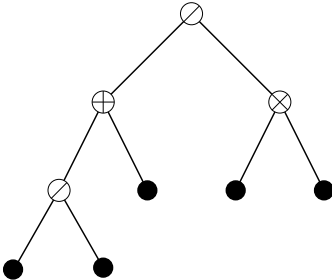


Table I. Directed cograph expressions  $X_1, \dots, X_5$ , the defined digraphs, and a di-cotree for the directed cograph defined by expression  $X_5$ .

*Lemma 2*

For every directed cograph its complement digraph is a directed cograph.

*Proof*

Let  $G$  be a directed cograph and  $X$  be a cograph expression for  $G$ . An expression  $c(X)$  for the complement graph  $co-G$  can recursively be defined as follows. If  $X = \bullet$ , then  $c(X) = \bullet$ . If  $X = X_1 \oplus X_2$ , then  $c(X) = c(X_1) \otimes c(X_2)$ . If  $X = X_1 \otimes X_2$ , then  $c(X) = c(X_1) \oplus c(X_2)$ . If  $X = X_1 \circ X_2$ , then  $c(X) = c(X_2) \circ c(X_1)$ .  $\square$

An obvious but important property of directed cographs is that they are hereditary, that is, every induced subdigraph of a directed cograph is a directed cograph. The proof follows, since an expression  $X'$  for an induced subdigraph  $G'$  can be obtained from an expression  $X$  of original graph  $G$  by restricting  $X$  to the vertices of  $G'$ .

*Lemma 3*

The set of directed cographs is hereditary.

Regarding the algorithmic use of special digraphs it is often helpful to know whether they are transitive or at least quasi transitive (cf. Section 6.7 in [1]). A digraph  $G$  is *transitive* if for every pair of arcs  $(x, y)$  and  $(y, z)$  with  $x \neq z$  the arc  $(x, z)$  also belongs to  $G$ . A digraph  $G$  is *quasi transitive* if for every pair of arcs  $(x, y)$  and  $(y, z)$  with  $x \neq z$  at least one of the arcs  $(x, z)$  and  $(z, x)$  also belongs to  $G$ . By the simple  $(X_1 \oplus X_3) \otimes X_2$  expression, where  $X_1$  consists of one vertex  $x$ ,  $X_2$  consists of one vertex  $y$ , and  $X_3$  consists of one vertex  $z$  the following result follows.

*Lemma 4*

The set of directed cographs is not transitive and not quasi transitive.

In the next sections we give new and efficient algorithms for problems on directed cographs. By Lemma 1 and [8] we can assume that we are given our directed input cograph as a binary di-cotree. As an introductory example, we briefly consider the task of finding a subdigraph of special arc sets.

### 3. Independent Sets and Cliques

Let  $G = (V, E)$  be a digraph.  $G$  is *edgeless* if for all  $u, v \in V$ ,  $u \neq v$ , none one of the two pairs  $(u, v)$  and  $(v, u)$  belongs to  $E$ .  $G$  is a *tournament* if for all  $u, v \in V$ ,  $u \neq v$ , exactly one of the two pairs  $(u, v)$  and  $(v, u)$  belongs to  $E$ .  $G$  is *semicomplete* if for all  $u, v \in V$ ,  $u \neq v$ , at least one of the two pairs  $(u, v)$  and  $(v, u)$  belongs to  $E$ .  $G$  is (*bidirectional*) *complete* if for all  $u, v \in V$ ,  $u \neq v$ , both of the two pairs  $(u, v)$  and  $(v, u)$  belong to  $E$ . The problems of finding a subdigraph of maximal size with one of these four properties  $P$  can be summarized as follows.

