Finding an induced subdivision of a digraph.

Jørgen Bang-Jensen, Frédéric Havet, Nicolas Trotignon

HAL Id: hal-00749187

https://inria.hal.science/hal-00749187

Finding an induced subdivision of a digraph[*]

Jørgen Bang-Jensen[†]   Frédéric Havet[‡]   Nicolas Trotignon[§]

March 10, 2012

Abstract

We consider the following problem for oriented graphs and digraphs: Given an oriented graph (digraph) G, does it contain an induced subdivision of a prescribed digraph D? The complexity of this problem depends on D and on whether G must be an oriented graph or is allowed to contain 2-cycles. We give a number of examples of polynomial instances as well as several NP-completeness proofs.

Keywords: NP-completeness, induced paths and cycles, linkings, 3-SAT.

1 Introduction

Many interesting classes of graphs are defined by forbidding induced subgraphs, see [4] for a survey. This is why the detection of several kinds of induced subgraphs is interesting, see [7] where several such problems are surveyed. In particular, the problem of deciding whether a graph G contains, as an induced subgraph, some graph obtained after possibly subdividing prescribed edges of a prescribed graph H has been studied. This problem can be polynomial or NP-complete depending on H and to the set of edges that can be subdivided. The aim of the present work is to investigate various similar problems in digraphs, focusing only on the following problem: given a digraph H, is there a polynomial algorithm to decide whether an input digraph G contains a subdivision of H?

Of course the answer depends heavily on what we mean by "contain". Let us illustrate this by surveying what happens in the realm of non-oriented graphs. If the containment relation is the subgraph containment, then for any fixed H, detecting a subdivision of H in an input graph G can be performed in polynomial time by the Robertson and Seymour linkage algorithm [9] (for a short explanation of this see e.g. [2]). But if we want to detect an *induced* subdivision of H then the answer depends on H (assuming P$\neq$NP). It is proved in [7] that detecting an induced subdivision of K_5 is NP-complete, and the argument can be reproduced for any H whose minimum degree is at least 4. Polynomial-time solvable instances trivially exist, such as detecting an induced subdivision of H when H is a path, or a graph on at most 3 vertices. But non-trivial polynomial-time solvable instances also exist, such as detecting an induced subdivision of $K_{2,3}$ that can be performed in time

$O(n^{11})$ by the Chudnovsky and Seymour's three-in-a-tree algorithm, see [5]. Note that for many graphs H, nothing is known about the complexity of detecting an induced subdivision of H: when H is cubic (in particular when $H = K_4$) or when H is a disjoint union of 2 triangles, and in many other cases.

When we move to digraphs, the situation becomes more complicated, even for the subdigraph containment relation. All the digraphs we will consider here are *simple*, i.e. they have no loops nor multiple arcs. We rely on [1] for classical notation and concepts. A *subdivision of a digraph D*, also called a *D-subdivision*, is a digraph obtained from D by replacing each arc ab of D by a directed (a, b)-path. From the NP-completeness of the 2-linkage problem, proved by Fortune, Hopcroft and Wyllie [6], it is straightforward to construct an oriented graph H such that deciding whether a given oriented graph G contains a subdivision of H as a subgraph is NP-complete. See Theorem 33.

Let us now think about the induced subdigraph relation. An induced subdigraph of a digraph G which is a subdivision of D is called an *induced subdivision* of D. When D is a digraph, we define:

PROBLEM Π_D

Input: A digraph G.

Question: Does G contain an induced subdivision of D?

In Π_D, the instance digraph G may have (directed) 2-cycles, where the 2-*cycle* is the digraph C_2 on 2 vertices a, b with 2 arcs ab and ba. Because of these 2-cycles, NP-completeness results are often quite easy to obtain, because no induced directed path can go through a 2-cycle (which by itself contains a chord). Hence 2-cycles are very convenient to force an induced directed path to go through many places of a large digraph that models an instance of 3-SAT. This yields NP-completeness results that cover large classes of detection problems. See Section 4. In fact, it can be easily shown (see Section 2) that if D is the disjoint union of *spiders* (trees obtained from disjoint directed paths by identifying one end of each path into a vertex) and at most one 2-cycle, then Π_D is polynomial-time solvable. However, except from those digraphs, we are not aware of any D for which Π_D is polynomial time solvable. We indeed conjecture that there are none. As an evidence, we show that if D is an *oriented graph*, i.e. a digraph with no 2-cycles, then Π_D is NP-complete unless it is the disjoint union of spiders (see Corollary 13).

It seems that allowing or not allowing 2-cycles is an essential distinction. Hence we also consider the restricted problem Π'_D in which the input graph G is an oriented graph.

PROBLEM Π'_D

Input: An oriented graph G.

Question: Does G contain an induced subdivision of D?

Observe that if Π_D is polynomial-time solvable then Π'_D is also polynomial-time solvable. Conversely, if Π'_D is NP-complete then Π_D is also NP-complete. Hence, NP-completeness results cover less cases for Π'_D.

Similarly to Π_D, for several D's, Π'_D is solvable by very simple polynomial-time algorithms (See Section 2). However, in this case they are not the only ones. We could obtain several digraphs for which Π'_D is solvable in polynomial time with non-trivial algorithms.

We denote by TT_3 the transitive tournament on 3 vertices a, b, c and arcs ab, ac, bc. In Subsection 5.1, we use a variant of Breadth First Search that computes only induced trees to solve Π'_{TT_3} in polynomial time.

We also study oriented paths in Subsection 5.2. An *oriented path* is an orientation of a path. The *length* of an oriented path P is its number of arcs and is denoted $l(P)$. Its first vertex is called its *origin* and its last vertex its *terminus*. The *blocks* of an oriented paths are its maximal directed subpaths. We

2

denote by A_k^- the path on vertices $s_1, s_2, \ldots, s_k, s_{k+1}$ and arcs $s_2 s_1, s_2 s_3, s_4 s_3, s_4 s_5, \ldots$ and A_k^+ the path on vertices $s_1, s_2, \ldots, s_k, s_{k+1}$ and arcs $s_1 s_2, s_3 s_2, s_3 s_4, s_5 s_4, \ldots$. These two paths are the *antidirected paths* of length $k-1$. Observe that A_k^- is the converse of A_k^+ (i.e. it is obtained from A_k^+ by reversing all the arcs); if k is odd they are isomorphic but the origin and terminus are exchanged. Clearly, an oriented path with k-blocks can be seen as a subdivision of A_k^- or A_k^+. In particular, paths with one block are the directed paths. We show that if P is an oriented path with three blocks such that the last one has length one then Π_P is polynomial-time solvable. We also use classical flow algorithms to prove that $\Pi'_{A_4^-}$ is polynomial-time solvable.

If D is any of the two tournaments on 3 vertices, namely the directed 3-cycle C_3 and the transitive tournament TT_3, then Π'_D is polynomial time solvable. Hence it is natural to study the complexity of larger tournaments. In Section 6, it is shown that if D is a transitive tournament on more than 3 vertices or the strong tournament on 4 vertices, then Π'_D is NP-complete.

Finally, in Section 7, we point out several open questions.

2   Easily polynomial-time solvable problems

There are digraphs D for which Π_D or Π'_D can be easily proved to be polynomial-time solvable. For example, it is the case for the directed k-path P_k on k vertices. Indeed, a P_k-subdivision is a directed path of length at least $k-1$ and an induced directed path of length at least $k-1$ contains an induced P_k. Hence a digraph has a P_k-subdivision if and only if it has P_k as an induced subdigraph. This can be checked in time $O(n^k)$ by checking for every set of k vertices whether or not it induces a P_k.

A vertex of a digraph is a *leaf* if its degree is one, a *node* if its out-degree or its in-degree is at least 2, and a *continuity* otherwise, that is if both its out- and in-degree equal 1. A *spider* is a tree having at most one node.

Proposition 1. *If D is the disjoint union of spiders then Π_D is polynomial-time solvable.*

Proof. A digraph G contains an induced D-subdivision if and only if it contains D as an induced subdigraph. This can be checked in time $O(n^{|V(D)|})$. □

It is also not difficult to see that Π_{C_2} is polynomial-time solvable.

Proposition 2. Π_{C_2} *is polynomial-time solvable.*

Proof. A subdivision of the directed 2-cycle is a directed cycle. In a digraph, a shortest cycle is necessarily induced, hence a digraph has a C_2-subdivision if and only if it is not acyclic. Since one can check in linear time if a digraph is acyclic or not [1, Section 2.1], Π_{C_2} is polynomial-time solvable. □

Since an oriented graph contains no 2-cycle, then $\Pi'_{C_2} = \Pi'_{C_3}$. Similarly to Π_{C_2}, this problem is polynomial-time solvable.

Proposition 3. Π'_{C_3} *is polynomial-time solvable.*

Proof. An oriented graph contains an induced subdivision of C_3 if and only if it is not acyclic. □

Moreover, the following is polynomial-time solvable.

Proposition 4. *If D is the disjoint union of spiders and a C_2 then Π_D is polynomial-time solvable.*

Proof. $D' = D - C_2$ is a collection of spiders. Let p be its order. For each set A of p vertices, we check if the digraph $G\langle A \rangle$ induced by A is D' and if yes we check if $G - (A \cup N(A))$ has a directed cycle. □

3

Similarly,

Proposition 5. *If D is the disjoint union of spiders and a C_3 then Π'_D is polynomial-time solvable.*

3 NP-completeness results for oriented graphs

In all proofs below it should be clear that the reductions can be performed in polynomial time and hence we omit saying this anymore. Before starting with the NP-completeness proofs, we state a proposition.

Proposition 6. *Let D be a digraph and C a connected component[1] of D. If Π_C is NP-complete then Π_D is NP-complete. Similarly, if Π'_C is NP-complete then Π'_D is NP-complete.*

Proof. Let $D_1, \ldots, D_k$ be the components of D and assume that Π_{D_1} is NP-complete. To show that Π_D is NP-complete, we will give a reduction from Π_{D_1} to Π_D.

