Arc-disjoint paths in decomposable digraphs

Jørgen Bang-Jensen * Alessandro Maddaloni †

June 25, 2012

Abstract

We prove that the weak \( k \)-linkage problem is polynomial for every fixed \( k \) for totally \( \Phi \)-decomposable digraphs, under appropriate hypothesis on \( \Phi \). We then apply this and recent results by Fradkin and Seymour (on the weak \( k \)-linkage problem for digraphs of bounded independence number or bounded cut-width) to get polynomial algorithms for some class of digraphs like quasi-transitive digraphs, extended semicomplete digraphs, locally semicomplete digraphs and directed cographs.

Keywords: weak linkages, modular partition, decomposable digraph, locally semicomplete digraph, quasi-transitive digraph, arc-disjoint paths, cut-width.

1 Introduction

Notation not given below is consistent with [2]. Note that in this paper we allow both parallel arcs and loops and (for simplicity) we still use the name digraph rather than directed pseudograph.

Paths and cycles are always directed unless otherwise specified. For a digraph \( D \) we denote by \( V(D) \) and \( A(D) \) the set of vertices, respectively, the set of arcs of \( D \) (or just \( V \) and \( A \) when \( D \) is clear from the context). Given two sets of not necessarily distinct vertices \( V_1, V_2 \subseteq V(D) \), we denote by \( A(V_1, V_2) \) the set of arcs from \( V_1 \) to \( V_2 \). The same notation is used when \( V_1, V_2 \) are subdigraphs.

An \((s,t)\)-path in a digraph \( D \) is a directed path from the vertex \( s \) to the vertex \( t \). A digraph \( D = (V,A) \) is strongly connected (or just strong) if there exists an \((x,y)\)-path and a \((y,x)\)-path in \( D \) for every choice of distinct vertices \( x, y \) of \( D \), and \( D \) is \( k \)-arc-strong if \( D \) \( \setminus X \) is strong for every subset \( X \subseteq A \) of size at most \( k - 1 \).

The underlying graph of a digraph \( D \), denoted \( U(G) \), is obtained from \( D \) by suppressing the orientation of each arc and replacing multiple edges by one edge. A digraph \( D \) is connected if \( U(G) \) is a connected graph.

Given a subdigraph, or a subset of arcs, \( F \) and a set of vertices \( U \), we denote by \( N^+(F)(U) \) (\( N^-(F)(U) \)) the out-neighborhood (in-neighborhood) of \( U \) restricted to \( F \), we denote by \( d^+(F)(U) \) (\( d^-(F)(U) \)) the out-degree (in-degree) of \( U \) restricted to \( F \). For \( u, v \in V(D) \) we denote by \( \mu_D(uv) \) the number of arcs from \( u \) to \( v \), i.e. the multiplicity of the arc \( uv \) in \( D \).

A digraph is called semicomplete when there is an arc between every two distinct vertices. A semicomplete digraph without 2-cycles is called a tournament.

Let \( D = (V,A) \) be a digraph and let \( s_1, \ldots, s_k, t_1, \ldots, t_k \) be a collection of (not necessarily distinct) vertices of \( D \). A weak \( k \)-linkage from \( (s_1, \ldots, s_k) \) to \( (t_1, \ldots, t_k) \) is a collection of \( k \) arc-disjoint paths \( P_1, \ldots, P_k \) such that \( P_i \) is an \((s_i, t_i)\)-path or a proper cycle containing \( s_i = t_i \) for each \( i \in [k] \).

The weak \( k \)-linkage problem is the following. Given a digraph \( D = (V,A) \) and not necessarily distinct vertices \( s_1, \ldots, s_k, t_1, \ldots, t_k \); decide whether \( D \) contains a weak \( k \)-linkage from \( (s_1, \ldots, s_k) \) to \( (t_1, \ldots, t_k) \). It is well-known that the weak \( k \)-linkage problem is NP-complete already when \( k = 2 \) [5].

Until very recently, the only non-trivial class of digraphs for which the weak \( k \)-linkage problem was known to be polynomial for every fixed \( k \) was the class of acyclic digraphs, for which there was an

*Department of Mathematics and Computer Science, University of Southern Denmark, Odense DK-5230, Denmark (email: jbj@imada.sdu.dk).
†Department of Mathematics and Computer Science, University of Southern Denmark, Odense DK-5230, Denmark (email: maddaloni@imada.sdu.dk)
Theorem 1.1. [5] The weak $k$-linkage problem is polynomially solvable for every fixed $k$ when the input is an acyclic digraph.

Even for the class of tournaments a polynomial algorithm was only known for the case of $k = 2$ [1]. The following results due to Fradkin and Seymour change that drastically.

Theorem 1.2 (Fradkin-Seymour). [6] The weak $k$-linkage problem is polynomial for every fixed $k$, when we consider digraphs that are obtained from a semicomplete digraph by replacing some arcs with multiple copies of those arcs and adding any number of loops.

Theorem 1.3 (Fradkin-Seymour). [6] For every natural number $\alpha$ the weak $k$-linkage problem is polynomial for every fixed $k$, when we consider digraphs with independence number at most $\alpha$.

Theorem 1.4 (Fradkin-Seymour). [6] For every natural number $\theta$ the weak $k$-linkage problem is polynomial for every fixed $k$, when we consider digraphs with cutwidth at most $\theta$.

The following easy consequence will be used in our algorithms.

Theorem 1.5. For every natural number $p$ the weak $k$-linkage problem is polynomial, for every fixed $k$, when we consider digraphs with at most $p$ directed cycles.

Proof. Let $D$ be a digraph with at most $p$ directed cycles. Then the cutwidth of $D$ is at most $p$: we may delete an arbitrary arc from each of the at most $p$ cycles to get a digraph with cutwidth 0, so $D$ has cutwidth at most $p$. Now the claim follows from Theorem 1.4.

Assume we want to decide the existence of a weak $k$-linkage from $(s_1, \ldots, s_k)$ to $(t_1, \ldots, t_k)$. We will denote by $\Pi$ the list of pairs $^2$ $(s_1, t_1), \ldots, (s_k, t_k)$. In the rest of the paper we will think of $\Pi$ both as a list of $k$ pairs and as a collection of all the terminals $s_1, \ldots, s_k, t_1, \ldots, t_k$.

We say that $D$ has a weak $\Pi$-linkage if it contains a weak $k$-linkage from $(s_1, \ldots, s_k)$ to $(t_1, \ldots, t_k)$, we sometimes also say that $(D, \Pi)$ has a weak linkage.

Given a subdigraph $H$ of $D$, the contraction of $H$ into the vertex $h$ is the digraph $D/H$ with vertex set $V(D \setminus H)$, plus a new vertex $h$ and, for all $u, v \in V(D \setminus H)$, $\mu_{D/H}(uv) = \mu_D(uv)$ and $\mu_{D/H}(hv) = \sum_{y \in V(H)} \mu_D(yv)$. Given vertex disjoint subdigraphs $H_1, \ldots, H_s$ of $D$, the contraction of $H_1, \ldots, H_s$ in $D$ is the digraph $((D/H_1)/\ldots)/H_s$. Clearly the resulting digraph does not depend on the order of $H_1, \ldots, H_s$.

Let $v_1, \ldots, v_n$ be the vertex set of $D$ and let $H_1, \ldots, H_n$ be digraphs which are pairwise vertex disjoint. The composition $D[H_1, \ldots, H_n]$ is the digraph with vertex set $\cup_{i=1}^n V(H_i)$ and arc set $(\cup_{i=1}^n A(H_i)) \cup \{h_i h_j | h_i \in V(H_i), h_j \in V(H_j), v_i, v_j \in A(D)\}$. Even this operation does not depend on the order of $H_1, \ldots, H_n$. We will use the term blow up of $v_i$ into $K$ (in $D$) meaning the composition $D[v_1, \ldots, v_i-1, K, v_{i+1}, \ldots, v_n]$.

The paper is organized as follows: we start with some reduction result and useful observations. Then in Section 3 we prove the most general result that, roughly, sounds like this: if a digraph $D$ can be decomposed as $D = S[H_1, \ldots, H_s]$ and we have polynomial algorithms for the weak $k$-linkage problem on $S, H_1, \ldots, H_s$, then, in many cases, one can produce a polynomial algorithm also for $D$. In Sections 4 and 6 we apply this result to get polynomial algorithms for the following classes of digraphs, all of which contain the class of tournaments: quasi-transitive digraphs, extended semicomplete digraphs and locally semicomplete digraphs: they can all be decomposed using few classes of digraphs. For some of the latter (like semicomplete digraphs) a polynomial algorithm is known to exist (Theorem

\footnote{The independence number of a digraph $D$ is the cardinality of the largest set $I \subseteq V(D)$ such that $A(D(I)) = \emptyset$.}

\footnote{Note that the same pair (or the same vertex) may appear more than once in the list and we may have $s_i = t_i$.}
1.2); for some other, like round digraphs, we develop a polynomial algorithm in Section 5. In the last section we apply our general result to the class of directed cographs and point out possible further generalizations of some results.