**Name** DIRECTED  $P$ -SET

**Instance** A directed graph  $G = (V, E)$  and a positive integer  $k$ .

**Question** Is there a subdigraph  $(V', E')$  with property  $P$  on  $k$  vertices in  $G$ ?

Let  $G$  be our directed input cograph defined by some expression  $X$ . Table II shows how we can recursively compute the size  $F(X)$  of a largest edgeless subdigraph (column 2), a largest subdigraph which is a tournament (column 3), a largest semicomplete subdigraph (column 4), and a largest complete subdigraph (column 5) in digraph defined by expression  $X$  from the subexpressions  $X_1$  and  $X_2$ . Operation  $+$  means that  $F(X) = F(X_1) + F(X_2)$  and operation  $\max$  means that  $F(X) = \max\{F(X_1), F(X_2)\}$ .

	edgeless	tournament	semicomplete	complete
$F(\bullet)$	1	1	1	1
$F(X_1 \oplus X_2)$	+	max	max	max
$F(X_1 \otimes X_2)$	max	max	+	+
$F(X_1 \circ X_2)$	max	+	+	max

Table II. The table shows the operations to compute size  $F(X)$  of a largest edgeless subdigraph, largest subdigraph which is a tournament, largest semicomplete subdigraph, and largest complete subdigraph in digraph defined by expression  $X$  from the subexpressions  $X_1$  and  $X_2$ .

*Theorem 1*

For every directed cograph on  $n$  vertices the size of a largest edgeless subdigraph, the size of a largest subdigraph which is a tournament, the size of a largest semicomplete subdigraph, and the size of a largest complete subdigraph can be computed in  $\mathcal{O}(n)$  time.

Besides computing the sizes of the special vertex sets we additionally can compute a vertex set of this size following Table II. If the entry in the table is 1, we store the corresponding vertex. For  $+$  we store the union of the sets of the involved subexpressions. And for  $\max$  we store the set of maximum size among the sets of the involved subexpressions. The correctness follows since on the one hand all computed vertex sets induce a subdigraph of the

given property. On the other hand a larger set of the given property is not possible. In the case of  $+$  is this obvious by the size of the subdigraph and in the case of  $\max$  the considered operation does not allow to choose vertices from both involved subdigraphs in order to fulfill the property, so it is best possible to choose the larger subdigraph.

#### 4. Directed Cut

**Name** DIRECTED CUT

**Instance** A directed graph  $G = (V, E)$  and a positive integer  $k$ .

**Question** Is there a partition of  $V$  into  $V_1$  and  $V_2$ , such that  $|\{(v_1, v_2) \in E \mid v_1 \in V_1, v_2 \in V_2\}| \geq k$ ?

Let  $G = (V, E)$  be our directed input cograph defined by some binary expression  $X$ . Let  $F(X)$  to be the set of all triples  $(i, x, n - i)$  for all disjoint partitions of  $V = \{u_1, \dots, u_n\}$  into  $V_1 = \{u_1, \dots, u_i\}$  and  $V_2 = \{u_{i+1}, \dots, u_n\}$  such that  $x$  is the number of directed edges starting from vertices in  $V_1$  and ending in vertices in  $V_2$ . Then  $F(X)$  has at most  $(|V| + 1) \cdot (|E| + 1) \cdot (|V| + 1) \in \mathcal{O}(|V|^2 \cdot |E|)$  triples.

The following observations show that  $F(\bullet)$  is computable in  $\mathcal{O}(1)$  time, and  $F(X_1 \oplus X_2)$ ,  $F(X_1 \otimes X_2)$ , and  $F(X_1 \circ X_2)$ , are computable in polynomial time from  $F(X_1)$  and  $F(X_2)$ .

1.  $F(\bullet) = \{(1, 0, 0), (0, 0, 1)\}$

Given a single vertex graph, we allow this vertex to be in  $V_1$  or in  $V_2$  of our partition.

2.  $F(X_1 \oplus X_2) = \{(n'_1 + n''_1, x' + x'', n'_2 + n''_2) \mid (n'_1, x', n'_2) \in F(X_1), (n''_1, x'', n''_2) \in F(X_2)\}$ .