Let G_1 be an instance of Π_{D_1} and G be the digraph obtained from D by replacing D_1 by G_1. We claim that G has an induced D-subdivision if and only if G_1 has an induced D_1-subdivision.

Clearly, if G_1 has an induced D_1-subdivision S_1 then the disjoint union of S_1 and the D_i, $2 \leq i \leq k$ is an induced D-subdivision in G.

Reciprocally, assume that G contains an induced D-subdivision S. Let S_i, $1 \leq i \leq k$ be the connected components of S such that S_i is an induced D_i-subdivision. Set $G_i = D_i$ if $i \geq 2$. Then the G_i's are the connected components of G. Thus S_1 is contained in one of the G_i's. If it is G_1 then we have the result. Otherwise, it is contained in some other component say $G_2 = D_2$. In turn, S_2 is contained in some G_j. Hence G_j contains a D_1-subdivision because S_2 contains a D_1-subdivision since D_2 contains S_1. Thus G_j cannot be G_2 since G_2 already contains D_1 and $|S_2| \geq |G_2|$. If $j = 1$ then we have the result. If not we may assume that $j = 3$. And so on, for every $i \geq 3$, applying the same reasoning, we show that one of the following occurs:

- S_i is contained in G_1 and thus G_1 contains a D_1-subdivision because S_i did.

- S_i is contained in G_j which cannot be any of the G_i, $1 \leq l \leq i$, for cardinality reasons. Hence we may assume that $G_j = G_{i+1}$ and that G_{i+1} and hence S_{i+1} contains a D_1-subdivision.

Since the number of components is finite, the process must stop, so G_1 contains an induced D_1-subdivision. □

3.1 Induced (a,b)-path in an oriented graph

Our first result is an easy modification of Bienstock's proof [3] that finding an induced cycle through two given vertices is NP-complete for undirected graphs.

Lemma 7. *It is NP-complete to decide whether an oriented graph contains an induced (a,b)-path for prescribed vertices a and b.*

Proof. Given an instance I of 3-SAT with variables $x_1, x_2, \ldots, x_n$ and clauses $C_1, C_2, \ldots, C_m$ we first create a variable gadget V_i^1 for each variable x_i, $i = 1, 2, \ldots, n$ and a clause gadget C_j^1 for each clause C_j, $j = 1, 2, \ldots, m$ as shown in Figure 1. Then we form the digraph $G_1(I)$ as follows (see Figure 2): Form a chain U of variable gadgets by adding the arcs $b_i a_{i+1}$ for $i = 1, 2, \ldots, n-1$ and a chain W of clause gadgets by adding the arcs $d_j c_{j+1}$, $j = 1, 2, \ldots, m-1$. Add the arcs $aa_1, b_n c_1, c_m b$ to get a chain from a to b. For each clause C, we connect the three literal vertices of the gadget for C to the variable gadgets for variables occuring as literals in C in the way indicated in the figure. To be precise, suppose $C_p = (x_i \vee \bar{x}_j \vee x_k)$, then we add the following three 3-cycles $l_p^1 x_i v_i l_p^1$, $l_p^2 \bar{x}_j \bar{v}_j l_p^2$ and $l_p^3 x_k v_k l_p^3$. This concludes the construction of $G_1(I)$.

---

[1] A connected component of a digraph H is a connected component in the underlying undirected graph of H.
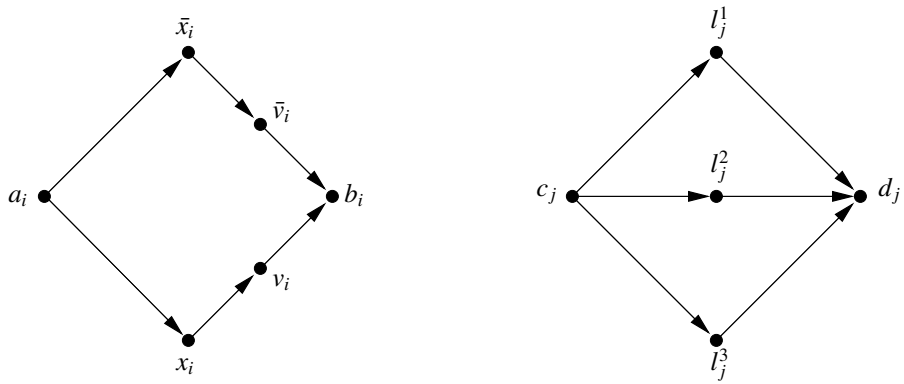
4

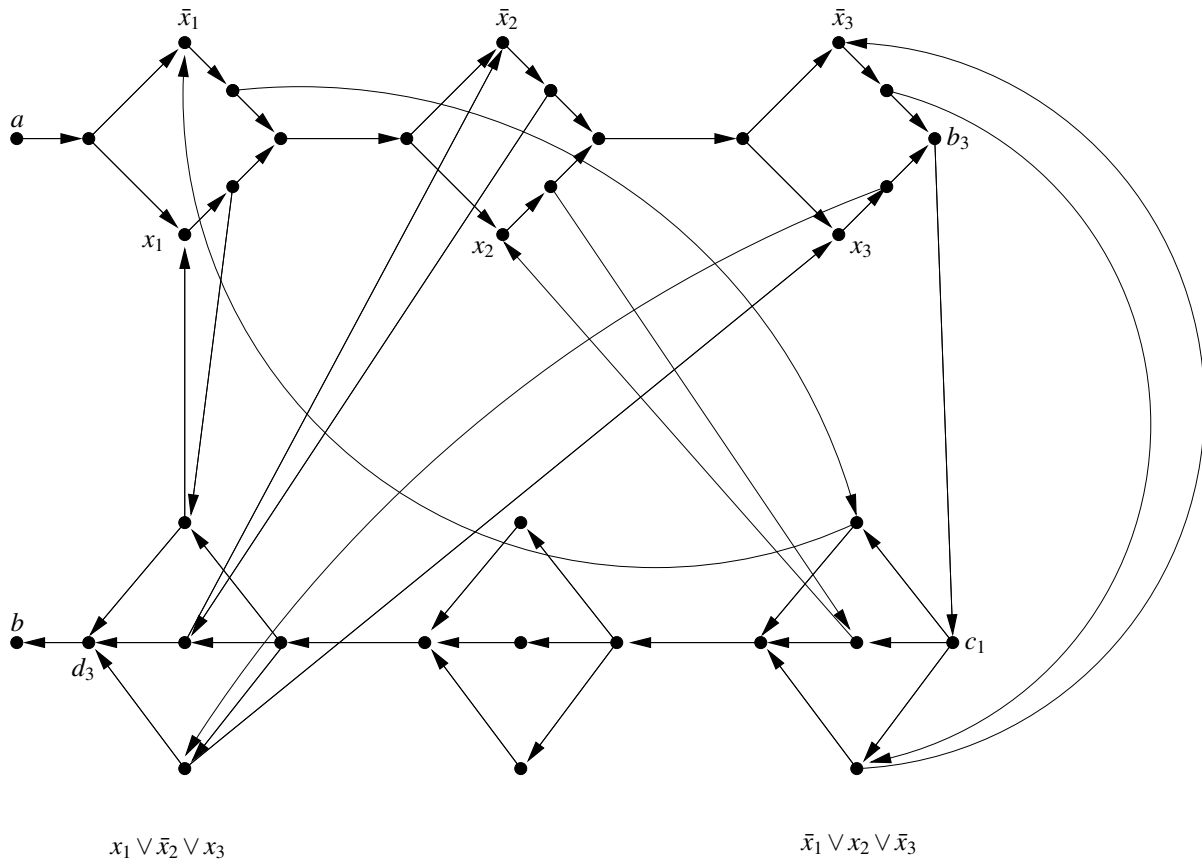Figure 1: The variable gadget V_i^1 (left) and the clause gadget C_j^1 (right).

Figure 2: The digraph $G_1(I)$ when I has variables x_1, x_2, x_3 and three clauses C_1, C_2, C_3 where $C_1 = (\bar{x}_1 \vee x_2 \vee \bar{x}_3)$ and $C_3 = (x_1 \vee \bar{x}_2 \vee x_3)$ (for clarity we do not show the arcs corresponding to C_2)

We claim that there is an induced directed (a,b)-path in $G_1(I)$ if and only if I is satisfiable. Suppose first that I is satisfiable and consider a truth assignment T which satisfies I. Now form a directed (a,b)-path P by taking the arcs aa_1, c_mb and the following subpaths: for each variable x_i take the subpath $a_i\bar{x}_i\bar{v}_ib_i$ if T sets x_i true and otherwise take the subpath $a_ix_iv_ib_i$. For each clause C_j we fix a litteral l'_j which is satisfied by T and take the subpath $c_jl'_jd_j$. It is easy to check that P is induced as we navigate it to avoid each of the arcs between the variable chain U and the clause chain W. Suppose now that Q is an induced directed (a,b)-path in $G_1(I)$. It follows from the construction that Q starts by a directed (a_1, b_n)-path through all variable gadgets which contains no vertices from W and continues with a directed (c_1, d_m)-path through all clause gadgets which contains no vertices from U. This follows from the presence of the directed 3-cycles that prevent Q from using any of the arcs going from a variable gadget to a clause gadget other than the arc b_nc_1. Similarly there is no induced directed (c_1, d_m)-path which contains any vertex from U. Now form a truth assignment by setting x_i true if and only if Q uses the subpath $a_i\bar{x}_i\bar{v}_ib_i$ and false otherwise. Since Q is induced, for each clause C_j if Q uses the subpath $c_jl'_jd_j$, then we claim that l'_j will be true with the truth assignment just described: if $l'_j = x_k$ for some k then since Q is induced the presence of the arc l'_jx_k implies that Q uses the path $a_k\bar{x}_k\bar{v}_kb_k$ and similarly, if $l'_j = \bar{x}_k$ then Q uses the path $a_kx_kv_kb_k$ and again C_j is satisfied. $\square$

3.2 Induced subdivisions of directed cycles

We first show that for any $k \geq 4$, the problem Π'_{C_k} is NP-complete.