2 Reductions

Recall that we allow multiple arcs (and loops) in our digraphs. We will assume through all the paper, unless otherwise stated, that $k$ stands for the number of pairs to be linked. An instance of the problem $(D, \Pi)$ is equivalent to $(D', \Pi)$ where $V(D') = V(D)$ and for every $u, v \in V(D')$ one has $\mu_{D'}(uv) = \min(\mu_D(uv), k)$. Therefore from now on we will only consider digraphs $D$ with

$$\mu_D(uv) \leq k \quad \forall u, v \in V(D)$$

while studying the weak $k$-linkage problem.

**Definition 2.1.** Let $D = (V, A)$ be a digraph and $H$ an induced subdigraph of $D$. We say that $H$ is a module if for every $a, b \in V(H)$, $v \in V(D \setminus H)$ we have that $\mu_D(va) = \mu_D(vb)$ and $\mu_D(av) = \mu_D(bv)$. We say that $H$ is a clean module with respect to $\Pi$ if it is a module containing no terminals of $\Pi$.

**Definition 2.2.** A digraph $D$ is decomposable if $D = S[H_1, ..., H_s]$, for some digraph$^3$ $S$, with $s = |V(S)| \geq 2$ and some choice of disjoint modules $H_1, ..., H_s$.

Let $\Phi$ be a class of digraphs. We say that $D$ is totally $\Phi$-decomposable if either $D \in \Phi$ or $D = S[H_1, ..., H_s]$ is decomposable with $S \in \Phi$ and $H_i$ totally $\Phi$-decomposable, for $i = 1, ..., s$. We call this decomposition a total $\Phi$-decomposition of $D$.

The algorithms we develop rely on the following fundamental fact: a weak linkage need not use any arc inside clean modules.

**Lemma 2.3.** Let $D$ be a digraph, $\Pi$ a list of $k$ terminal pairs and $H \subset D$ a clean module with respect to $\Pi$. Let $D'$ be the contraction of $H$ into a single vertex $h$. Then $D$ has a weak $\Pi$-linkage if and only if $D'$ has a weak $\Pi$-linkage.

**Proof.** If $(D, \Pi)$ has a weak linkage, then this clearly induces a weak linkage on $(D', \Pi)$.

We will prove by induction on $k$ that if $D'$ has a weak linkage $P_1', ..., P_k'$, then there is a weak linkage $P_1, ..., P_k$ in $D$ such that $P_i' = P_i$ on $D \setminus H$ and $A(P_i) \cap A(H) = \emptyset$ for $i = 1, ..., k$.

Suppose $k = 1$. If the vertex $h$ does not belong to $P_1'$, then $P_1'$, as a path of $D$, is the required weak linkage, otherwise let $u$ (resp. $v$) be the in-neighbor (resp. out-neighbor) of $h$ in $P_1'$, let $a$ be a vertex of $H$. As the arcs $ua, av$ must be in $A(D)$, the path $P_1 := P_1'[s_1, a][P_1'[v, t_1]$ is well defined and is the required weak linkage.

Now suppose $(D', \Pi)$ has a weak linkage $P_1', ..., P_k'$. By induction hypothesis, there is a collection $Q_1, ..., Q_{k-1}$ of arc-disjoint paths in $D$, such that each $Q_i$ links $s_i$ to $t_i$ and is arc-disjoint from $P_k'$, if restricted to $D \setminus H$, because $Q_i = P_i'$ on $D \setminus H$. Hence if $h$ doesn’t belong to $P_k'$, then $Q_1, ..., Q_{k-1}$ together with $P_k'$ (as a path of $D$) form the desired weak linkage, so suppose $h \in P_k'$. Let $Q = \bigcup_{i=1}^{k-1} Q_i$ and $F = A(Q)$. As $h \not\in P_k'$, there must be arcs $ua, bv \in A(D) \setminus F$, with $a, b \in V(H)$.

Let $I = N^+_{Q \setminus H}(\{a, b\})$, $O = N^-_{Q \setminus H}(\{a, b\})$. Observe that if $a = b$ or $a \not\in I$ then $Q_1, ..., Q_{k-1}, P_k := P_k'[s_k, u][P_k'[v, t_k]$ is the desired weak linkage. Similarly if $v \not\in O$ it is easy to construct a weak linkage, so we may assume that $a \neq b$, $u \in I$, $v \in O$.

We want to use either $av$ or $ub$ for the $(s_k, t_k)$-path. To do this, we show that there are arcs from $I$ to $b$ or from $a$ to $O$ that are not in $F$ and use these to redirect some of the paths through $H$, so that either $av$ or $ub$ becomes available.

As $H$ is a module, we have:

$$d^+_F(a) + |A(I, a) \setminus F| = d^+_F(b) + |A(I, b) \setminus F|$$
$$d^+_F(a) + |A(a, O) \setminus F| = d^+_F(b) + |A(b, O) \setminus F|$$

$^3$S is called sometimes the quotient digraph (of $D$) induced by $H_1, ..., H_s$. 

3
Now note that \( d^+_D(a) = d^+_F(a), \) \( d^+_D(b) = d^+_F(b), \) hence \( c := |(A(I, b) \cup A(a, O)) \setminus F| = |(A(I, a) \cup A(b, O)) \setminus F| \geq 2, \) since \( ua, bv \notin F. \)

Let \( A_k = (A(I, b) \cup A(a, O)) \setminus F \neq \emptyset, A_i = A(Q_i), \) for \( i = 1, \ldots, k-1 \) and \( A_0 = \{ua, bv\}, A_{-1} = (A(I, a) \cup A(b, O)) \setminus (F \cup A_0). \) Construct the auxiliary digraph \( \tilde{D} \) with vertex set \( \{-1, 0, 1, \ldots, k-1, k\} \) and arc set as described below: for every \( z \in I, \) enumerate arbitrarily the arcs in \( A(z, a) = \{za_1, \ldots, za_{\mu(za)}\} \) and \( A(z, b) = \{zb_1, \ldots, zb_{\mu(zb)}\} \) and create arcs \( ij \) in \( \tilde{D} \) if and only if, for some \( l, za_l \in A_i \) and \( zb_l \in A_j. \) Similarly, for every \( w \in O, \) enumerate arbitrarily the arcs in \( A(a, w) \) and \( A(b, w) \) and create arcs \( ij \) in \( \tilde{D} \) if and only if, for some \( l, bw_l \in A_i \) and \( aw_l \in A_j. \)

Given that \( A_0 \) has only arcs to \( a \) and from \( b, \) we have \( d^-_D(0) = 0. \) Similarly it is easy to see that \( d^-_D(-1) = d^-_D(0) = 0, d^+_D(0) = 2, d^+_D(k) = c. \) Moreover, for \( i = 1, \ldots, k-1, d^+_D(i) = d^-_D(i) = 1 \) if \( Q_i \cap \{a, b\} \neq \emptyset, d^+_D(i) = d^-_D(i) = 0 \) otherwise. Therefore there exists a path \( 0i_1 \ldots i_k \) in \( \tilde{D} \) with \( ub \in A(Q_{i_j}), b \in Q_{i_j} \) for \( j \) odd, \( a \in Q_{i_{j-1}} \) for \( j \) even. Now we redirect all these paths \( Q_{i_j} \) from \( b \) to \( a \) (from \( a \) to \( b \)), so that we end up having the arcs \( ub, bv \) available for the \( (s_k, t_k) \) path. In other words, let \( z_j (w_j) \) be the in-neighbor (out-neighbor) of \( a \) or \( b \) in \( Q_{i_j}, \) note that \( z_j = z_{j+1} \) for \( j \) even and \( w_j = w_{j+1} \) for \( j \) odd. We define the arc sets of our desired weak linkage as follows:

\[ A_{i_j} := A(Q_{i_j}) \cup \{za, aw_j\} \setminus \{z_j, a, w_j\} \text{ for } j \text{ even, } \]
\[ A_{i_j} := A(Q_{i_j}) \cup \{za, aw_j\} \setminus \{z_j, b, bw_j\} \text{ for } j \text{ odd, } \]
\[ A_k := A(P'_k) \cup \{ub, bv\} \setminus \{ub, hv\} \text{ and } A_i := A(Q_i) \text{ for all the other indices. It is easy to see that these paths are well defined and arc-disjoint.} \]

The proof of Lemma 2.3 has an immediate consequence:
Corollary 2.4. Let $\Pi$ be a list of terminal pairs and $H \subset D$ be a clean module with respect to $\Pi$. For every weak linkage $P'_1, \ldots, P'_k$ of $(D, \Pi)$, there exists another weak linkage $P_1, \ldots, P_k$ such that $P'_i = P_i$ on $D \setminus H$, and for $i = 1, \ldots, k$, $A(P_i \cap H) = \emptyset$.

Definition 2.5. Given a decomposable digraph $D = S[H_1, \ldots, H_s]$ and a path $P$ we say that $P$ is internal if $P \subseteq H_j$ for some $H_j$, we say that $P$ is external otherwise. Similarly, we say that a pair $(s, t)$ is an internal pair if $s, t \in H_j$ for some $j$, we say that $(s, t)$ is an external pair otherwise.

If a module $H$ is not clean, i.e. it contains terminals, then some of the arcs in $A(H)$ may be necessary to guarantee a weak linkage. See Figure 2. The following lemma shows that, in a precise sense, a weak linkage need not use too many arcs inside a given module. Together with Corollary 2.4, this will allow a polynomial brute-force (Theorem 3.2).