For the disjoint union operation, we know the sizes of vertex sets in every partition  $V'_1 \cup V''_1$  of the vertex set of the graph defined by  $X_1$  and also the sizes of vertex sets in every partition  $V'_2 \cup V''_2$  of the vertex set of the graph defined by  $X_2$ . To realize the disjoint union, we store the size of the sets  $V'_1 \cup V'_2$  and  $V''_1 \cup V''_2$ . Since no additional edges are inserted, the number of edges is the sum of those of the two involved subdigraphs.

3.  $F(X_1 \otimes X_2) = \{(n'_1 + n''_1, x' + x'' + n'_1 \cdot n''_2 + n'_2 \cdot n''_1, n'_2 + n''_2) \mid (n'_1, x', n'_2) \in F(X_1), (n''_1, x'', n''_2) \in F(X_2)\}$

For the series operation, we perform the same combination of the sizes of the computed partitions. Since all possible edges are inserted in both directions, the number of relevant edges for the new partition  $V_1 \cup V_2$  is the sum of those of the two involved subdigraphs, i.e.  $x'$  from  $V'_1$  to  $V''_1$  and  $x''$  from  $V'_2$  to  $V''_2$  plus the number of new edges from  $V'_1$  to  $V''_2$ , i.e.  $n'_1 \cdot n''_2$  plus the number of new edges from  $V'_2$  to  $V''_1$ , i.e.  $n'_2 \cdot n''_1$ .

4.  $F(X_1 \circ X_2) = \{(n'_1 + n''_1, x' + x'' + n'_1 \cdot n''_2, n'_2 + n''_2) \mid (n'_1, x', n'_2) \in F(X_1), (n''_1, x'', n''_2) \in F(X_2)\}$

For the order operation, we perform the same combination of the sizes of the computed partitions. Since only edges from  $V_1$  to  $V_2$  are inserted, the number of relevant edges for the new partition  $V_1 \cup V_2$  is the sum of those of the two involved subdigraphs, i.e.  $x'$  from  $V'_1$  to  $V''_1$  and  $x''$  from  $V'_2$  to  $V''_2$  plus the number of new edges from  $V'_1$  to  $V''_2$ , i.e.  $n'_1 \cdot n''_2$ .

There is a partition of the vertex set of directed cograph  $G$  into two disjoint sets  $V_1, V_2$ , such that there are at least  $r$  directed edges starting from vertices in  $V_1$  and ending in vertices in  $V_2$ , if and only if there is some triple  $(n_1, x, n_2) \in F(X)$  such that  $x \geq r$ .

*Theorem 2*

The directed cut problem for directed cographs on  $n$  vertices and  $m$  arcs is computable in  $\mathcal{O}(n^2 \cdot m)$  time.

#### 5. Directed Hamiltonian Path and Hamiltonian Cycle

**Name** DIRECTED HAMILTONIAN PATH

**Instance** A directed graph  $G = (V, E)$ .

**Question** Is there a simple directed path on  $|V|$  vertices in  $G$ ?

In order to solve the directed hamiltonian path and also the directed hamiltonian cycle problem on directed cographs one may consider the fact that directed cographs are totally decomposable into acyclic digraphs and semicomplete graphs (cf. Corollary 7.2 in [2]) and that for those digraphs both problems can be solved in polynomial time (cf. Section 6.7 in [1]). We next give different dynamic programming solutions.

Let  $G = (V, E)$  be our directed input cograph and  $X$  be a binary expression for  $G$ . For every partition of  $G$  into directed paths we want to store the number of these paths. To this end, for every subset  $E' \subseteq E$  such that the underlying undirected graph  $u(G')$  of subdigraph  $G' = (V, E')$  contains no cycles and every vertex in  $G'$  has indegree and outdegree at most 1, we define an integer  $n_{E'}$  as follows. Integer  $n_{E'}$  is the number of directed path  $v_1, \dots, v_r, r \geq 1$ , in  $G'$ , where  $\text{in-deg}(v_1) = 0$  and  $\text{out-deg}(v_r) = 0$ . Let  $F(X)$  be the set of all integers  $n_{E'}$  for all such subsets  $E' \subseteq E$ . Since  $1 \leq n_{E'} \leq |V|$ , we know that  $F(X)$  has at most  $|V|$  integers.