Theorem 8. *It is NP-complete to decide whether an oriented graph contains an induced subdivision of a fixed directed cycle of length at least 4.*

Proof. Given an instance I of 3-SAT with variables $x_1, x_2, \ldots, x_n$ and clauses $C_1, C_2, \ldots, C_m$ we form the digraph $G_1^*(I)$ from $G_1(I)$ which we defined above by adding the arc ba.

Let C be an induced cycle of $G_1^*(I)$. Since the variable chain U and the clause chain W are both acyclic, C must contain an arc with tail l in W and head y in U. If $ly \neq ba$, then there exists i such that $y \in \{x_i, \bar{x}_i\}$ and so $C = lx_iv_il$ or $C = l\bar{x}_i\bar{v}_il$ by construction of $G_1^*(I)$. Hence every induced directed cycle of length at least 4 contains the arc ba. Thus $G_1^*(I)$ has an induced cycle of length at least 4 if and only if $G_1(I)$ has an induced directed (a,b)-path. As shown in the proof of Lemma 7 this is if and only if I is satisfiable. $\square$

Theorem 9. *Let D be an oriented graph containing an induced directed cycle of length at least 4 with a vertex of degree[2] 2. It is NP-complete to decide whether a given oriented graph contains an induced subdivision of D.*

Proof. Let D be given and let I be an arbitrary instance of 3-SAT. Fix an induced directed cycle C of length at least 4 in D and fix an arc uv on C such that u is of degree 2. Let $G'_1(I)$ be the oriented graph that we obtain by replacing the arc uv by a copy of $G_1(I)$ and the arcs ua, bv. We claim that $G'_1(I)$ contains an induced subdivision of D if and only if I is satisfiable (which is if and only if $G_1(I)$ contains an induced directed (a,b)-path).

Clearly, if $G_1(I)$ has an induced directed (a,b)-path, then we may use the concatention of this path with ua and bv instead of the deleted arc uv to obtain an induced D-subdivision in $G'_1(I)$ (the only subdivided arc will be uv).

Conversely, suppose that $G'_1(I)$ contains an induced subdivision D' of D. Clearly D' has at least as many vertices as D and thus must contain at least one vertex z of $V(G_1(I))$. Since u is of degree 2, the digraph $D \setminus uv$ has fewer induced directed cycles of length at least 4 than D. (Note that the fact

---

[2]The degree of a vertex v in a digraph is the number of arcs with one end in v, that is, the sum of the in- and out-degree of v.

6

that u is of degree 2 is important: if u has degree more than 2, deleting uv could create new induced directed cycles.) Thus z must be on a cycle of length at least 4 in D'. But this and the fact that $G_1(I)$ has no induced directed cycle of length at least 4 implies that $G_1'(I)$ contains an induced directed (a,b)-path (which passes through z). □

We move now to the detection of induced subdivisions of digraphs H when H is the disjoint union of one or more directed cycles, all of length 3. If there is just one cycle in H, the problem is polynomial-time solvable by Proposition 3. But from two on, it becomes NP-complete. We need results on the following problem.

PROBLEM DIDPP

Input: An acyclic digraph G and two vertex pairs $(s_1,t_1),(s_2,t_2)$. Moreover, there is no directed path from $\{s_2,t_2\}$ to $\{s_1,t_1\}$.

Question: Does G have two paths P_1, P_2 such that P_i is a directed (s_i,t_i)-path, $i = 1,2$, and $G\langle V(P_1) \cup V(P_2)\rangle$ is the disjoint union of P_1 and P_2?

Problem k-DIDPP was shown to be NP-complete by Kobayashi [8] using a proof similar to Bienstock's proof in [3].

Theorem 10. *Let D be the disjoint union of two directed cycles with no arcs between them. Then Π_D' is NP-complete.*

Proof. Let G be an instance of DIDPP and H the oriented graph obtained from it by adding new vertices u_1, u_2 and the arcs t_1u_1, u_1s_1, t_2u_2 and u_2s_2. Since G was acyclic it is not difficult to see that H is a yes-instance of Π_D' if and only if G is a yes-instance of DIDPP. □

4 NP-completeness results for digraphs

Figure 3: The variable gadget V_i^2 (left) and the clause gadget C_j^2 (right). Unoriented bold edges represent 2-cycles.

Theorem 11. *Let $k \geq 3$ be an integer. Then Π_{C_k} is NP-complete.*

Proof. Reduction from 3-SAT. Let I be an instance of 3-SAT with variables $x_1, x_2, \ldots, x_n$ and clauses $C_1, C_2, \ldots, C_m$. We first create a variable gadget V_i^2 for each variable x_i, $i = 1,2,\ldots,n$ and a clause gadget C_j^2 for each clause C_j, $j = 1,2,\ldots,m$ as shown in Figure 3. Then we form the digraph $G_2(I)$ as follows (see Figure 4): Form a chain U of variable gadgets by adding the arcs b_ia_{i+1} for $i =$

7

1 $1, 2, \ldots, n-1$ and a chain W of clause gadgets by adding the arcs $d_j c_{j+1}$, $j = 1, 2, \ldots, m-1$. Add the
2 arcs $aa_1, b_n c_1, c_m b$ to get a chain from a to b. For each clause C, we connect the three literal vertices
3 of the gadget for C to the variable gadgets for variables occuring as literals in C in the following way.
4 Suppose $C_p = (x_i \vee \bar{x}_j \vee x_k)$, then we add the following three 2-cycles $l_p^1 x_i l_p^1$, $l_p^2 \bar{x}_j l_p^2$ and $l_p^3 x_k l_p^3$. This
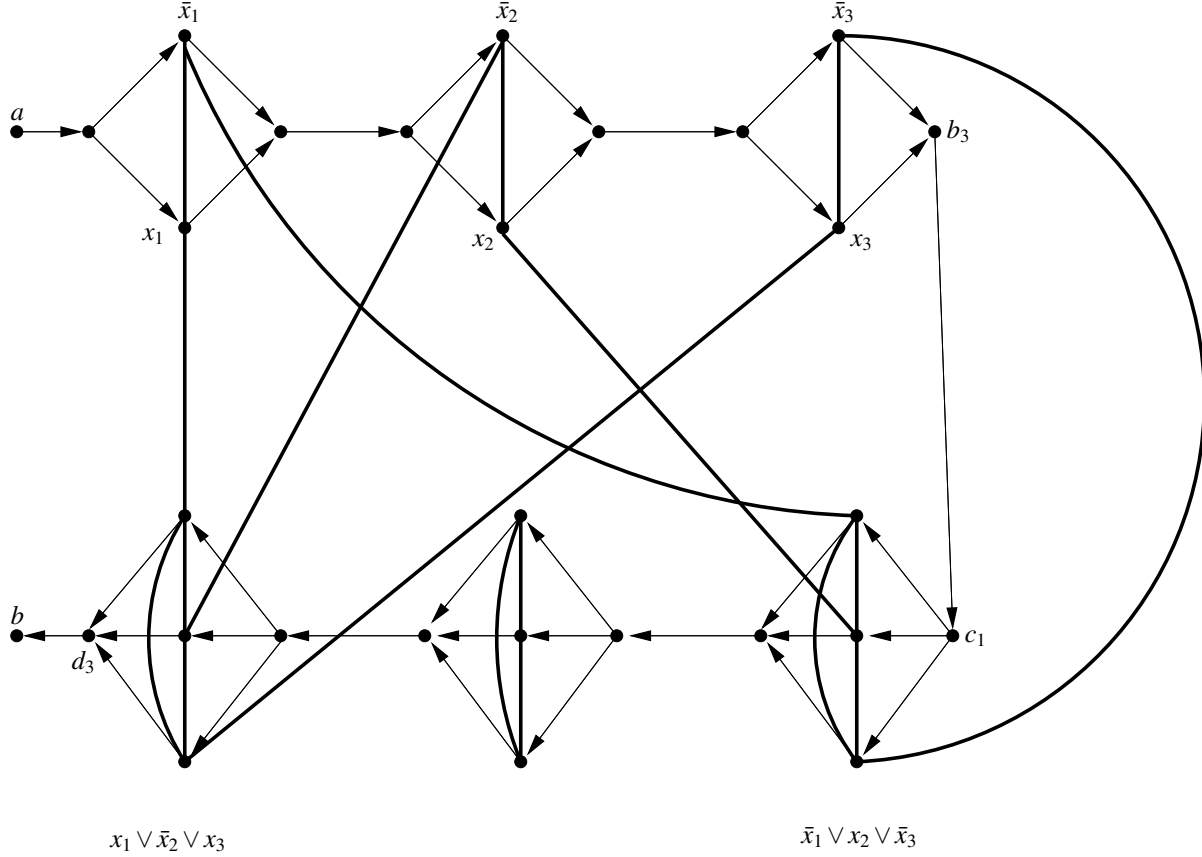5 concludes the construction of $G_2(I)$. See Figure 4.