For technical reasons that will become clear later, we consider the more general case where a set of arcs $F$ has been deleted from $D$.

Lemma 2.6. Let $D = S[H_1, \ldots, H_s]$ be a decomposable digraph, let $\Pi'$ be a list of $h$ terminal pairs and let $F$ be a set of arcs in $D$ satisfying that $d^+_F(v), d^-_F(v) \leq r$ for all $v \in V(D)$. If $(D \setminus F, \Pi')$ has a weak linkage, then it has a weak linkage $P_1, \ldots, P_h$ such that, we have $|V(\bigcup_{i} P_i \cap H_j)| \leq 2h(h + r)$, for every $j$, where $E$ denotes the set of indices $i$ for which $P_i$ is external.

Proof. We will prove that for every $j \in \mathcal{E}$ and every $i \in E$, $|V(P_i \cap H_j)| \leq 2h(r + h)$, which implies our claim.

Consider a solution $P_1, \ldots, P_h$ that minimizes $|V(P_1)| + |V(P_2)| + \ldots + |V(P_h)|$ and suppose, by contradiction, that for some $j$ and $i \in E$, one has $|V(P_i \cap H_j)| > 2h(r + h)$. As $P_i$ is not internal, there must then exist a vertex $a \in P_i \setminus H_j$ such that either $a$ is an in-neighbor of $H_j$ and $V' := V(P_i[a, t_i] \cap H_j)$ has more than $h + r$ vertices, or $a$ is an out-neighbor of $H_j$ and $V'' := V(P_i[s_i, a] \cap H_j)$ has more than $h + r$ vertices. Suppose, without loss of generality that the latter holds. Then at least one arc of the form $ua$, with $a \in V''$, will not be used by any path $P_i$ and will not be in $F$, because there are more than $h + r$ arcs of the form $ua$ and at most $r$ of these can belong to $F$ and at most $h$ to the current linking. Hence the path $P'_i := P_i[s_i, u]P_i[a, t_i]$ uses fewer vertices than $P_i$ and is still arc-disjoint from the other paths: contradiction.

Note that from the previous proof we have that for every $j = 1, \ldots, s$ and every $i \in E$, $|A(P_i \cap H_j)| < 2h(r + h)$.

Lemma 2.7. Let $C$ be a class of digraphs for which there exists an algorithm $A$ to decide the weak $k$-linkage problem, whose running time is bounded by $f(n, k)$. Let $D = (V, A)$ be a digraph, $\Pi$ a list of $k$ pairs of terminals and $F \subseteq V \times V$ such that $D' := (V, A \cup F)$ is a member of $C$. There exists an algorithm $A^-$, whose running time is bounded by $f(n, k + |F|)$, to decide whether $D$ has a weak $\Pi$-linkage.

Proof. Suppose $F = \{s'_1t'_1, \ldots, s'_kt'_k\}$, where $k' = |F|$ and let $\Pi' = (s'_1t'_1, \ldots, s'_kt'_k)$. $D$ has a weak $\Pi$-linkage if and only if $D'$ has a weak $(\Pi \cup \Pi')$-linkage, from which the claim follows. Indeed if $D$ has

---

\[\text{Figure 2: An example with } |\Pi| = 2, \text{ the only weak } \Pi\text{-linkage uses the arc inside } H_1.\]
a weak \( \Pi \) linkage, then this extends to a weak \((\Pi \cup \Pi')\)-linkage of \( D' \) by simply taking the arcs \( s'_it'_i \)
as \((s'_it'_i)\)-paths. If \( D' \) has a weak \((\Pi \cup \Pi')\)-linkage \( L \), it is easy to see that \( A(L) \setminus F \) contains a weak
\( \Pi \)-linkage of \( D \). \( \Box \)

3 The main algorithm

Given a digraph \( D \) and a non negative integer \( c \), let \( D(c) \) denote the set of digraphs that can be
obtained from \( D \) by first adding any number of arcs parallel to the already existing ones and then
blowing up \( b \) vertices, with \( 0 \leq b \leq c \), to digraphs of size less than or equal to \( c \) each.

**Definition 3.1.** We say that a class of digraphs \( \Phi \) is **bombproof** if there exists a polynomial algo-

rithm \( A_\Phi \) to find a total \( \Phi \)-decomposition of every totally \( \Phi \)-decomposable digraph and, for every

integer \( c \), there exists a polynomial algorithm\(^5\) \( B_\Phi \) to decide the weak \( k \)-linkage problem for the class

\[
\Phi(c) := \bigcup_{D \in \Phi} D(c).
\]

From now until the end of this section \( \Phi \) will denote a bombproof class of digraphs.

**Theorem 3.2.** There is a polynomial algorithm \( M \) which takes as input a \( 5 \) tuple \([D,k,k',\Pi,F]\),

where \( D \) is a totally \( \Phi \)-decomposable digraph, \( k,k' \) are natural numbers with \( k' \leq k \), \( \Pi \) is a list of \( k' \)
terminal pairs and \( F \subseteq A(D) \) is a set of arcs satisfying

\[
d_F^-(v), d_F^+(v) \leq k - k' \quad \text{for all } v \in V(D),
\]

\[
|F| \leq (k - k')2k
\]

and decides whether \( D \setminus F \) contains a weak \( \Pi \)-linkage.

**Proof.** The following is a description of the polynomial algorithm \( M \).

1. If \( \Pi = \emptyset \) output that a solution exists and return.
2. Run \( A_\Phi \) to find a total \( \Phi \)-decomposition of \( D = S[H_1,...,H_s] \).
3. If this decomposition is trivial, that is \( D = S \), then \( D \in \Phi \subseteq \Phi(1) \), so run \( B_\Phi \) on \((D \setminus F,\Pi)\) to
decide the problem and return.
4. Find among \( H_1,...,H_s \) those modules \( K_1,...,K_l \) that contain at least one terminal. Let \( D' \) be
obtained contracting all the modules distinct from \( K_1,...,K_l \). Let \( F' \) be the set of arcs obtained
from \( F \) after the contraction.
5. Let \( \Pi^c \subseteq \Pi \) (\( \Pi^i \subseteq \Pi \)) be the list of external (internal) pairs \((s_q,t_q)\) in \( \Pi \).
6. For every partition of \( \Pi^i = \Pi_1 \cup \Pi_2 \) look for external paths linking the pairs in \( \Pi^c \cup \Pi_1 \) and
internal paths linking the pairs in \( \Pi_2 \). This is done in the following way:

   (a) If \( \Pi^c \cup \Pi_1 = \emptyset \), then for \( i = 1,...,l \) run \( M \) recursively on input \([K_i,k,k',\Pi \cap K_i,F \cap A(K_i)]\),
where \( \Pi \cap K_i \) denotes the list of terminal pairs that lie inside \( K_i \) and \( k_i' \) is the number of
those pairs.
   (b) If \( \Pi^c \cup \Pi_1 \neq \emptyset \), let \( k_i' \) be the number of pairs in \( \Pi_2 \cap K_i \). We do the following for each possible
choice of \( l \) vertex sets \( W_i \subseteq V(K_i), \) \( i = 1,...,l \) of size \( \min(|V(K_i)|,2[k' - k_i')(k - k')] \) and
arc sets\(^6\) \( F_i \subseteq A(K_i(W_i)) \setminus F, \) \( i = 1,...,l \), with \( F_i \) satisfying

   \[
   d^-_{(F \cap A(K_i)) \cup F_i}(v), d^+_{(F \cap A(K_i)) \cup F_i}(v) \leq k' - k_i',
   \]

   \[
   |F_i| \leq 2(k' - k_i')(k - k')
   \]

\(^5\)Note that the running time of \( B_\Phi \) may depend heavily on \( c \).
\(^6\)\( K_i(W_i) \) is the subdigraph of \( K_i \) induced by \( W_i \).
• For every module $K_i$ remove all the vertices of $V(K_i) \setminus W_i$ and then all remaining arcs except those in $F_i$.

• Define $D''$ to be the digraph obtained from $D'$ with this procedure.

• Run $B_\Phi$ on $(D'' \setminus F', \Pi'' \cup \Pi_1)$.

• For $i = 1, \ldots, l$, run $\mathcal{M}$ recursively on input $[K_i, k, k', \Pi_2 \cap K_i, (F \cap A(K_i)) \cup F_i]$.

If at step 6(a) all the instances examined are linked or at step 6(b) there is a choice of $W_i, F_i$, $i = 1, \ldots, l$, such that all instances examined are linked, then output that a weak linkage exists and return.

7. If all choices of $\Pi_1, \Pi_2$ have been considered, without verifying the existence of any weak linkage, then output that no weak linkage exists.

Let us prove by induction on the size of $D$ that the algorithm $\mathcal{M}$ is correct:

If $D \in \Phi$, then clearly $D \in \Phi(1)$ and so, by Lemma 2.7, the correctness follows from the correctness of the algorithm $B_\Phi$. Hence assume that $D \notin \Phi$.

Suppose that there exists a weak $\Pi$-linkage in $D \setminus F$ consisting of $k'$ arc-disjoint paths of which $k'' \leq k'$ are external paths forming a weak $\Pi''$-linkage for a sublist $\Pi''$ of $\Pi$. By Lemma 2.6, there exists a weak $\Pi$-linkage $L$ in $D \setminus F$ whose $k''$ external paths use at most $2k''(k'' + (k - k')) \leq 2kk''$ arcs inside every module. Recall that all the modules that do not contain any terminal are clean modules.