*Example 2 ( $F(X)$ )*

Let  $G = (V, E)$  be some digraph defined by some cograph expression  $X$ . Figure 1 shows two examples for subdigraphs  $G' = (V, E')$  such that the underlying undirected graph  $u(G')$  of  $G'$  contains no cycles and every vertex in  $G'$  has indegree and outdegree at most 1. These graphs lead the values 3 and 5 for  $n_{E'}$ . That is,  $F(X) = \{\dots, 3, 5, \dots\}$ .

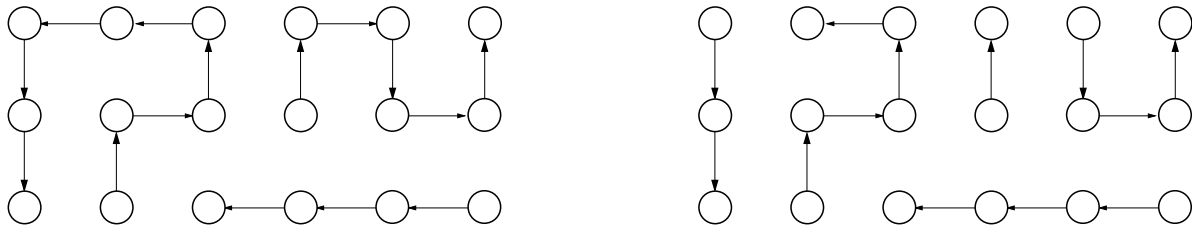


Figure 1. Two examples for subdigraphs  $G' = (V, E')$  such that the underlying undirected graph of  $G'$  contains no cycles and every vertex in  $G'$  has indegree and outdegree at most 1.

The following observations show that  $F(\bullet)$  is computable in  $\mathcal{O}(1)$  time, and  $F(X_1 \oplus X_2)$ ,  $F(X_1 \otimes X_2)$ , and  $F(X_1 \circ X_2)$ , are computable in polynomial time from  $F(X_1)$  and  $F(X_2)$ .

1.  $F(\bullet) = \{1\}$

A single vertex graph corresponds to one path.

2.  $F(X_1 \oplus X_2) = \{n_1 + n_2 \mid n_1 \in F(X_1), n_2 \in F(X_2)\}$

For the disjoint union operation, no additional edges are inserted and thus we add the number of paths of the graph defined by  $X_1$  and the graph defined by  $X_2$ .

3.  $F(X_1 \otimes X_2)$  can be obtained as follows. Since we insert all new edges from the graph defined by  $X_1$  to the graph defined by  $X_2$  and all edges from the graph defined by  $X_2$  to the graph defined by  $X_1$ , we can glue a number of paths from  $X_1$  and a number of paths from  $X_2$  to one new path. Both numbers may differ at most by one. We start with the set of triples  $D = \{0\} \times F(X_1) \times F(X_2)$  and for every existing triple  $(n, n', n'') \in D$  we insert into  $D$  the triples

- $(n + 1, n' - 1, n'')$  in order to pick one path of  $X_1$ ,
- $(n + 1, n', n'' - 1)$  in order to pick one path of  $X_2$ ,
- $(n + 1, n' - x, n'' - x)$  in order to pick  $x \leq n'$  paths from  $X_1$  and  $x \leq n''$  paths from  $X_2$  and glue them together by the edges inserted by the series operation,
- $(n + 1, n' - x, n'' - x - 1)$  in order to pick  $x \leq n'$  paths from  $X_1$  and  $x + 1 \leq n''$  paths from  $X_2$  and glue them together by the edges inserted by the series operation, and

- $(n + 1, n' - x, n'' - x + 1)$  in order to pick  $x \leq n'$  paths from  $X_1$  and  $x - 1 \leq n''$  paths from  $X_2$  and glue them together by the edges inserted by the series operation.