Figure 4: The digraph $G_2(I)$ when I has variables x_1, x_2, x_3 and three clauses C_1, C_2, C_3 where $C_1 = (\bar{x}_1 \vee x_2 \vee \bar{x}_3)$ and $C_3 = (x_1 \vee \bar{x}_2 \vee x_3)$ (for clarity we do not show the arcs corresponding to C_2)

6 Similarly to the proof of Lemma 7, one can show that there is an induced directed (a,b)-path in
7 $G_2(I)$ if and only if I is satisfiable.
8 Let $G_2^k(I)$ be the digraph obtained from C_k by replacing one arc ab by $G_2(I)$. It is easy to check
9 that $G_2(I)$ has no induced cycle of length at least 3. Hence $G_2^k(I)$ has an induced directed cycle of
10 length k if and only if $G_2(I)$ has an induced directed (a,b)-path. Hence by Lemma 7, $G_2^k(I)$ has an
11 induced D-subdivision if and only if I is satisfiable. □

12 A *branch* is a directed walk such that all the vertices are distinct except possibly its ends, its ends
13 are nodes or leaves and all its internal vertices are continuities. A branch is *central* if its two ends are
14 nodes.
15 The *skeleton* of a multidigraph D is the digraph whose vertices are the nodes and leaves in D and
16 in which ab is an arc if and only if there is a directed (a,b)-branch in D. Observe that a skeleton may
17 have loops and multiple arcs. Clearly, any subdivision of D has the same skeleton as D.

18 **Theorem 12.** *Let D be an oriented graph. If D contains a central branch, then Π_D is NP-complete.*

8

Proof. Reduction from 3-SAT. Let I be an instance of 3-SAT. Let B be a central branch with origin a and terminus c. Let $G_2^D(I)$ be the digraph obtained from D by replacing the first arc ab of B by $G_2(I)$.

Clearly if $G_2(I)$ has an induced directed (a,b)-path P, then the union of P and $D \setminus ab$ is a D-subdivision (in which only ab is subdivided) in $G_2^D(I)$.

Conversely, assume that $G_2^D(I)$ contains an induced D-subdivision S. It is easy to check that no vertex in $V(G_2(I)) \setminus \{a,b\}$ can be a node of S (the 2-cycles prevent this). Then since S has the same skeleton as D, a and b are nodes of S. In addition, since the number of central branches in $D \setminus ab$ is one less than the number of central branches in D, one central branch of D must use vertices of $G_2(I)$. Thus, there is an induced directed (a,b)-path in $G_2(I)$.

Hence $G_2^D(I)$ has an induced D-subdivision if and only if $G_2(I)$ has an induced directed (a,b)-path and thus if and only if I is satisfiable. □

Corollary 13. *Let D be an oriented graph. Then Π_D is NP-complete unless D is the disjoint union of spiders.*

Proof. Let D be an oriented graph. If one of its connected components is neither a directed cycle nor a spider, then it contains at least one central branch. So Π_D is NP-complete by Theorem 12.

If one of the components is directed cycle of length at least 3, then Π_D is NP-complete by Theorem 11 and Proposition 6.

Finally, if all its connected components are spiders then Π_D is polynomial-time solvable according to Theorem 5. □

We believe that Corollary 13 can be generalized to digraphs.

Conjecture 14. *Let D be a digraph. Then Π_D is NP-complete unless D is the disjoint union of spiders and at most one 2-cycle.*

As support for this conjecture, we give some other digraphs D (which are not oriented graphs), for which Π_D is NP-complete. In particular, when D is the *lollipop*, that is the digraph L with vertex set $\{x,y,z\}$ and arc set $\{xy,yz,zy\}$. Note that the lollipop seems to be the simplest digraph that is not an oriented graph nor a C_2. So it should be an obvious candidate for a further polynomial case if one existed.

Theorem 15. *Deciding if a digraph contains an induced subdivision of the lollipop is NP-complete.*

Proof. Reduction from 3-SAT. Let I be an instance of 3-SAT with variables $x_1, x_2, \ldots, x_n$ and clauses $C_1, C_2, \ldots, C_m$. We first create a variable gadget V_i^3 for each variable x_i, $i = 1, 2, \ldots, n$ and a clause gadget C_j^3 for each clause C_j, $j = 1, 2, \ldots, m$ as shown in Figure 5. Then we form the digraph $G_3(I)$ as follows: Form a chain U of variable gadgets by adding the arcs $b_i a_{i+1}$ for $i = 1, 2, \ldots, n-1$ and a chain W of clause gadgets by adding the arcs $d_j c_{j+1}$, $j = 1, 2, \ldots, m-1$. Add the arcs $aa_1, b_n c_1, c_m b$ to get a chain from a to b. For each clause C, we connect the three literal vertices of the gadget for C to the variable gadgets for variables occuring as literals in C in the way indicated in the figure.

Similarly to the proof of Lemma 7, one can check that there is an induced directed (a,b)-path in $G_3(I)$ if and only if I is satisfiable.

The digraph $G_3^L(I)$ is obtained from L and $G_3(I)$ by deleting the arc yz and adding the arcs ya and bz.

It is easy to see that $G_3(I)$ has no induced directed cycle of length 3 and that no 2-cycle is contained in an induced lollipop. Hence if $G_3^L(I)$ contains an L-subdivision, the induced directed cycle in it is the concatenation of the path $bzya$ and a induced directed (a,b)-path in $G_3(I)$. Thus I is satisfiable. The other direction is (as usual) clear. □
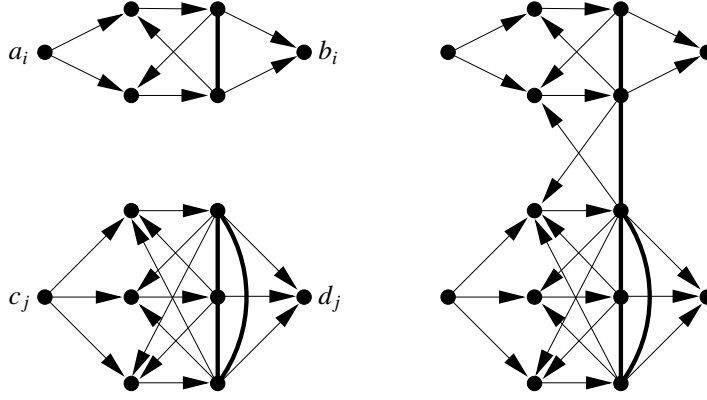
9

Figure 5: The variable gadget V_i^3, (top left), the clause gadget C_i^3 (bottom left) and the way to connect them in $G_3(I)$ (right). Bold unoriented edges represent 2-cycles. Only the connection for one variable gadget and one clause gadget is shown and the general strategy for connecting variable and clause gadgets is the same as in $G_1(I)$ (Figure 2).

Remark 16. The *cone* is the digraph C with vertex set $\{x, y, z\}$ and arc set $\{xy, xz, yz, zy\}$. In the very same way as Theorem 15, one can show that finding an induced subdivision of the cone in a digraph is NP-complete.

5 Polynomial-time algorithms for induced subdivisions in oriented graphs

According to Conjecture 14, the only digraphs for which Π_D is polynomial-time solvable are disjoint unions of spiders and possibly one 2-cycle. For such digraphs, easy polynomial-time algorithms exist (See Section 2).

In this section, we show that the picture is more complicated for Π'_D than for Π_D. We show some oriented graphs D for which Π'_D is polynomial-time solvable. For all these oriented graphs, Π_D is NP-complete by Corollary 13.

5.1 Induced subdivision of cherries in oriented graphs

Let s, u, v be three vertices such that $s \neq v$ and $u \neq v$ (so $s = u$ is possible). A *cherry on* (s, u, v) is any oriented graph made of three induced directed paths P, Q, R such that:

- P is directed from s to u (so when $s = u$ it has length 0);

- Q and R are both directed from u to v (so they both have length at least 1 and since we do not allow parallel edges, at least one of them has length at least 2);

- u, v are the only vertices in more than one of P, Q, R;

- there are no other arcs than those from P, Q, R.

The cherry is *rooted at s*.

An induced cherry contains an induced TT_3-subdivision (made of Q and R) and a TT_3-subdivision is a cherry (with $u = s$). Hence detecting an induced cherry is equivalent to detecting an induced TT_3-subdivision.

In order to give an algorithm that detects a cherry rooted at a given vertex, we use a modification of the well-known Breadth First Search algorithm (BFS), see e.g. [1, Section 3.3]. Given an oriented graph G and a vertex $s \in V(D)$, BFS returns an out-tree rooted at s and spanning all the vertices reachable from s. It proceeds as follows:

BFS(G, s)

    Create a queue Q consiting of s; Intialize $T = (\{s\}, \emptyset)$

    while Q *is not empty* **do**

        Consider the head u of Q and *visit u*, that is

        foreach *out-neighbour v of u in D* **do**

            if $v \notin V(T)$ **then**

                $V(T) := V(T) \cup \{v\}$ and $A(T) := A(T) \cup \{uv\}$

                Put v to the end of Q

            Delete u from Q

Note that the arc-set of the out-branching produced by BFS depends on the order in which the vertices are visited, but the vertex-set is always the same: it is the set of the vertices reachable from s. See [1] p. 92 for more details on BFS. We need the following variant:

IBFS(G, s)

    Create a queue Q consisting of s; Intialize $T = (\{s\}, \emptyset)$

    while Q *is not empty* **do**

        Consider the head u of Q and *visit u*, that is

        foreach *out-neighbour v of u in G* **do**

            if $N_G(v) \cap V(T) = \{u\}$ **then**

                $V(T) := V(T) \cup \{v\}$ and $A(T) := A(T) \cup \{uv\}$

                Put v to the end of Q

            Delete u from Q

Observe that IBFS (which we also call *induced-BFS*) is the same as BFS except that we add the out-neighbour v of u to T only if it has no other neighbour already in T, hence ensuring that the resulting out-tree is an induced subdigraph of G. Contrary to BFS, the vertex-set of a tree obtained after IBFS may depend on the order in which the vertices are visited.

IBFS can easily be implemented to run in time $O(n^2)$. When T is an oriented tree, we denote by $T[x, y]$ the unique oriented path from x to y in T.