Hence by Corollary 2.4, we can get from $L$ a weak $\Pi$-linkage $L'$ such that no path in $L'$ uses arcs inside the clean modules and $L'[D \setminus \bigcup_{i=1}^l K_i] \equiv L'$.7 Hence $L'$ is such that its external paths use only a subset of arcs $F'_i \subseteq A(K_i)$, inside $K_i$ with $|F'_i| \leq 2kk''$, for $i = 1, \ldots, l$ and no arcs inside the other modules. Note that, since every arc in these $F'_i$ lies on one of the $k''$ external paths, we have $d_{F_i \cup F'_i}(v), d_{F_i \cup F'_i}(v) \leq (k - k') + k'' = k - (k' - k'')$ for every $v \in V(K_i)$. Thus if there are $k'_i$ internal paths of $L'$ inside $K_i$, then, given that $k'_i \leq k''$, we have

$$d_{F_i \cup F'_i}(v), d_{F_i \cup F'_i} \leq k - (k' - k'') \leq k - k'_i \quad \text{for } i = 1, 2, \ldots, l$$

and

$$|F \cap A(K_i)| \cup F_i \leq |F| + |F_i| \leq 2(k - k') + 2k(k'') \leq 2(k - k') + 2k(k' - k'_i) = 2k(k - k'_i).$$

implying that the conditions (2) hold for all the tuples $[K_i, k, k'_i, \Pi_2 \cap K_i, (F \cap A(K_i)) \cup F_i]$, $i = 1, \ldots, l$.

Suppose that the algorithm $\mathcal{M}$ does not output that a solution exists before examining the list $\Pi'' \cup \Pi_2$ of $k''$ external pairs and the arc sets $F_1, \ldots, F_l$ corresponding to the linkage above. At this point, if $k'' = 0$, and thus $F_1, \ldots, F_l = \emptyset$, then step 6(a) is entered and, by induction hypothesis, the algorithm will verify in each $K_i \setminus (F \cap A(K_i))$ the existence of a linkage for the $k'_i$ internal pairs that are linked inside $K_i$ by $L'$. If $k'' > 0$, step 6(b) is entered and the digraph $D''$ constructed. Note that $D'' \in \Phi(2k'')$, hence, by the correctness of the algorithm $B_\Phi$ of Lemma 2.7, our algorithm will establish the existence of a weak $\Pi''$-linkage in $D'' \setminus F'$ (there is at least one induced by the $k''$ external paths of $L'$). By induction hypothesis our algorithm will also find in each $K_i \setminus ((F \cap A(K_i)) \cup F_i)$ a linkage for the $k'_i$ internal pairs that are internally linked in $K_i$ by $L'$. Indeed, by (5) and (6) all the recursive calls are made on valid instances of the algorithm. Therefore at the end of step 6 the algorithm will conclude correctly that $D$ has a weak $\Pi$-linkage.

Vice versa, if the algorithm outputs that a solution exists, it means that there is a choice of $\Pi_1, \Pi_2$ and of subsets of arcs $F_i \subseteq A(K_i) \setminus F$ such that $[K_i, k, k'_i, \Pi_2 \cap K_i, (F \cap A(K_i)) \cup F_i]$ is a yes instance, for $i = 1, \ldots, l$. Thus, by induction, these are all linked.

Moreover, if $\Pi'' \cup \Pi_1 \neq \emptyset$, then, with the choice of $F_1, \ldots, F_l$ above, $(D'' \setminus F', \Pi'' \cup \Pi_1)$ is linked, because it has been verified at step 6(b) with the algorithm $B_\Phi$. By Lemma 2.3, we have that $(D \setminus F, \Pi'' \cup \Pi_1)$ is also linked and it has a weak linkage that inside any module uses only arcs from the set $\bigcup_{i=1}^l F_i$.

This weak linkage and the previous ones are arc-disjoint by construction, therefore together they form a weak $\Pi$-linkage of $D \setminus F$.  

\footnote{\textit{L$'$}\textit{[D \setminus \bigcup_{i=1}^l K_i]} stands for the linkage $L'$ restricted to the digraph $D \setminus \bigcup_{i=1}^l K_i$.}

7
Let \( T(n, k, k') \) be an upper bound for the running time of \( \mathcal{M} \) when the input is a digraph of order \( n \) with a list of terminal pairs of size \( k' \leq k \). We will prove by induction on \( n + k' \) that \( T(n, k, k') \) is \( O(n^{c(k)}) \), for some fixed \( c(k) \).

If \( n + k' = 0 \) the running time is constant, as there is nothing to do.

By definition of \( \Phi \) step 2 is polynomial, and by Lemma 2.7 step 3 is also polynomial, step 4 is clearly polynomial. Let \( b(k) \) be a (non decreasing) function of \( k \) such that running step 2, 3 and 4 takes time \( O(n^{b(k)}) \). At step 6, for each of the \( O(2^k) \) possible choices of \( \Pi_1, \Pi_2 \), either step 6(a) or 6(b) is entered.

Step 6(a) takes time bounded by \( B_a := \sum_i T(n_i, k, k'_i) \), with \( n_i = |V(K_i)| \) and \( \sum_i n_i = n \), \( \sum_i k'_i = k' \).

If step 6(b) is entered, for every possible choice of \( W_1, \ldots, W_t, F_1, \ldots, F_t \) the digraph \( D'' \backslash F' \) is constructed and the algorithm \( B''_n \) is run on it. Now for every recursive call on input \( [D, k, k', \Pi, F] \), the set \( F \) is such that \( |F| \leq (k - k')2k \leq 2k^2 \), hence \( |F'| \leq 2k^2 \). Therefore, by Lemma 2.7, \( B''_n \) always runs in polynomial time. The construction of \( D'' \backslash F' \) is also polynomial. Let \( d(k) \) be a (non decreasing) function of \( k \) such that executing \( B'_{n} \) and constructing \( D'' \backslash F' \) takes time \( O(n^{d(k)}) \). Additionally some recursive calls are made. Note also that \( l \leq 2k' \leq 2k \) and \( |F_i| \leq 2k^2 \), so the number of possible choices of \( W_1, \ldots, W_t, F_1, \ldots, F_t \) is at most \((\frac{n}{2k^2}) \cdot (\frac{4k^3}{2k^2})^{2k} = O(n^{4k^3})\). We can therefore conclude that the running time of step 6(b) is bounded by \( B_b := O(n^{4k^3}) \sum_{i=1}^{t} T(n_i, k, k'_i) + O(n^{d(k)}) \), with \( \sum_i n_i = n \), \( \sum_i k'_i = k' \), and \( k'_i < k' \) for every \( i \). Thus \( B_b \leq O(n^{4k^3}) \sum_{i} (2k \cdot T(n, k - 1, k' - 1) + O(n^{d(k)})) \).

We will show that our claim is true with \( c(k) := 10^k + d(k) + b(k) \). By the induction hypothesis we have

\[
B_{a} \leq \sum_{i} O(n_{i}^{c(k_{i})}) + O(n^{d(k)}) = O\left(\sum_{i} n_{i}^{c(k_{i})}\right) + O(n^{d(k)}) = O(n^{c(k)}). 
\]

Moreover

\[
B_{b} \leq O(n^{4k^3}) \cdot 2k(T(n, k - 1, k' - 1) + O(n^{d(k)})) = O(n^{c(k - 1) + 4k^3}) + O(n^{d(k) + 4k^3}),
\]

The last term is \( O(n^{c(k)}) \), since \( d(k - 1) + b(k - 1) \leq d(k) + b(k) \) and \( 4k^3 + 10^{k - 1} \leq 10^k \) for every positive \( k \) and so \( c(k) \geq c(k - 1) + 4k^3 \) and \( c(k) \geq d(k) + 4k^3 \). Finally we have that the total running time:

\[
T(n, k, k') \leq O(2^k)(B_{a} + B_{b}) + O(n^{b(k)}) = O(n^{c(k)}).
\]

Taking \( k' = k \) running the previous algorithm on input \( [D, k, k, \Pi, \emptyset] \) where \( D \) is any totally \( \Phi \)-decomposable digraph and \( \Pi \) is a list of \( k \) terminal pairs from \( V(D) \), we obtain the main result of this section.

**Theorem 3.3.** For every fixed \( k \) there exists a polynomial algorithm for the weak \( k \)-linkage problem for the totally \( \Phi \)-decomposable digraphs.

### 4 Quasi-transitive and extended semicomplete digraphs

We turn now to quasi-transitive digraphs, which generalize both transitive and semicomplete digraphs.

**Definition 4.1.** A digraph \( D \) is **quasi-transitive** if, for every triple of distinct vertices \( x, y, z \in V(D) \), with \( xy, yz \in A(D) \), there is at least one arc between \( x \) and \( z \).

We will use the following result from [3] to get the so called **canonical decomposition** of a quasi-transitive digraph.

**Theorem 4.2.** [3] Let \( D \) be a quasi-transitive digraph.

1. If \( D \) is not strong, then there exist a transitive acyclic digraph \( T \) on \( t \) vertices and strong quasi-transitive digraphs \( H_1, \ldots, H_t \) such that \( D = T[H_1, \ldots, H_t] \).