The last three operations are applied for all  $x$  meeting the stated conditions.

Then  $D$  gets at most  $(|V| + 1)^3$  triples and thus is computable in polynomial time.  $F(X_1 \otimes X_2) = \{n \mid (n, 0, 0) \in D\}$

4.  $F(X_1 \circ X_2)$  can be obtained as follows. Since we now only insert all new edges from the graph defined by  $X_1$  to the graph defined by  $X_2$ , we can glue one path from  $F(X_1)$  and one path from  $F(X_2)$  to one new path. We start with the set of triples  $D = \{0\} \times F(X_1) \times F(X_2)$  and for every existing triple  $(n, n', n'') \in D$  we add into  $D$  the three triples

- $(n + 1, n' - 1, n'')$  in order to pick one path of  $X_1$ ,
- $(n + 1, n', n'' - 1)$  in order to pick one path of  $X_2$ , and
- $(n + 1, n' - 1, n'' - 1)$  in order to pick one path from  $X_1$  and one path from  $X_2$  and glue them together by one of the edges inserted by the order operation.

Then  $D$  gets at most  $(|V| + 1)^3$  triples and thus is computable in polynomial time.  $F(X_1 \circ X_2) = \{n \mid (n, 0, 0) \in D\}$

There is a directed hamiltonian path in digraph defined by expression  $X$  if and only if  $1 \in F(X)$ .

### Theorem 3

The directed hamiltonian path problem for directed cographs is computable in polynomial time.

**Name** DIRECTED HAMILTONIAN CYCLE

**Instance** A directed graph  $G = (V, E)$ .

**Question** Is there a simple directed cycle on  $|V|$  vertices in  $G$ ?

Let  $G$  be our directed input graph defined by some directed cograph expression  $X$ . Obviously  $G$  can only contain a directed hamiltonian cycle, if there are two cograph expressions  $X'$  and  $X''$ , such that expression  $X$  is obtained by  $X = X' \otimes X''$ . In order to decide whether there is a directed hamiltonian cycle in digraph  $G$  we can apply the same data structure  $F(X)$  as for the hamiltonian path problem. There is a directed hamiltonian cycle in digraph defined by expression  $X$  if and only if there exists some integer  $x$ , such that  $x \in F(X') \cap F(X'')$ . By Theorem 3 we conclude the following result.

### Theorem 4

The directed hamiltonian cycle problem for directed cographs is computable in polynomial time.

## 6. Regular Subdigraph

Let  $G = (V, E)$  be a directed graph and  $d$  be some fixed integer.  $G$  is *d-in-regular*, if  $\text{in-deg}(v) = d$  for every  $v \in V$ ,  $G$  is *d-out-regular*, if  $\text{out-deg}(v) = d$  for every  $v \in V$ , and  $G$  is *d-regular*, if  $G$  is *d-in-regular* and *d-out-regular*.

**Name** *d*-REGULAR SUBDIGRAPH

**Instance** A directed graph  $G = (V, E)$ .

**Question** Is there a *d*-regular subdigraph  $(V', E')$  of  $G$ ?

For  $d = 1$  this problem is trivial since such a subdigraph exists unless  $G$  is acyclic. For  $d \geq 2$  the problem is NP-complete [9]. The related problem of finding a spanning *d*-regular subdigraph  $(V, E')$  can be solved in polynomial time using an algorithm based on flows (cf. Section 4.11.3 in [1]). In order to give a dynamic programming approach, we next apply the recursive definition of a directed cograph.