Theorem 17. *Let G be an oriented graph, s a vertex and T a tree obtained after running IBFS(G, s). Then exactly one of the following outcomes is true:*

(i) D contains an induced subdigraph that is a cherry rooted at s;

(ii) for every vertex x of T, any out-neighbour of x not in T has an out-neighbour that is an ancestor of x in T.

This is algorithmic in the sense that there is an $O(n^2)$ algorithm that either outputs the cherry of (i) or checks that (ii) holds.

Proof. Suppose that T does not satisify (ii). Then some vertex x of T has an out-neighbour y not in T and no out-neighbour of y is an ancestor of x. Without loss of generality, we assume that x is the first vertex added to T when running IBFS with such a property. In particular, $T[s, x]y$ is an induced directed path because a chord would contradict (ii) or the choice of x. Let v be the neighbour of y in T, different from x, that was first added to T when running IBFS. Note that v exists for otherwise y would have been added to T when visiting x. If x is the parent of v in T then $T[s, x]y$ together with v

11

form a cherry rooted at s (whatever the orientation of the arc between y and v). So we may assume that x is not the parent of v. When visiting x, vertex y was not added to T, hence v was already visited (because x is not the parent of v). In addition, when v was visited, it was the unique neighbour of y in the current out-tree, so y is an in-neighbour of v, for otherwise it would have been added to T. Let u be the common ancestor of x and v in T, chosen closest to x. Since T does not satisfy (ii) by the choice of x and y, $u \neq v$. Now the directed paths sTu, $T[u,x]yv$ and $T[u,v]$ form an induced cherry rooted at s. Indeed since T is an induced out-tree, it suffices to prove that y has no neighbour in these three paths except x and v. By definition of v, there is no neighbour of y in $T[s,u]$ and $T[u,v]$ except v. Moreover, y has no out-neighbour in $T[u,x]$ by the assumption that (ii) does not hold for y and x and it has no in-neighbour in $T[u,x]$ except x by the choice of x.

Conversely, let us assume that T satisfies (ii) and suppose by contradiction that G contains an induced cherry C rooted at s. Since T is an induced out-branching, some vertices of C are not in T. So, let y be a vertex of $V(C) \setminus V(T)$ as close to s as possible in the cherry. Let x be an in-neighbour of y in $C \cup T$. From the choice of y, x and all its ancestors along the cherry are in T. Since T is induced, the ancestors of x along the cherry are in fact the ancestors of x along T. Hence, x is a vertex of T with an out-neighbour y not in T having no out-neighbour among the ancestors of x along T. This contradicts T satisifying (ii).

All this may be turned in an $O(n^2)$-algorithm that finds a cherry rooted at s if it exists or answer no otherwise. Indeed we first run IBFS and then check in time $O(n^2)$ if the obtained tree T satisfies (ii). If not, then we can find the cherry following the first paragraph of the proof. □

Remark 18. Since a digraph contains an induced TT_3-subdivision if and only if it contains an induced cherry, Theorem 17 implies directly that Π'_{TT_3} is solvable in time $O(n^3)$ (because we need to enumerate all potential roots).

We can slightly extend our result. A *tiny cherry* is a cherry such that the path Q and R as in the definition form a TT_3.

Corollary 19. *For any tiny cherry D, the problem Π'_D is solvable in time $O(n^{|V(D)|})$.*

Proof. Let P be the path of D as in the definition of cherry. Let G be the input oriented graph. Enumerate by brute force all induced directed paths of order $|P|$ by checking all the possible subdigraphs of order $|P|$. For each such path P' with terminus x, look for a cherry rooted at x in the graph G' obtained by deleting all the vertices of $P - x$ and their neighbourhoods except x. If there is such a cherry C then the union of P and C is an induced D-subdivision. □

Similarly to Propositions 4 and 5, we have the following.

Corollary 20. *If D is the disjoint union of spiders and a tiny cherry then Π'_D is polynomial-time solvable.*

5.2 Induced subdivision of oriented paths with few blocks in oriented graphs.

By Proposition 1, for any oriented path P with at most two blocks Π_P and thus Π'_P are polynomial-time solvable. In this section, we shall prove that Π'_P is polynomial-time solvable for some oriented paths with three or four blocks. In contrast, Π_P is NP-complete for every oriented path with at least three blocks as shown in Corollary 13.

5.2.1 Oriented path with three blocks

Theorem 21. *There exists an algorithm of complexity $O(m^2)$ that given a connected oriented graph on n vertices and m arcs with a specified vertex s returns an induced A_2^+-subdivision with origin s if one exists, and answer 'no' if not.*

Proof. Observe that any induced A_2^+-subdivision with origin s contains an induced A_2^+-subdivision with origin s such that the directed path corresponding to the arc $s_3 s_2$ is some arc f. Such a subdivision is called *f-leaded*.

Given an oriented graph G, we enumerate all arcs $f = s_3' s_2'$. For each arc in turn we either show that there is no f-leaded induced A_2^+-subdivision with origin s or give an induced subdivision of A_2^+ with origin s, (but not necessarily f-leaded). This will detect the A_2^+-subdivision since if some exists, it is f-leaded for some f.

We do this as follows. We delete all in-neighbours of s and all neighbours of s_3' except s_2'. Let us denote by G' the resulting graph. Then we compute by BFS a shortest directed path P from s to s_2'. If it is induced, together with $s_3' s_2'$, it forms the desired A_2^+ subdivision. So, as P has no forward chord (since it is a shortest path), there is an arc uv in $G'\langle V(P)\rangle$ such that u occurs after v on P. Take such an arc $b_3 b_2$ such that b_2 is as close as possible to s (in P). Observe that since we deleted all in-neighbours of s we have $b_2 \neq s$. Now, $P[s_1, b_2]$ together with $b_3 b_2$ forms the desired A_2^+-subdivision.

There are $O(m)$ arcs and for each of them we must find a shortest path in G' which can be done in $O(m)$. Hence the complexity of the algorithm is $O(m^2)$. □

From this theorem, one can show that finding an induced A_3^--subdivision is polynomial-time solvable. It is enough to enumerate all arcs $s_2' s_1'$, to delete s_1' and its neighbours except s_2', and to decide whether there exists in what remains an A_2^+-subdivision with origin s_2. One can also derive polynomial-time algorithms for finding induced subdivisions of other oriented paths with three blocks.

Corollary 22. *Let P be a path with three blocks such that the last one has length 1. One can check in time $O(n^{|P|-2}m^2)$ whether a given oriented graph contains an induced P-subdivision.*

Proof. By directional duality, we may assume that P is an A_3^--subdivision. Let Q be the subdigraph of P formed by the first block of P and the second block of P minus the last arc. Let s be the terminus of Q. For each induced oriented path Q' in the instance graph, isomorphic to Q (there are at most $O(n^{|P|-2})$ of them), we delete $Q' - s$ and all vertices that have neighbours in $Q - s$ except s. We then detect an A_2^+-subdivision rooted at s in the resulting graph. This will detect a P-subdivision if there is one. □

5.2.2 Induced subdivision of A_4^- in an oriented graph

We show how to check the presence of an induced copy of A_4^- by using flows (for definitions and algorithms for flows see e.g. [1, Chapter 4]).

Theorem 23. *There exists an algorithm of complexity $O(nm^2)$ that given an oriented graph on n vertices and m arcs with a specified vertex s returns an induced A_3^+-subdivision rooted at s, if one exists, and answer 'no' if not.*

Proof. The general idea is close to the one of the proof of Theorem 21. Observe that any induced A_3^+-subdivision with origin $s = a_1$ contains an induced subdivision of A_3^+ with origin $s = a_1$ such that the directed path corresponding to the arc $s_3 s_4$ is some arc f. If, in addition, the vertex corresponding to s_2 is v, such a subdivision is called (v, f)-*leaded*.

Given an oriented graph G, we enumerate all pairs $(a_2, a_3 a_4)$ such that a_2, a_3, a_4 are distinct vertices and $a_3 a_4 \in E(G)$. For each such pair in turn we either show that there is no $(a_2, a_3 a_4)$-leaded induced A_3^+-subdivision with origin a_1 or give an induced subdivision of A_3^+ with origin a_1 (but not necessarily $(a_2, a_3 a_4)$-leaded).

We do this as follows. We first delete all the neighbours of a_4 except a_3, all in-neighbours of a_1 and a_3 and finally all out-neighbours of a_2. If this results in one or more of the vertices $a_1, \ldots, a_4$ to

13

be deleted, then there cannot be any (a_2, a_4a_3)-leaded induced A_3^+-subdivision with origin a_1 because there is an arc in $G\langle\{a_1, \ldots, a_4\}\rangle$ which is not in $\{a_1a_2, a_3a_2, a_3a_4\}$. So we skip this pair and proceed to the next one. Otherwise we delete a_4 and we use a flow algorithm to check in the resulting digraph G' the existence of two internally-disjoint directed paths P, Q such that the origin of P and Q are a_1 and a_3 respectively and such that a_2 is the terminus of both P and Q. Moreover, we suppose that these two paths have no forward chord (this can easily be ensured by running BFS on the graphs induced by each of them). If no such paths exist , then we proceed to the next pair because there is no (a_2, a_3a_4)-leaded induced A_3^+-subdivision. If we find such a pair of directed paths P, Q, then we shall provide an induced subdivision of A_3^+ with origin a_1. If P and Q are induced and have no arcs between them, then these paths together with the arc a_3a_4 form the desired induced subdivision of A_3^+.

Suppose that P is not induced. As P has no forward chord, there is an arc uv in $G'\langle V(P)\rangle$ such that u occurs after v on P. Take such an arc b_3b_2 such that b_2 is as close as possible to a_1 (in P), and subject to this, such that b_3 is as close as possible to a_2. Observe that since we deleted all in-neighbours of a_1 and all out-neighbours of a_2 before, we must have $b_2 \neq a_1$ and $b_3 \neq a_2$. Let b_4 be the successor of b_3 on P. Now $P[a_1, b_2]$ and the arcs b_3b_2, b_3b_4 form the desired induced subdivision of A_3^+. From here on, we suppose that P is induced.