2. If \( D \) is strong, then there exist a strong semicomplete digraph \( S \) on \( s \) vertices and quasi-transitive digraphs \( Q_1, \ldots, Q_s \) such that each \( Q_i \) is either a single vertex or is non-strong and \( D = S[Q_1, \ldots, Q_s] \).
Moreover one can find the above decompositions in polynomial time.

Based on this recursive structure we want to show that there is a polynomial algorithm for the weak $k$-linkage problem on quasi-transitive digraphs. Let $\Phi_1$ be the union of all acyclic digraphs and all semicomplete digraphs. Stated in other words, Theorem 4.2 says that the quasi-transitive digraphs are totally $\Phi_1$-decomposable.

**Lemma 4.3.** The class $\Phi_1$ is bombproof

**Proof.** We can get a polynomial algorithm for the total $\Phi_1$-decomposition easily from a result in [2, Section 2.11], where a polynomial algorithm is given for the class of all acyclic and all semicomplete multipartite digraphs. Given a positive integer $c$ and a digraph $D \in \Phi_1$, consider a digraph in $D' \in D(c)$: if $D$ is semicomplete, then $D'$ misses no more than $c^3$ arcs to be semicomplete. If $D$ is acyclic, then $D'$ has at most $O(c^{e+1})$ cycles or $O(c \cdot (ck)^c)$ in case there are (at most $k$) parallel arcs, because all the cycles must lie in one of the blown up subdigraphs. By Theorem 1.2 and Lemma 2.7 in the first case and Theorem 1.5 in the second case, there is a polynomial algorithm to decide the weak $k$-linkage problem in $D(c)$ and hence in $\Phi_1(c)$. Thus we can conclude that $\Phi_1$ is bombproof.

**Theorem 4.4.** For every fixed $k$ there exists a polynomial algorithm for the weak $k$-linkage problem for quasi-transitive digraphs.

**Proof.** By Theorem 4.2 quasi-transitive digraphs are totally $\Phi_1$-decomposable. By Lemma 4.3 $\Phi_1$ is bombproof, hence we can apply Theorem 3.3.

We can apply Theorem 3.3 to another class of digraphs: extended semicomplete digraphs.

**Definition 4.5.** A digraph $D$ is extended semicomplete if $D = S[H_1, ..., H_s]$, where $S$ is a semicomplete digraph and $H_1, ..., H_s$ are independent sets of vertices.

Extended semicomplete digraphs are clearly totally $\Phi_1$-decomposable. Hence, from Theorem 3.3, we have the following

**Theorem 4.6.** For every fixed $k$ there exists a polynomial algorithm for the weak $k$-linkage problem for extended semicomplete digraphs.

5 Round digraphs

In order to study the weak $k$-linkage problem on locally semicomplete digraphs, we focus on a subclass of these digraphs, namely round digraphs. These have nice properties and generalize, for example, powers of cycles.

**Definition 5.1.** A digraph on $n$ vertices is **round** if we can label its vertices $v_1, ..., v_n$ so that for each $i$, we have $N^+(v_i) = \{v_{i+1}, ..., v_{i+\delta(i)}\}$ and $N^-(v_i) = \{v_{i-\delta'(i)}, ..., v_{i-1}\}$. We call the labeling $v_1, ..., v_n$ a round ordering.\(^8\)

**Definition 5.2.** Given a round ordering $O = v_1, ..., v_n$ we say that $w_1, ..., w_n$ is the $i$-th **reverse** ordering if, for $j = 1, ..., n$, $w_j = v_{n-j+1}$ (the indices related to an ordering of the vertices of a round digraph are always meant modulo $n$).

We say that a round digraph, with a round ordering $O$ has **round cutwidth** at least $\theta$ if the $i$-th reverse ordering (with respect to $O$) has cutwidth at least $\theta$, for $i = 1, ..., n$.

We prove that the weak $k$-linkage problem is polynomially solvable for round digraphs. The first step is to prove that a round digraph with large round cutwidth has a weak $\Pi$-linkage for every list $\Pi$ of $2k$ distinct terminals.

Note that large cutwidth doesn’t necessarily imply high arc-connectivity, even for round digraphs. For example consider the round digraph with vertices $v_1, ..., v_n$ and arcs given by $\{v_iv_j\} 1 \leq i < j \leq n$.

\(^8\)Round digraphs can be recognized (and a round ordering provided) in polynomial time [2, section 2.10.1], [7].
We have that

**Proof.**

By Lemma 5.3 we conclude that

\[ |N^+(v_{i+1})| \leq \sqrt{\frac{\theta}{k}}. \]

Consider the i-th reverse ordering: as the digraph is round, the arcs that contribute to the cutwidth of this ordering are exactly those from \( N^-(v_{i+1}) \) to \( N^+(v_i) \). Let us recall that, as stated previously, our digraphs never have more than \( k \) parallel arcs, therefore it is easy to see that

\[ |A(N^-(v_{i+1}), N^+(v_i))| \leq k^2. \]

Hence we have:

\[ \theta \leq |A(N^-(v_{i+1}), N^+(v_i))| \leq k^2 \]

from which the statement follows. \( \square \)

Note that Lemma 5.3 and the property of round digraph imply that for every \( i \) we have that

\[ |N^+(v_i)| \geq \sqrt{\frac{\theta}{k}} - 1 \text{ or } |N^-(v_i)| \geq \sqrt{\frac{\theta}{k}}. \]

Given a subset of vertices \( U \), with \( |U| \leq h \), we want, for reasons that will become clear later, a linkage that uses (almost) no arc inside \( U \). We show that this is possible if the cutwidth is larger than \( k(6k + 3h)^2 \). Let us first give some nicknames that will be useful below.

- We call a vertex \( v \) **useful** (with respect to \( U \)) if \( |N^+(v)| \geq 6k + 3h \).

- Given a round ordering \( O = v_1, ..., v_n \), we call an arc \( v_i v_j \) a **maximal jump**, if \( v_i v_{j+1} \notin A(D) \).

**Lemma 5.4.** Let \( D \) be a round digraph, with round ordering \( v_1, ..., v_n \). Let \( U \subseteq V(D) \) with \( |U| \leq h \) and assume the round cutwidth of \( D \) is at least \( k(6k + 3h)^2 \). If \( v_i \) is not useful (with respect to \( U \)) and \( v_iv_j \) is a maximal jump, then \( v_j \) is useful.

**Proof.** We have that \( N^+(v_{j+1}) \subseteq N^+(v_i) \) because \( D \) is round and \( v_i v_j \) is a maximal jump. Hence, given that \( v_i \) is not useful, we have that

\[ |N^+(v_{j+1})| \leq |N^+(v_i)| < 6k + 3h \]

By Lemma 5.3 we conclude that \( |N^+(v_{j+1})| \geq 6k + 3h \) \( \square \)

**Lemma 5.5.** Let \( D \) be a round digraph with round ordering \( v_1, ..., v_n \), let \( \Pi \) be a list of \( k \) distinct terminal pairs and \( U \subseteq V(D) \) such that \( |U| \leq h \). If the round cutwidth of \( D \) is at least \( k(6k + 3h)^2 \), then \((D, \Pi)\) has a weak linkage \( P_1, ..., P_k \) such that, for every \( i \), all the arcs in \( A(P_i) \cap A(U) \) start in \( s_i \) or end in \( t_i \).

**Proof.** Let \( l = \lfloor \frac{n}{2k + h} \rfloor - 1 \) and let us divide the vertices into segments \( V_0, V_1, ..., V_{l-1} \) such that

\[ V_j = \{v_{(2k+j)i+1}, ..., v_{(2k+j)i+1 + 1}, ..., v_{(2k+h)(j+1)}\}, \]

for \( j = 0, 1, ..., l - 2 \) and \( V_{l-1} = \{v_{(2k+h)(l-1)+1}, ..., v_n\} \). Let us color the vertices of \( D \) in such a way that \( s_i, t_j \) have color \( i \), the vertices of \( U \setminus \Pi \) have color 0 and each segment contains at least one vertex of color \( i \), for \( i = 1, ..., k \). This is possible because every segment has at least \( 2k + h \) vertices.

Let \( P_1, ..., P_k \) be such that \( P_i \) is a \((s_i, t_i)\)-path that always uses an arc \( v \) to the next vertex which is colored \( i \) and if no such arc exists, it follows a maximal jump arc.

Note that if \( v \in V_j \) is useful, then \( N^+(v) \supseteq V_{j+1} \), hence there is an arc from \( v \) to a vertex which is colored \( i \). If \( v \) is not useful, then, by Lemma 5.4, the maximal jump arc goes to a useful vertex.

Therefore, given that \( s_i \) has color \( i \), the vertices of \( P_i \) have color \( i \) or are useful.

\footnote{This is an arc \( v_j v_{j+1} \) such that \( v_j + h \) has color \( i \) and no other vertex among \( v_{j+1}, ..., v_{j+h-1} \) has color \( i \).}
Figure 3: The sets $L_i, I_i, R_i$ in the construction of $D_{\Pi}$. In this example $|\Pi| = 2$ and the arcs are omitted.