Let  $G = (V, E)$  be our directed input cograph defined by some cograph expression  $X$ . For a subdigraph  $G' = (V, E')$ ,  $E' \subset E$ , of  $G$  let  $\mathcal{M}$  be the multiset such that for every vertex  $u$  of  $V$ ,  $\mathcal{M}$  has a pair  $(\text{in-deg}(u), \text{out-deg}(u))$ , where  $\text{in-deg}(u)$  and  $\text{out-deg}(u)$  is the indegree and outdegree, respectively, of  $u$  in the subdigraph  $G'$ . For some fixed integer  $d$ , let  $F(X)$  be the set of all mutually different multisets  $\mathcal{M}$  for all subdigraphs of  $G$  in which all vertices have indegree at most  $d$  and outdegree at most  $d$ . Then  $F(X)$  has at most  $(|V| + 1)^{(d+1)^2}$  mutually different multisets each with at most  $|V|$  pairs  $(d_1, d_2)$ , where  $d_1$  and  $d_2$  are integers from 0 to  $d$ .

The following observations show that  $F(\bullet)$  is computable in  $\mathcal{O}(1)$  time, and  $F(X_1 \oplus X_2)$ ,  $F(X_1 \otimes X_2)$ , and  $F(X_1 \circ X_2)$ , are computable in polynomial time from  $F(X_1)$  and  $F(X_2)$ .

1.  $F(\bullet) = \{ \langle (0, 0) \rangle \}$

A single vertex has indegree and outdegree 0.

2.  $F(X_1 \oplus X_2) = \{ \mathcal{M}' \cup \mathcal{M}'' \mid \mathcal{M}' \in F(X_1), \mathcal{M}'' \in F(X_2) \}$

For the disjoint union operation, no additional edges are inserted and thus we combine every subdigraph of the graph defined by  $X_1$  and subdigraph of the graph defined by  $X_2$ .

3.  $F(X_1 \otimes X_2)$  can be obtained as follows. Since we insert all new edges from the graph defined by  $X_1$  to the graph defined by  $X_2$  and all edges from the graph defined by  $X_2$  to the graph defined by  $X_1$ , we can increase the indegree and outdegree of the vertices in graph defined by  $X_1$  by the vertices in graph defined by  $X_2$ , see (a), and increase the indegree and outdegree of the vertices in graph defined by  $X_2$  by the vertices in graph defined by  $X_1$ , see (b). It suffices to do this up to our bound of  $d$ . We start with set  $D = \{ \langle \rangle \} \times F(X_1) \times F(X_2)$  and insert into  $D$  all triples that can be obtained from some triple  $(\mathcal{M}, \mathcal{M}', \mathcal{M}'') \in D$  by

- (a) removing a pair  $(d'_1, d'_2)$  from  $\mathcal{M}'$  and choosing  $r$ ,  $0 \leq r \leq d - d'_2$ , pairs

$$(d''_{1,1}, d''_{2,1}), \dots, (d''_{1,r}, d''_{2,r})$$

from  $\mathcal{M}''$  and  $s$ ,  $0 \leq s \leq d - d'_1$ , pairs

$$(d''_{1,r+1}, d''_{2,r+1}), \dots, (d''_{1,r+s}, d''_{2,r+s})$$

from  $\mathcal{M}''$  such that  $d''_{1,t} < d$  for  $t = 1, \dots, r$  and  $d''_{2,t} < d$  for  $t = r + 1, \dots, r + s$ , and inserting  $(d'_1 + s, d'_2 + r)$  into  $\mathcal{M}$  and changing every  $(d''_{1,t}, d''_{2,t})$  into  $(d''_{1,t} + 1, d''_{2,t})$  for  $t = 1, \dots, r$  and every  $(d''_{1,t}, d''_{2,t})$  into  $(d''_{1,t}, d''_{2,t} + 1)$  for  $t = r + 1, \dots, r + s$  without removing them from  $\mathcal{M}''$ .