Suppose now that there is an arc e with an end $x \in V(P)$ and the other $y \in V(Q)$. Choose such an arc so that the sum of the lengths of $P[a_1, x]$ and $Q[a_3, y]$ is as small as possible. If e is from x to y we have $y \neq a_3$ because we removed all the in-neighbours of a_3, else e is from y to x and we have $x \neq a_1$ because we removed all the in-neighbours of a_1. In all cases, we get an induced subdivision of A_3^+ by taking the paths $P[a_1, x]$ and $Q[a_3, y]$ and the arcs a_3a_4, e. From here on, we suppose that there are no arcs with an end in $V(P)$ and the other in $V(Q)$.

The last case is when Q is not induced. Since Q has no forward chord, there is an arc uv in $G'\langle V(Q)\rangle$ such that u occurs after v on Q. Take such an arc b_3b_4 such that b_3 is as close as possible to a_2 (in Q). Observe that since we deleted all out-neighbours of a_2 before, we must have $b_3 \neq a_2$. Now $P, Q[b_3, a_2]$ and the arc b_3b_4 form the desired induced subdivision of A_3^+.

There are $O(nm)$ pairs (a_2, a_3a_4) and for each of them, we run an $O(m)$ flow algorithm (we just need to find a flow of value 2, say, by the Ford-Fulkerson method [1, Section 4.5.1]) and do some linear-time operations. Hence the complexity of the algorithm is $O(nm^2)$. □

One can check in polynomial time if there is an induced A_4^--subdivision: it is enough to enumerate all arcs t_2t_1, to delete t_1 and its neighbours except t_2, and to decide whether there exists in what remains an A_3^+ subdivision with origin t_2. One can also derive polynomial-time algorithm for finding induced subdivision of other oriented paths with four blocks.

Corollary 24. *Let P be an oriented path that can be obtained from A_4^- by subdividing the first arc and the second arc. One can check in time $O(n^{|P|-1}m^2)$ whether a given oriented graph contains an induced subdivision of P.*

Proof. Let R be the subdigraph of P formed by the first block of P and its second block minus the last arc. Let s be the last vertex of R. For each induced oriented path Q in the instance graph, isomorphic to R (there are $O(n^{|P|-3})$ of them), we delete $Q - s$, all vertices that have neighbours in $Q - s$ except s and detect an A_3^--subdivision with origin s. This will detect a P-subdivision if there is one. □
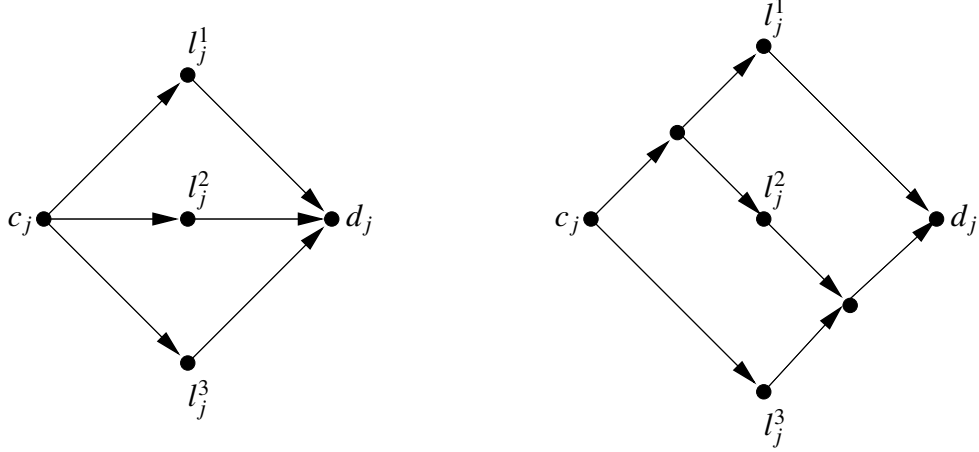
Figure 6: Left: clause gadget of $G_1(I)$. Right: clause gadget of $G_4(I)$.

6 Induced subdivisions of tournaments in oriented graphs

6.1 Induced subdivision of transitive tournaments

The transitive tournament on k vertices is denoted TT_k. We saw in Section 5.1 that Π'_{TT_3} is polynomial. The next result shows that Π'_{TT_k} is NP-complete for all $k \geq 4$.

Theorem 25. *For all $k \geq 4$, Π'_{TT_k} is NP-complete*

Proof. For a given instance I of 3-SAT, let $G_4(I)$ be the digraph we obtain from $G_1(I)$ by replacing each clause gadget C_j^1 by the modified one C_j^4 from Figure 6. Also for each variable, modify the gadget V_i^1 as follows: replace the path $a_i x_i v_i b_i$ by a path $a_i x_i^1 v_i^1 x_i^2 v_i^2 \ldots x_i^m v_i^m b_i$, and similarly the path $a_i \bar{x}_i \bar{v}_i b_i$ by a path $a_i \bar{x}_i^1 \bar{v}_i^1 \bar{x}_i^2 \bar{v}_i^2 \ldots \bar{x}_i^m \bar{v}_i^m b_i$. Then in $G_4(I)$ the links representing a variable x_i and a clause C_j that uses this variable are represented by arcs between vertices from the variable gadget with superscript j (as in Figure 2).

Recall that $G_1(I)$ has an induced directed (a,b)-path if and only if I is satisfiable. It is easy to see that the same holds for $G_4(I)$. Note that in $G_4(I)$ no vertex has in- or out-degree larger than 2.

Given an instance I of 3-SAT we form the digraph $G_4^k(I)$ from $G_4(I)$ and a copy of TT_k (with vertices $v_1, v_2, \ldots, v_k$ and arcs $v_i v_j$, $1 \leq i < j \leq k$) by deleting the arc $v_1 v_k$ and adding the arcs $v_1 a, b v_k$. We claim that $G_4^k(I)$ contains an induced subdivision of TT_k if and only if $G_4(I)$ has an induced directed (a,b)-path which is if and only if I is satisfiable.

Clearly, if I is satisfiable, we may use the concatenation of an induced directed (a,b)-path in $G_4(I)$ with $v_1 a$ and $b v_k$ in place of $v_1 v_k$ to obtain an induced TT_4-subdivision in $G_4^k(I)$.

Conversely, suppose that $G_4^k(I)$ contains an induced subdivision of TT_k and let $h(v_i)$, $1 \leq i \leq k$, denote the image of v_i in some fixed induced subdivision H of TT_k. Then we must have $h(v_1) = v_1$ and $h(v_k) = v_k$, because $G_4(I)$ does not contain any vertex of out-degree $k-1$ or in-degree $k-1$ because $k \geq 4$. For all i, $1 < i < k$, the vertex $h(v_i)$ could not be in $V(G_4(I))$ since otherwise there must be either two disjoint directed (v_i, v_k)-paths to v_k or two disjoint directed (v_1, v_i)-paths. This is impossible because there is no directed (v_i, v_k)-path in $G_4^k(I) \setminus b v_k$ and no directed (v_1, v_i)-path in $G_4^k(I) \setminus v_1 a$. Hence $h(v_i) = v_i$ for all $1 \leq i \leq k$ and so it is clear that we have an induced directed (a,b)-path in $G_4(I)$, implying that I is satisfiable. $\square$

In the proof above we used that the two vertices v_1, v_k cannot be mapped to vertices of $G_4(I)$, the fact that the connectivity between these and the other vertices is too high to allow any of these

15

to be mapped to vertices of $G_4(I)$ and finally we could appeal to the fact that $G_4(I)$ has an induced directed (a,b)-path if and only if I is satisfiable. Refining this argument it is not difficult to see that the following holds where a (z,X)-*path* is a path whose initial vertex is z and whose last vertex belongs to X.

Theorem 26. *Let $D = (V,A)$ be a digraph and let X (resp. Y) be the subset of vertices with out-degree (resp. in-degree) at least 3 and let $Z = V \setminus (X \cup Y)$ (note that $X \cap Y \neq \emptyset$ is possible and also $Z = \emptyset$ is possible). Suppose that for every $z \in Z$ the digraph D contains either two internally disjoint (X,z)-paths or two internally disjoint (z,Y)-paths. Then Π'_D is NP-complete.*

Proof. (Sketch) For a given instance I of 3-SAT we form the digraph $G'_4(I)$ from D by replacing one arc uv with at least one of its endvertices in $X \cup Y$ by $G_4(I)$ and the arcs ua, bv. Again it is clear how to obtain an induced subdivision of D in $G'_4(I)$ when I is satisfiable. Let us now assume that $G'_4(I)$ contains an induced subdivision D' of D. Let $\{h(v)|v \in V\}$ be the vertices corresponding to the vertices of V in the subdivision. For degree reasons, none of the vertices in $X \cup Y$ can have $h(v) \in V(G_4(I))$ and because of connectivity, none of the vertices of Z can have $h(z) \in V(G_4(I))$ because there is only one arc entering and leaving $V(G_4(I))$ in $G'_4(I)$. Thus $\{h(v)|v \in V\} = V$ (possibly with $h(v) \neq v$ for several vertices). However, since we deleted the arc uv and replaced it by $G_4(I)$ and the arcs ua, bv, it follows that $G'_4(I)$ and so $G_4(I)$ contains an induced directed (a,b)-path, implying that I is satisfiable. $\square$

6.2 Induced subdivision of the strong tournament on 4 vertices

Let ST_4 be the unique strong tournament of order 4. It can be seen has a directed cycle $\alpha\gamma\beta\delta\alpha$ together with two *chords* $\alpha\beta$ and $\gamma\delta$. The aim of this section is to show that Π'_{ST_4} is NP-complete.