The paths $P_1, \ldots, P_k$ are arc-disjoint because the only vertices they possibly share are useful vertices, but the head of an arc starting in an useful vertex on path $P_i$ has color $i$, therefore the paths cannot share any arc. Moreover it is easy to see that each arc of $P_i$ has at least one endpoint of color $i$, hence the only arcs that $U$ and $P_i$ could possibly share are arcs from $s_i$ or arcs to $t_i$, given that all the other vertices in $U$ do not have color $i$.

Let us define the quantity $\Theta = k(6k + 36k^2(2k + 1))^2$, which we are going to use for the rest of the section. Below $D$ is a round digraph with round ordering $O = v_1, \ldots, v_n$ and cutwidth at least $\Theta$ and $\Pi$ is a list of $k$ pairs of terminals. We describe how to construct another digraph $D_{\Pi}$, which we will call the compression of $D$ with respect to $\Pi$. This digraph is of constant size (depending only on $k$), but it has enough information to decide the problem.

In the description that follows the relations between the vertices are with respect to the round ordering, so, for example, by an interval we mean a subset of vertices $v_i, v_{i+1}, \ldots, v_{j-1}, v_j$, which are consecutive in the round ordering and by next vertex on the left of $v_i$ we mean $v_{i-1}$.

Let us consider disjoint intervals $I_1, \ldots, I_l$, with $\Pi \subseteq \bigcup I_i$, so that the left end vertex of each $I_i$ is a terminal such that the next $6k$ vertices on the left are not terminals. Analogously the right end vertex of each $I_i$ is a terminal such that the next $6k$ vertices on the right are not terminals. Moreover the endpoints are the only vertices of the interval with such property. As we want the compression to be unique we enforce that the left endpoint of $I_1$ is the terminal with the lowest possible number in the round ordering $O$.

One can construct such intervals with the following procedure: find the lowest numbered terminal $\tau$ such that the $6k$ vertices on the left of $\tau$ are not terminals. Set $\tau$ to be the left endpoint of $I_1$. Check the $6k$ vertices on the right of $\tau$: if they contain no terminal, set $\tau$ to be the right endpoint of $I_1$; if they contain another terminal $\tau'$, repeat the check starting from $\tau'$ until the next $6k$ vertices on the right do not contain terminals, in which case the last terminal found is set to be the right endpoint of the interval. In general the left endpoint of $I_i$, $i \geq 2$, is the lowest numbered terminal which is not in $\bigcup_{j=1}^{i-1} I_j$ and the right endpoint is found with same procedure used for $I_1$. The last terminal found is the right endpoint of $I_l$.

Note that unless the size of the digraph is smaller than $12k^2$ (and this cannot happen given that the cutwidth is $\Theta$), these intervals are well defined: there is at least one such interval. Let $L_i$ ($R_i$) be the set containing the $3k$ vertices on the left (right) of $I_i$. For $i = 1, \ldots, l$ define $U_i = I_i \cup L_i \cup R_i$, it is
easy to see that the $U_i$ are pairwise disjoint. Define $W_i$ as\footnote{\(W_i\) can be of any size, even empty.} the set of vertices between $U_i$ and $U_{i+1}$ (the indices are taken modulo $l$). See Figure 3.

Let $D_{II}$ be the digraph obtained from the contraction of each of the $W_i$ after, if necessary, pruning down to $k$ the multiplicity of pairs of terminals. By the definition of the intervals $l \leq 2k$ and $|I_i| \leq 2k \cdot 6k$, for every $i$. Therefore $|V(D_{II})| \leq 2k + 2k \cdot 2k \cdot 6k$ and $A(D_{II}) \leq k|V(D)|^2 = 4k^3(1 + 12k^2)^2$.

As announced, we want to show that ($D$, $\Pi$) and ($D_{II}$, $\Pi$) are equivalent instances of the problem, so that we can decide it on ($D_{II}$, $\Pi$) in a time depending only on $k$.

**Lemma 5.6.** Let $D$ be a round digraph with round ordering $O$ and cutwidth at least $\Theta$. Let $\Pi$ be a list of pairs of terminals. $D$ has a weak $\Pi$-linkage if and only if its compression with respect to $\Pi$, $\Pi_{II}$, has a weak $\Pi$-linkage.

**Proof.** Suppose that $D$ has a weak $\Pi$-linkage $P_1, ..., P_k$. It induces $k$ arc-disjoint trails (i.e. paths where the same vertex can be repeated) $T_1, ..., T_k$ in $D_{II}$ and it is easy to see that $\bigcup T_i$ contains paths $P'_1, ..., P'_k$ forming a $\Pi$-linkage in $D_{II}$.

Viceversa suppose we have a weak $\Pi$-linkage $P_1, ..., P_k$ in $D_{II}$, we want to produce new distinct terminals and reduce to Lemma 5.5, with $U = \bigcup U_i$. Note that the size of each $U_i$ is not greater than $6k \cdot 2k + 6k = 12k^2 + 6k$, hence $|U| \leq 2k(12k^2 + 6k) = 12k^2(2k + 1) =: h$.

Let $\Pi' \subseteq \Pi$ be the list of pairs of terminals that are not linked by a path completely contained in some $I_i$. For each $i$ we do the following:

Let \( \{s_{a_1}, ..., s_{a_n}, t_{b_1}, ..., t_{b_l}\} = \Pi' \cap U_i \). Let us divide $L_i (R_i)$ in three intervals of $k$ vertices and call them from right to left (from left to right) $L_1^i, L_2^i, L_3^i$ ($R_1^i, R_2^i, R_3^i$). We show how to produce arc-disjoint $\tau_{b_j}$-paths $P_{b_j}$ and $(s_{a_j}, \sigma_{a_j})$-paths $Q_{a_j}$, for some choice of $\tau_{b_1} \neq ... \neq \tau_{b_l} \in L_i (a_{a_1} \neq ... \neq a_{a_n} \in R_1^i$), there are arcs $s_{a_1}, s_{a_1}, ..., a_{a_n}, a_{a_n}$. Let $Q_{a_1} = s_{a_1}, a_{a_1}, ..., Q_{a_n} = s_{a_n}, a_{a_n}$, in this case.

1. There is no useful vertex (with respect to $U$) in $R_1^i$. By Lemma 5.3, all the vertices of $R_1^i$ have in-neighborhood of size at most $6k + 3h - 1 > |I_i|$, hence, if one fixes $\sigma_{a_1} \neq ... \neq a_{a_n} \in R_1^i$, there are arcs $s_{a_1}, s_{a_1}, ..., a_{a_n}, a_{a_n}$. Let $Q_{a_1} = s_{a_1}, a_{a_1}, ..., Q_{a_n} = s_{a_n}, a_{a_n}$, in this case.

2. There is a useful vertex $u \in R_1^i$. Let $\sigma_{a_1} \neq ... \neq a_{a_n} \in R_1^i$. For $j = 1, ..., \alpha$, let $c_j, d_j$ be an arc of $P_{b_j}$ such that $P_{b_j}[s_{a_j}, c_j] \in L_1^i \cup R_1^i$ and such that in the round ordering $O$, $c_j \leq u \leq d_j$. If $d_j \in R_1^i \cup R_2^i$, we define $Q_{a_j} = P_{b_j}[s_{a_j}, d_j, c_j]$. Note that an arc $d_j, c_j$ exists because $|N^+(d_j)| > 6k$, since it is close (in the round ordering) enough to an useful vertex. If $d_j \notin R_1^i \cup R_2^i$, let $u_j \in R_2^i$ be a vertex that is not contained in $Q_{a_1}, ..., Q_{a_{\alpha-1}}$, and define $Q_{a_j} = P_{b_j}[s_{a_j}, c_j, u_j, c_j]$. Such a vertex $u_j$ exists, since the paths $Q_{a_1}, ..., Q_{a_{\alpha-1}}$ use each at most one vertex of $R_2^i$, moreover the arc $c_j, u_j$ exists, since $D_{II}$ is round and the arc $u_j, a_{a_n}$ exists because the out-neighborhood of $u_j$ is greater than $4k$.

It is not difficult to see that in both cases the paths $Q_{a_j}$ are arc-disjoint, contained in $L_i \cup R_i$ and such that $Q_{a_j} = P_{b_j}[s_{a_j}, d_j]$ (for the second case one may observe that the paths are vertex-disjoint inside $R_2^i \cup R_3^i$ and $Q_{a_j}[I_i \cup R_1^i] = P_{b_j}[s_{a_j}, d_j]$). With a symmetric argument one gets arc-disjoint paths $P_{b_j}$ contained in $L_i \cup L_i$ such that $P_{b_j}[I_i] = P_{b_j}$, thus the paths $P_{b_1}, ..., P_{b_l}$, use each at most one vertex of $R_2^i$, moreover the paths $Q_{a_1}, ..., Q_{a_\alpha}$, are arc-disjoint.