- (b) removing a pair  $(d''_1, d''_2)$  from  $\mathcal{M}''$  and choosing  $r$ ,  $0 \leq r \leq d - d''_2$ , pairs

$$(d'_{1,1}, d'_{2,1}), \dots, (d'_{1,r}, d'_{2,r})$$

from  $\mathcal{M}'$  and  $s$ ,  $0 \leq s \leq d - d''_1$ , pairs

$$(d'_{1,r+1}, d'_{2,r+1}), \dots, (d'_{1,r+s}, d'_{2,r+s})$$

from  $\mathcal{M}'$  such that  $d'_{1,t} < d$  for  $t = 1, \dots, r$  and  $d'_{2,t} < d$  for  $t = r + 1, \dots, r + s$ , and inserting  $(d''_1 + s, d''_2 + r)$  into  $\mathcal{M}$  and changing every  $(d'_{1,t}, d'_{2,t})$  into  $(d'_{1,t} + 1, d'_{2,t})$  for  $t = 1, \dots, r$  and every  $(d'_{1,t}, d'_{2,t})$  into  $(d'_{1,t}, d'_{2,t} + 1)$  for  $t = r + 1, \dots, r + s$  without removing them from  $\mathcal{M}'$ .

$$F(X_1 \otimes X_2) = \{ \mathcal{M} \mid (\mathcal{M}, \langle \rangle, \langle \rangle) \in D \}$$

4.  $F(X_1 \circ X_2)$  can be obtained as follows. Since we now only insert all new edges from the graph defined by  $X_1$  to the graph defined by  $X_2$ , we only can increase the outdegree of the vertices in graph defined by  $X_1$  by the vertices in graph defined by  $X_2$ , see (a), and increase the indegree of the vertices in graph defined by  $X_2$  by the vertices in graph defined by  $X_1$ , see (b). Again, it suffices to do this up to our bound of  $d$ . We start with set  $D = \{ \langle \rangle \} \times F(X_1) \times F(X_2)$  and insert in  $D$  all triples that can be obtained from some triple  $(\mathcal{M}, \mathcal{M}', \mathcal{M}'') \in D$  by



- (a) removing a pair  $(d'_1, d'_2)$  from  $\mathcal{M}'$  and choosing  $r$ ,  $0 \leq r \leq d - d'_2$ , pairs

$$(d''_{1,1}, d''_{2,1}), \dots, (d''_{1,r}, d''_{2,r})$$

from  $\mathcal{M}''$  such that  $d''_{1,t} < d$  for  $t = 1, \dots, r$  and inserting  $(d'_1, d'_2 + r)$  into  $\mathcal{M}$  and changing every  $(d''_{1,t}, d''_{2,t})$  into  $(d''_{1,t} + 1, d''_{2,t})$  for  $t = 1, \dots, r$  without removing them from  $\mathcal{M}''$ .

- (b) removing a pair  $(d''_1, d''_2)$  from  $\mathcal{M}''$  and choosing  $s$ ,  $0 \leq s \leq d - d''_1$ , pairs

$$(d'_{1,1}, d'_{2,1}), \dots, (d'_{1,s}, d'_{2,s})$$

from  $\mathcal{M}'$  such that  $d'_{2,t} < d$  for  $t = 1, \dots, s$ , and inserting  $(d''_1 + s, d''_2)$  into  $\mathcal{M}$  and changing every  $(d'_{1,t}, d'_{2,t})$  into  $(d'_{1,t}, d'_{2,t} + 1)$  for  $t = 1, \dots, s$  without removing them from  $\mathcal{M}'$ .

$$F(X_1 \circ X_2) = \{\mathcal{M} \mid (\mathcal{M}, \langle \rangle, \langle \rangle) \in D\}$$

There is a  $d$ -regular subdigraph in the directed cograph defined by  $X$  if and only if there is some  $\mathcal{M} \in F(X)$  such that for all  $(d_1, d_2) \in \mathcal{M}$ ,  $d_1 = d_2 = d$  or  $d_1 = d_2 = 0$  and there is at least one  $(d_1, d_2) \in \mathcal{M}$  with  $d_1 = d$  and  $d_2 = d$ . We conclude the following result.