An (x,y_1,y_2)-*switch* is the digraph with vertex set $\{x,z,y_1,y_2\}$ and edge set $\{xz,xy_1,zy_1,zy_2,y_2y_1\}$. See Figure 7.

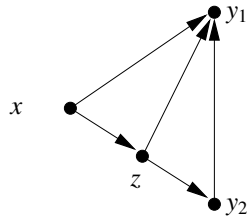Figure 7: An (x,y_1,y_2)-switch.

A *good* (x,y_1,y_2)-*switch* in a digraph D is an induced switch Y such that all the arcs entering Y have head x and all arcs leaving Y have tail in $\{y_1,y_2\}$.

Lemma 27. *Let Y be a good (x,y_1,y_2)-switch in a digraph D. Then every induced subdivision S of ST_4 in D intersects Y on either the path (x,y_1), the path (x,z,y_2), or the empty set.*

Proof. Suppose for a contradiction, that $y_2y_1 \in E(S)$. Then S must contain the unique in-neighbour z of y_2 and the unique in-neighbour x of z. Hence y_1 has in-degree 3 in S, a contradiction.

Suppose for a contradiction, that $zy_1 \in E(S)$. Then S must contain x the unique in-neighbour of z. Hence xy_1 is a chord of S and so z must have degree 3 in S. Thus $y_2 \in V(S)$ and y_1 has in-degree 3 in S, a contradiction. $\square$

Theorem 28. Π'_{ST_4} *is NP-complete.*

16

Proof. Reduction from 3-SAT. Let I be an instance of 3-SAT with variables $x_1, x_2, \ldots, x_n$ and clauses $C_1, C_2, \ldots, C_m$. We first create a variable gadget V_i^5 for each variable x_i, $i = 1, 2, \ldots, n$ and a clause gadget C_j^5 for each clause C_j, $j = 1, 2, \ldots, m$ as shown in Figure 8. Then we form the digraph $G_5(I)$ as follows: Form a chain U of variable gadgets by adding the arcs $b_i a_{i+1}$ for $i = 1, 2, \ldots, n-1$ and a chain W of clause gadgets by adding the arcs $d_j c_{j+1}$, $j = 1, 2, \ldots, m-1$. Add the arcs $aa_1, b_n b, cc_1, t_m d$. For each clause C, we connect the three literal vertices of the gadget for C to the variable gadgets for variables occuring as literals in C in the way indicated in Figure 9.
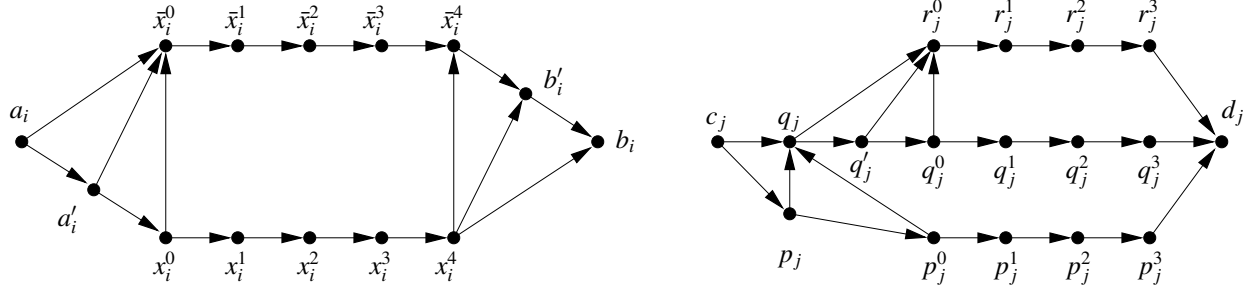

Figure 8: The variable gadget V_i^5 (left) and the clause gadget C_j^5 (right).
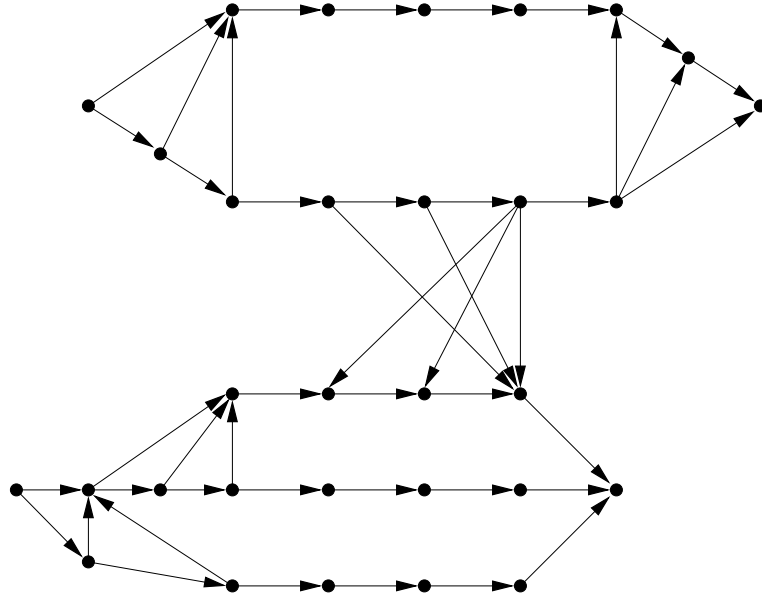

Figure 9: Connections between a clause gadget and a variable gadget in $G_5(I)$. Only the connection for one variable gadget and one clause gadget is shown and the general strategy for connecting variable and clause gadgets is the same as in $G_1(I)$ (Figure 2).

We denote by X_i the path $a_i a_i' x_i^0 x_i^1 x_i^2 x_i^3 x_i^4 b_i$, by $\bar{X}_i$ the path $a_i \bar{x}_i^0 \bar{x}_i^1 \bar{x}_i^2 \bar{x}_i^3 \bar{x}_i^4 b_i' b_i$, by P_j the path $c_j p_j p_j^0 p_j^1 p_j^2 p_j^3 d_j$, by Q_j the path $c_j q_j q_j' q_j^0 q_j^1 q_j^2 q_j^3 d_j$, and by R_j the path $c_j q_j r_j^0 r_j^1 r_j^2 r_j^3 d_j$.

Similarly to the proof of Lemma 7, one can check that I is satisfiable if and only if there are two induced disjoint directed (a,b)- and (c,d)-paths in $G_5(I)$.

Let $G_5^*(I)$ be the digraph obtained from $G_5(I)$ by adding the edges ac, cb, bd and da. Observe that $G_5^*(I) \setminus da$ is acyclic.

Let us prove that $G_5^*(I)$ contains an induced ST_4-subdivision if and only if I is satisfiable.

If I is satisfiable, then in $G_5(I)$ there are two induced disjoint directed (a,b)- and (c,d)-paths. The union of these paths and the directed cycle $acbd$ is an induced ST_4-subdivision in $G_5^*(I)$.

Conversely, assume that $G_5^*(I)$ contains an induced subdivision S of ST_4. For sake of simplicity (and with a slight abuse of notation), we will denote the vertices of S corresponding to α, β, γ and δ by the same names. Let T_1 and T_2 be the paths corresponding to the chord $\alpha\beta$ and $\gamma\delta$ respectively in S and let C be the directed cycle corresponding to $\alpha\gamma\beta\delta\alpha$. Observe that the ends of T_1 and T_2 must alternate on C.

Notice that the subdigraphs induced by the sets $\{a_i, a_i', x_i^0, \bar{x}_i^0\}$, $1 \le i \le n$, $\{c_j, p_j, p_j^0, q_j\}$ and $\{q_j, q_j', q_j^0, r_j^0\}$ are good switches. In addition, the subdigraphs induced by the sets $\{b_i, b_i', x_i^4, \bar{x}_i^4\}$, $1 \le i \le n$, are the converse of good switches. Hence Lemma 27 (and its converse) imply the following proposition.

Claim 28.1.

(i) For $1 \le i \le n$, if $a_i \in V(S)$, then exactly one of the two paths (a_i, a_i', x_i^0) and $(a_i, \bar{x}_i^0)$ is in S.

(ii) For $1 \le i \le n$, if $b_i \in V(S)$, then exactly one of the two paths $(\bar{x}_i^4, b_i', b_i)$ and (x_i^4, b_i) is in S.

(iii) For $1 \le j \le m$, if $c_j \in V(S)$, then exactly one of the three paths (c_j, p_j, p_j^0), (c_j, q_j, q_j', q_j^0) and (c_j, q_j, r_j^0) is in S.

Since $G_5^*(I) \setminus da$ is acyclic, C must contain the arc da. Moreover since there is no arc with tail in some clause gadget and head in some variable gadget, C contains at most one arc with tail in some variable gadget and head in some clause gadget.