We can suppose, after a reenumeration, that $\Pi' = (s_1, t_1), ..., (s_p, t_p)$. We have worked out a list of distinct terminals $\Pi = (\sigma_1, \tau_1), ..., (\sigma_p, \tau_p)$ and paths $P_1, Q_1, ..., P_p, Q_p$ such that $V((P_j \cup Q_j) \cap \Pi) = \{\sigma_j, \tau_j\}$. Recall that $D$ has cutwidth at least $k(6k + 3h)^2$, so we can apply Lemma 5.5 to $D$, $\Pi$, $U$ and get arc-disjoint paths $Q_1, ..., Q_p$ as in the statement of that lemma. We now use these paths together to produce our desired linkage: define the $(s_i, t_i)$-paths

\[
P'_i = \begin{cases} Q_i, Q_i, P_i & \text{for } i = 1, ..., p \\ P'_i = P_i & \text{for } i = p + 1, ..., k \end{cases}
\]
For every $i = 1,\ldots,p$, $j = p + 1,\ldots,k$ the path $Q_i Q_j P_j$ is arc-disjoint from $P_j$, given that $P_j$ is completely contained in an interval $I_j \subset U$ and so it is arc-disjoint from $Q_i$ by Lemma 5.5 and it is arc-disjoint from $Q_i \cup P_j$ because $(Q_i \cup P_j)[I_j] = P_j[I_j]$. Moreover for $i \neq j \in [p]$ we have that $Q_i$ is arc-disjoint from $Q_j \cup P_j \subset U$, because the only arcs of $Q_i \cap U$ have tail $\sigma_i$ or head $\tau_j$ and $V((Q_i \cup P_j) \cap U) = \{\sigma_i, \tau_j\}$. Clearly $Q_i, Q_j$ are arc-disjoint and $Q_i \cup P_i, Q_j \cup P_j$ are arc-disjoint. Hence the paths $P_i', P_j'$ are arc-disjoint.

We can thereby conclude that the paths $P_1', \ldots, P_k'$ form a weak $\Pi$-linkage in $D$.

Assume $B$ is a digraph of the form $B = R[H_1,\ldots,H_r]$, where $R$ is round. Given a list of pairs of terminals $\Pi$ we can define the compression of $B$ with respect to $\Pi$ in the following way. Consider those modules $K_1,\ldots,K_l$ such that $|V(K_i)| > 1$ and $V(K_i) \cap \Pi \neq \emptyset$ and let $B'$ be obtained by contracting each module $H_i$ distinct from $K_1,\ldots,K_l$ into a single vertex. Hence $B' = R'[H_1',\ldots,H'_r]$, where $R'$ is a round digraph (which differs from $R$ only by the multiplicity of its arcs) and each $H'_i$ is either a vertex or is among $K_1,\ldots,K_l$. Let $v_1,\ldots,v_r$ be the vertices of $R'$, where $v_i$ corresponds to $H'_i$, let $\Pi'$ be the projection of $\Pi$ on $R'$, namely $\Pi'$ contains the pair $(v_i, v_j)$ for every $(\sigma, \tau) \in \Pi$, with $\sigma \in H'_i, \tau \in H'_j$. Let $R_{\Pi'}$ be the compression of $R'$ with respect to $\Pi'$, and $v_1,\ldots,v_r$ its vertices. Define the compression of $B$ to be the digraph $B_{\Pi} := R_{\Pi'}[H'_1,\ldots,H'_r]$.

The digraph $B_{\Pi}$ looks like $R_{\Pi'}$, the difference being that inside the intervals $I_j$ (defined with the compression) there are possibly some blown up vertices that make the digraph not anymore round. But in the proof of Lemma 5.6 the round property of arcs inside $I_j$ is not used to build the $(s_i, \sigma_j)$-paths (or the $(\tau_j, t_i)$-paths), and we can still join these paths with the $(\sigma_j, \tau_i)$-paths, that use no arc inside $I_j$, by Theorem 5.5. Using this fact and Lemma 2.3 we can restate Lemma 5.6, in a stronger version which will be useful later.

**Lemma 5.7.** Let $B$ be a digraph of the form $B = R[H_1,\ldots,H_r]$, where $R$ is round and has cutwidth at least $\Theta$. Let $\Pi$ be a list of pairs of terminals. $B$ has a $\Pi$-linkage if and only if $B_{\Pi}$ has a $\Pi$-linkage.

**Theorem 5.8.** For every fixed $k$ there exists a polynomial algorithm for the weak $k$-linkage problem for round digraphs.

**Proof.** Let $D$ be a round digraph and $\Pi$ a list of $k$ terminal pairs. In polynomial time one can compute the cutwidth of all the possible $i$-th reverse orderings of $D$. If all these cutwidths are greater than $\Theta$, then $(D_{\Pi}, \Pi)$ is an equivalent instance of the problem by Lemma 5.6. Given that the size of $D_{\Pi}$ does not depend on the size of $D$, but just on $k$, it is clearly polynomial to check for a weak $\Pi$-linkage in $D_{\Pi}$.

If an ordering with cutwidth smaller than $\Theta$ is found, then $D$ has cutwidth smaller than $\Theta$. Hence one can run the polynomial algorithm from Theorem 1.4 to decide the problem.

### 6 Locally semicomplete digraphs

We analyze the complexity of the weak $k$-linkage problem on the class of locally semicomplete digraphs. As the name suggests, they generalize semicomplete digraphs, with whom they share a number of nice properties. For an overview on locally semicomplete digraphs see [2].

**Definition 6.1.** A digraph $D$ is **locally semicomplete** (a local tournament) if for every $x \in V(D)$, $N^+(x)$ and $N^-(x)$ induce semicomplete digraphs (tournaments).

Locally semicomplete digraphs can be characterized by means of other classes of digraphs (Theorem 6.3).

**Definition 6.2.** A digraph $D$ is **round decomposable** if there exists a round local tournament $R$ on $r \geq 2$ vertices such that $D = R[H_1,\ldots,H_r]$, where each $H_i$ is a strong semicomplete digraph. We call $R[H_1,\ldots,H_r]$ the **round decomposition** of $D$.

In [2] there is a characterization of locally semicomplete digraphs that here is stated in a slightly weaker version.
Theorem 6.3. A locally semicomplete digraph with independence number $\alpha(D) > 2$ is round decomposable with a unique round decomposition. Moreover there exists a polynomial algorithm to decide whether a locally semicomplete digraph has a round decomposition and find it, if it exists.

Using this characterization we now prove that the weak $k$-linkage problem is polynomial on locally semicomplete digraphs using Theorem 1.3 and Theorem 3.3. We prove that the class $\Phi_2$ defined as the union of all round digraphs and all semicomplete digraphs is bombproof, in order to use Theorem 3.3 and show that the weak $k$-linkage problem is polynomial on round decomposable digraphs. We first need some lemmas: the following result immediately follows from a result in [2, Section 2.10.3].

Lemma 6.4. Let $R[H_1, ..., H_r]$ be a decomposition of a strong digraph $D$, where $R$ is a round digraph. Then for every minimal separating set $S$, there are two integers $i$ and $k$ such that $S = V(H_i) \cup \ldots \cup V(H_{i+k})$.

Lemma 6.5. There exists a polynomial algorithm to check whether a digraph $D$ is decomposable as $D = R[H_1, ..., H_r]$, with $R$ round.

Proof. The algorithm is as follows. Find a minimal separating set $S$, starting with $S' := N^+(x)$, for a vertex $x \in V(D)$ and deleting vertices from $S'$ until a minimal separating set is obtained. Construct the strong components of $D(S)$ and $D - S$ and label these $D_1, ..., D_r$, where $D_1, ..., D_p$ form an acyclic ordering of the strong components of $D - S$, and $D_{p+1}, ..., D_r$ form an acyclic ordering of the strong components of $D(S)$. Note that $p \geq 2$ and each strong component of $D - S$ must be contained in a single module. To see this assume $K_1, ..., K_a$ are the modules of $D - S$ induced by the decomposition of $D$. Note that, for $R$ to be round, the quotient digraph of $D - S$ induced by $K_1, ..., K_l$ must be round, because it is a subdigraph of $R$. If $x,y \in D_i$ were in different modules $K_a, K_b$, then the existence of $(x,y)$-paths in $D - S$ would imply, by the round property, that $K_i \rightarrow K_{i+1}$ for every $i = 1, ..., l$ (here $l+1$ means 1). This means that $D - S$ is strong, contradiction.

Now we identify two sets $D_i, D_j$, with $1 \leq i,j \leq p$ if $A(D_i, D_j) \neq \emptyset$ and $D_i \not\rightarrow D_j$, or if $A(D_j, D_i) \neq \emptyset$ and $D_j \not\rightarrow D_i$, until no pair of sets violates that condition and we are left with a collection of modules $K_1, ..., K_r$. On the other hand, if $S$ is not strong, we have again (for the same reason) that each strong component $D_j$ of $S$ must be contained in a single module, hence (unless $|S| = 1$, in which case we stop) we again identify sets $D_i, D_j$ following the same criteria as above, until we are left with a collection of modules $S_1, ..., S_m$. If $S$ is strong, to find the collection of modules, we iterate the algorithm starting with $D(S)$. Note that, by Lemma 6.4, any two sets of the form $K_i, S_j$ must be in different modules. Therefore it is easy to see that $R$ can be round if and only if the quotient digraph induced by the modules $K_1, ..., K_r, S_1, ..., S_m$ is a round digraph on at least two vertices. It is easy as well to see that the complexity of the algorithm is polynomial.

We say that a class of digraphs $\Phi$ is hereditary, if $D \in \Phi$ implies that every induced subdigraph of $D$ is in $\Phi$. The following theorem holds.

Theorem 6.6 ([2], Section 2.11). If $\Phi$ is an hereditary class of digraphs, then totally $\Phi$-decomposable digraphs are hereditary as well.