*Theorem 5*

The  $d$ -regular subdigraph problem for directed cographs on  $n$  vertices is computable in  $\mathcal{O}(n^{d^2})$  time.

## 7. Conclusions and Outlook

Since the algorithmic use of directed cographs seems to be nearly unexploited until now, we introduced new and efficient dynamic programming solutions for well known digraph problems using dynamic programming on directed cographs.

There are several interesting open questions. (a) For the non-linear time solutions: Can we find linear time algorithms for the problems on directed cographs? (b) On undirected cographs the hamiltonian cycle problem can be solved using the so called scattering number, see [12]. It would be interesting whether it is possible to define a similar number in order to decide the problem on directed graphs. (c) Are there any natural problems which can be solved in polynomial time on undirected cographs and its directed version becomes NP-hard on directed cographs? (d) Can we give a polynomial-time algorithm for the oriented vertex coloring (cf. [15] for a definition) problem on directed corecursively graphs?

## REFERENCES

1. J. Bang-Jensen, and G. Gutin, *Digraphs. Theory, Algorithms and Applications*, Springer-Verlag, Berlin, 2009.
2. J. Bang-Jensen, and A. Maddaloni, *Arc-disjoint paths in decomposable digraphs*, Journal of Graph Theory, vol. 77, pp. 89–110, 2014.
3. D. Bechet, P. de Groote, and C. Retoré, *A complete axiomatisation of the inclusion of series-parallel partial orders*, Proceedings of the 8th International Conference on Rewriting Techniques and Applications, Lecture Notes in Computer Science, vol. 1232, pp. 230–240, 1997.
4. H. Bodlaender, *Achromatic number is NP-complete for cographs and interval graphs*, Information Processing Letters, vol. 31, no. 3, pp. 135–138, 1989.
5. M. Buriel, and J. Uhry, *Parity graphs*, Annals of Discrete Mathematics, vol. 21, pp. 253–277, 1984.
6. D. Corneil, H. Lerchs, and L. Stewart-Burlingham, *Complement reducible graphs*, Discrete Applied Mathematics, vol. 3, pp. 163–174, 1981.
7. D. Corneil, Y. Perl, and L. Stewart, *A linear recognition algorithm for cographs*, SIAM Journal on Computing, vol. 14, no. 4, pp. 926–934, 1985.
8. C. Crespelle, and C. Paul, *Fully dynamic recognition algorithm and certificate for directed cographs*, Discrete Applied Mathematics, vol. 154, no. 12, pp. 1722–1741, 2006.
9. S. Friedlan, *Every 7-regular digraph contains an even cycle*, Journal of Combinatorial Theory, Series B, vol. 46, no. 2, pp. 249–252, 1989.
10. F. Gurski, and E. Wanke, *Vertex disjoint paths on clique-width bounded graphs*, Theoretical Computer Science, vol. 359, no. 1-3, pp. 188–199, 2006.

11. M. Habib, and C. Paul, *A simple linear time algorithm for cograph recognition*, Discrete Applied Mathematics, vol. 145, pp. 183–197, 2005.
12. H. Jung, *On a class of posets and the corresponding comparability graphs*, Journal of Combinatorial Theory, Series B, vol. 24, pp. 125–133, 1978.
13. H. Lerchs, *On cliques and kernels*, Technical Report, Dept. of Comput. Sci, Univ. of Toronto, 1971.
14. C. Retoré, *Pomset logic as a calculus of directed cographs*, Proceedings of the 4th Roma Workshop: Dynamic perspectives in Logic and Linguistics, CLUEB, pp. 221–247, 1999.
15. E. Sopena, *Oriented graph coloring*, Discrete Mathematics, vol. 229, pp. 359–369, 2001.
16. P. Sumner, *Dacey graphs*, Journal of Aust. Soc., vol. 18, pp. 492–502, 1974.