Claim 28.2. *For any $1 \le i \le n$ and any $1 \le j \le m$, the cycle C contains no arc with tail in $\{x_i^3, \bar{x}_i^3\}$ and head in $\{p_j^1, q_j^1, r_j^1\}$.*

Proof. Assume for a contradiction that C contains such an arc $y_i^3 l_j^1$. Then since l_j^1 and l_j^2 have out-degree 1 then C must also contain l_j^2 and l_j^3. Thus, in S, y_i^3 has out-degree 3 in S, a contradiction. $\square$

Claim 28.3. *For any $1 \le i \le n$ and any $1 \le j \le m$ the cycle C contains no arc with tail in $\{x_i^3, \bar{x}_i^3\}$ and head in $\{p_j^3, q_j^3, r_j^3\}$.*

Proof. Assume for a contradiction that C contains such an arc $y_i^3 l_j^3$. Then since y_i^3 and y_i^2 have in-degree 1 then C must also contain y_i^2 and y_i^1. Thus, in S, l_j^3 has in-degree 3 in S, a contradiction. $\square$

Claim 28.4. *For any $1 \le i \le n$ and any $1 \le j \le m$ then C contains no arc with tail in $\{x_i^3, \bar{x}_i^3\}$ and head in $\{p_j^2, q_j^2, r_j^2\}$.*

Proof. Assume for a contradiction that C contains such an arc $y_i^3 l_j^2$. The vertex l_j^2 has a unique out-neighbour l_j^3 which must be in C. It follows that $y_i^3 l_j^3$ corresponds to one of the chords $\alpha\beta$ or $\gamma\delta$. Thus l_j^2 must have degree 3 in S. It follows that l_j^1 is in $V(S)$ and so y_i^3 has out-degree 3 in S, a contradiction. $\square$

Claim 28.5. *For any $1 \le i \le n$ and any $1 \le j \le m$ the cycle C contains no arc with tail in $\{x_i^2, \bar{x}_i^2\}$ and head in $\{p_j^3, q_j^3, r_j^3\}$.*

Proof. Assume for a contradiction that C contains such an arc $y_i^2 l_j^3$. The vertex y_i^2 has a unique in-neighbour y_i^1 which must be in C. It follows that $y_i^1 l_j^3$ corresponds to one of the chords $\alpha\beta$ or $\gamma\delta$. Thus y_i^2 must have degree 3 in S. It follows that y_i^3 is in $V(S)$ and so l_j^3 has in-degree 3 in S, a contradiction. $\square$

Claim 28.6. *For any $1 \leq i \leq n$ and any $1 \leq j \leq m$ the cycle C contains no arc with tail in $\{x_i^1, \bar{x}_i^1\}$ and head in $\{p_j^3, q_j^3, r_j^3\}$.*

Proof. Assume for a contradiction that C contains such an arc $y_i^1 l_j^3$. Without loss of generality $y_i^1 = x_i^1$.

By the remark after Claim 28.1 this is the only arc from a variable gadget to a clause gadget. Furthermore, we have that b is not on C.

Thus, by Claim 28.1, for every $1 \leq k < i$, the intersection of C and V_k^4 is either X_k or $\bar{X}_k$, and for every $j < l \leq m$, the intersection of C and C_j^5 is either P_j, Q_j or R_j.

Consider $y \in \{\alpha, \beta\}$. It is on C and has outdegree 2. On the other hand, applying Claim 28.1 we see that the following must hold as none of these vertices can belong to S and at the same time have two of their out-neighbours in S:

- $y \notin \cup_{1 \leq j \leq m}\{c_j, p_j, q_j, q_j', q_j^0\}$,

- $y \notin \cup_{k \neq i}\{a_k, a_k', x_k^0, x_k^4\}$ and

- $y \notin \{a_i, a_i', x_i^0, x_i^4\}$.

By Claims 28.2-28.5, we have $y \notin \{x_i^2, x_i^3\}$ and since b is not on C we also have $y \neq b$. If $y = x_i^1$, then using that yl_j^3 is and arc of C we get a contradiction because $x_i^2 l_j^3$ is an arc (so we cannot obtain an induced copy of S using both arcs $yl_j^3, x_i^2 l_j^3$). Hence (as y was any of α, β) we have $a = \alpha = \beta$, a contradiction. $\square$

Claim 28.7. *$C = acbda$.*

Proof. Suppose not. Then by the above claims, C either does not intersect the clause gadget and intersect all the variable ones or does not intersect the variable gadget and intersect all the gadget ones. In both cases, similarly to the proof of Claim 28.5, one shows that $a = \alpha = \beta$, a contradiction. $\square$

Since $C = acbda$ and by construction of $G_5^*(I)$, T_1 and T_2 are two induced disjoint path in $G_5(I)$ and so I is satisfiable. $\square$

7 Remarks and open problems

It would be nice to have results proving a full dichotomy between the digraphs D for which Π_D (resp. Π_D') is *NP*-complete and the ones for which it is polynomial-time solvable. Regarding Π_D, Conjecture 14 gives us what the dichotomy should be. But for Π_D' we do not know yet.

A useful tool to prove such a dichotomy would be the following conjecture.

Conjecture 29. If D is a digraph such that Π_D (resp. Π_D') is NP-complete, then for any digraph D' that contains D as an induced subdigraph, $\Pi_{D'}$ (resp. $\Pi_{D'}'$) is NP-complete.

We were able to settle the complexity of Π_D' when D is a directed cycle, a directed path, or some paths with at most four blocks. The following problems are perhaps the natural next steps.

Problem 30. What is the complexity of Π_D' when D is an oriented cycle which is not directed?

Problem 31. What is the complexity of Π_D' when D is an oriented path which is not directed?

Note that the approach used above to find an induced subdivision of A_4^- relied on the fact that one can check in polynomial time (using flows) whether a digraph contains internally disjoint (x,z)-, (y,z)-paths for prescribed distinct vertices x,y,z. If we want to apply a similar approach for A_5^-, then for prescribed vertices x,y,z,w we need to be able to check the existence of internally disjoint paths P,Q,R such that P is an (x,y)-path, Q is a (z,y)-path and R is a (z,w)-path such that these paths are induced and have no arcs between them. However, the problem of deciding just the existence of internally disjoint paths P,Q,R with these prescribed ends is NP-complete by the result of Fortune et al. [6]. Thus we need another approach to obtain a polynomial-time algorithm (if one exists).
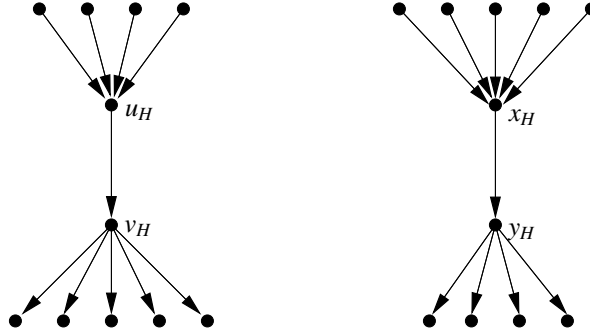

Figure 10: The digraph H with specified vertices u_H, v_H, x_H, y_H.

It seems that little is known about detecting a subdivision of some given digraph D as a subgraph (possibly non-induced). This leads us to the following problem:

Problem 32. When D is fixed directed graph, what is the complexity of deciding whether a given digraph G contains a D-subdivision as a subgraph?

The following shows that the problem above can be NP-complete.

Theorem 33. *Let H be the digraph in Figure 10. It is NP-complete to decide whether a given digraph G contains an H-subdivision.*

Proof. By the classical result of Fortune, Hopcroft and Wyllie [6], the so-called 2-linkage problem (given a digraph and four distinct vertices u,v,x,y; does G contain a pair of vertex-disjoint paths P,Q so that P is a directed (u,v)-path and Q is a directed (x,y)-path?) is NP-complete. By inspecting the proof (see [1, Section 10.2]) it can be seen that the problem is NP-complete even when G has maximum in- and out-degree at most 3. Given an instance G of the 2-linkage problem with maximum in- and out-degree at most 3 and a copy of H we form a new digraph G_H by identifying the vertices $\{u,v,x,y\}$ with $\{u_H,v_H,x_H,y_H\}$ in that order. Clearly, if G has disjoint directed (u,v), (x,y)-paths, then we can use these to realize the needed paths from u_H to v_H and from x_H to y_H (and all other paths are the original arcs of H). Conversely, suppose there is a subdivision H^* of H in G_H. For every $v \in \{u_H, v_H, x_H, y_H\}$, let us denote by v^* the vertex corresponding to v in H^*. Since $d^-(u_H) = 4, d^+(v_H) = 5, d^-(x_H) = 5, d^+(y_H) = 4$ in G_H, we have $u_H^* = u_H$, $v_H^* = v_H$, $x_H^* = x_H$, and $y_H^* = y_H$. Thus the two disjoint directed (u_H^*, v_H^*)- x_H^*, y_H^*)-paths in H^* are disjoint directed (u,v), (x,y)-paths in G implying that G is a 'yes'-instance. $\square$

Finally, we would like to point out that in all detection problems about induced digraphs, backward arcs of paths play an important role, especially in NP-completeness proofs. Also, these backward arcs make all "connectivity-flavoured" arguments fail: when two vertices x,y are given, it is not

possible to decide whether *x* can be linked to *y*. So, maybe another notion of induced subdigraph containment would make sense: chords should be kept forbidden between the different directed paths that arise from subdividing arcs, but backward arcs inside the paths should be allowed.

Acknowledgement

References

[1] J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer Monographs in Mathematics. Springer Verlag, London, 2008.

[2] J. Bang-Jensen and M. Kriesell Disjoint directed and undirected paths and cycles in digraphs *Theoretical Computer Science* 410, 5138–5144, 2009.

[3] D. Bienstock. On the complexity of testing for odd holes and induced odd paths. *Discrete Math.* 90:85–92, 1991. See also Corrigendum by B. Reed, *Discrete Math.* 102:109, 1992.

[4] M. Chudnovsky and P. Seymour. Excluding induced subgraphs. In *Surveys in Combinatorics*, volume 346, pages 99–119. London Mathematical Society Lecture Notes Series, 2007.

[5] M. Chudnovsky and P. Seymour. The three-in-a-tree problem. *Combinatorica*, 2007. To appear.

[6] S. Fortune, J.E. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10:111–121, 1980.

[7] B. Lévêque, D. Lin, F. Maffray, and N. Trotignon. Detecting induced subgraphs. *Discrete Applied Mathematics*, 157:3540–3551, 2009.

[8] Y. Kobayashi. Induced disjoint paths problem in a planar digraph *Discrete Applied Mathematics* 157, 3231–3238, 2009.

[9] N. Robertson and P.D. Seymour. Graph minors. XIII. The disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63:65–110, 1995.