Lemma 6.7. The class $\Phi_2$ defined above is bombproof.

Proof. To produce a polynomial algorithm for the total $\Phi_2$-decomposition we use recursively both the algorithm mentioned in the proof of Lemma 4.3, to find a decomposition $S[H_1, ..., H_s]$ with $S$ semicomplete and the algorithm of Lemma 6.5. This is fine, because of Theorem 6.6, given that both round and semicomplete digraphs are hereditary.

Given a positive integer $c$ and a digraph $R \in \Phi_2$, we have already seen in Lemma 4.3 that if $R$ is semicomplete, then the weak $k$-linkage problem is polynomial on $R(c)$, therefore assume that $R$ is round. We want to show that the weak $k$-linkage problem is polynomial on $R(c)$. Consider a digraph $B \in R(c)$ and a list of $k$ pairs of terminals $I$:

- If $R$ has round cutwidth at least $\Theta$, we can obtain a polynomial algorithm using Lemma 5.7: note that $B = R[H_1, ..., H_r]$, where each $H_i$ is either a vertex or a digraph on at most $c$ vertices,
and the latter holds for at most \( c \) choices of \( i \). Thus the compression of \( B \) with respect to \( \Pi \), has size that depends only on \( k \) and \( c \), and not on the size of \( B \). By Lemma 5.7, we can solve the problem on \( B_1 \) (with a brute-force algorithm) to decide the existence of a weak \( \Pi \)-linkage on \( B \). Therefore we have a polynomial algorithm to decide the problem, given that the construction of \( B_1 \) is clearly polynomial.

- If there exists a round ordering \( O \) of \( V(R) \) with cutwidth at most \( \Theta \), then \( B \) has cutwidth bounded by a function of \( k \) and \( c \) only. Indeed let \( v_i \) be the vertex of \( R \) corresponding to \( H_i \). Consider an ordering \( O' \) of \( V(B) \), such that for every \( u \in H_i, v \in H_j, \) with \( i \neq j, u < v \) in \( O' \) implies \( v_i < v_j \) in \( O \), it is easy to see that such an ordering can be done. Recall that throughout the paper we are considering digraphs such that the multiplicity of the arcs in at most \( k \). Now each arc contributing to the cutwidth \( \theta' \) of \( O' \) is either an arc of some \( H_j \), with \( |V(H_j)| > 1 \) or an arc between \( H_i \) and \( H_j \), with \( v_i < v_j \) in \( O \). The arcs of the first type are no more than \( c \cdot c^2 \cdot k \) because there are at most \( c \) modules of size greater than one, and they have at most \( c \) vertices. The arcs of the second type are at most \( c \cdot c \cdot k \cdot \Theta \). Indeed for every \( H_i, H_j \) the arcs between \( H_i \) and \( H_j \) are at most \( c \cdot c \cdot k \). We can conclude that \( \theta' \leq k(c^3 + c^2 \cdot \Theta) \) and hence the algorithm from Theorem 1.4 runs in polynomial time on \( (B, \Pi) \).

\[ \square \]

**Theorem 6.8.** For every fixed \( k \) there exists a polynomial algorithm for the weak \( k \)-linkage problem for round decomposable digraphs.

**Proof.** Round decomposable digraphs are clearly totally \( \Phi_2 \)-decomposable. By Lemma 6.7 the class \( \Phi_2 \) is bombproof, therefore the claim follows from Theorem 3.3 \( \square \)

We are now ready to prove the main result of this section:

**Theorem 6.9.** For every fixed \( k \) there exists a polynomial algorithm for the weak \( k \)-linkage problem for locally semicomplete digraphs.

**Proof.** Given a local semicomplete digraph \( D \) and a list of terminal pairs \( \Pi \), decide (polynomially) if \( \alpha(D) \leq 2 \), in which case use the algorithm from Theorem 1.3 on \((D, \Pi)\). Otherwise, by Theorem 6.3, on can run a polynomial algorithm to find a round decomposition of \( D = R[H_1, ..., H_r] \) and then run the algorithm from Theorem 6.8 on \((R[H_1, ..., H_r], \Pi)\).

\( \square \)

### 7 Further results and related open problems

Using Theorem 3.3 we can show that the weak \( k \)-linkage problem is polynomial for another class of digraphs: directed cographs. We sketch only the main properties of these digraphs here, for further details see [4].

Let \( D_1, ..., D_h \) be a set of \( h \) disjoint digraphs. The **disjoint union** of \( D_1, ..., D_h \) is the digraph \((\bigcup_i V(D_i), \bigcup_i A(D_i))\). The **series composition** of \( D_1, ..., D_n \) is the disjoint union of these \( h \) digraphs plus all possible arcs between vertices of different \( D_i \). The **order composition** is the disjoint union of the digraphs plus all possible arcs from \( D_i \) to \( D_j \) for every \( 1 \leq i < j \leq h \).

The class of digraphs recursively defined from the single vertex under the closure of these three operations is called the class of **directed cographs**. There are other equivalent ways of defining directed cographs, for example they are exactly the digraphs in which every subdiagram of size at least 4 has a non-trivial module. There is also a characterization by forbidden subdigraphs [4]. Note that the class of directed cographs is closed under the complementation (The complement of \( D \) is defined as \( D := (V(D), \{uv \mid u \neq v \in V(D) \} \setminus A(D)) \)). Moreover directed cographs are hereditary: any induced subdigraph of a directed cograph is a directed cograph.

A **strong** module of a graph is a module that does not overlap (\( A \) overlaps \( B \) if \( A \cap B \neq \emptyset, A \setminus B \neq \emptyset, B \setminus A \neq \emptyset \)) any other module. A **maximal strong module** is a strong module that is maximal with respect to inclusion. Note that a partition of the vertices of a digraph \( D \) into modules \( H_1, ..., H_s \) induces a decomposition \( D = S[H_1, ..., H_s] \), for some digraph \( S \). The so called fundamental theorem of directed cographs allows us to precis the shape of the digraph \( S \), when \( D \) is a directed cograph and \( H_1, ..., H_s \) is the (unique) partition of \( V(D) \) with maximal strong modules.
Theorem 7.1. [8] A directed cograph $D$ is either

1. Not connected, in which case its maximal strong modules are its connected components
2. Not co-connected\(^{12}\), in which case its maximal strong modules are its connected co-components
3. Connected and co-connected, in which case $D$ is obtained with the order operation on its maximal
   strong modules.

On the light of this theorem we can say that the partition of a directed cograph $D$ with maximal
strong modules $H_1, \ldots, H_s$ induces a decomposition $D = S[H_1, \ldots, H_s]$, where $S$ is either an independent
set of vertices (in the first case) or a semicomplete digraph (in the second and third case). Recall that
we defined $\Phi_1$ as the union of all acyclic digraphs and all semicomplete digraphs. Using the hereditary
property of directed cographs we have the following

Corollary 7.2. Directed cographs are totally $\Phi_1$-decomposable.

From the results of Section 4 we get the following result:

Corollary 7.3. For every fixed $k$ there exists a polynomial algorithm for the weak $k$-linkage problem
for directed cographs.

In our results about totally $\Phi$-decomposable digraphs we assumed that $\Phi$ is bombproof. We could
(almost) drop this assumption in case of positive answers to the following two questions. The first
one is in a way similar to Lemma 2.7 and it would be a tool towards a (positive) answer of Problem
7.6, the second one is interesting on its own.

Problem 7.4. Let $\Phi$ be a class of digraphs for which there exists an algorithm $A$ to decide the weak
$k$-linkage problem, whose running time is bounded by $f(n, k)$. Given $D \in \Phi$ and $c \in \mathbb{N}$, does there exist
an algorithm $B_\Phi$ to decide the weak $k$-linkage problem on $D(c)$ whose running time is polynomially
related to $f(n, k)$?

Problem 7.5. Let $\Phi$ be a class of digraphs, such that there exists a polynomial algorithm to test
the membership to $\Phi$. Does there exist a polynomial algorithm to find a $\Phi$-decomposition of a totally
$\Phi$-decomposable digraph?

These two properties, that we had to check case by case in the previous sections (in order to prove
that our sets were bombproof), would lead to a result generalizing Theorem 3.3.

Problem 7.6. Let $\Phi$ be a class of digraphs, such that there exists a polynomial algorithm to test the
membership to $\Phi$ and such that for every fixed $k$ there exists a polynomial algorithm for the weak
$k$-linkage problem on $\Phi$. Does there exist, for every fixed $k$, a polynomial algorithm for the weak
$k$-linkage problem on the class of totally $\Phi$-decomposable digraphs?

A digraph $D$ is **semicomplete multipartite** if it does not contain three vertices $x, y, z$ such that
there is an arc between $x$ and $y$ but there is no arc between $x$ and $z$ and no arc between $y$ and $z$.
Clearly every tournament is a semicomplete multipartite digraph.

Conjecture 7.7. The weak $k$-linkage problem is polynomial for every fixed $k$ for semicomplete mul-
tripartite digraphs.

Even for semicomplete bipartite digraphs (these are obtained from complete bipartite graphs by
replacing each edge by either a 2-cycle or an arc) the weak $k$-linkage problem is wide open and the

\(^{12}\) A digraph is co-connected if its complement is connected
References


