



HAL
open science

Finding a subdivision of a digraph

Jørgen Bang-Jensen, Frédéric Havet, Ana Karolinna Maia de Oliveira

► **To cite this version:**

Jørgen Bang-Jensen, Frédéric Havet, Ana Karolinna Maia de Oliveira. Finding a subdivision of a digraph. *Theoretical Computer Science*, 2015, 562, pp.20. hal-01111374

HAL Id: hal-01111374

<https://inria.hal.science/hal-01111374>

Submitted on 23 Oct 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Finding a subdivision of a digraph

Jørgen Bang-Jensen^{1,2}, Frédéric Havet³, A. Karolinna Maia³

Abstract

We consider the following problem for oriented graphs and digraphs: Given a directed graph D , does it contain a subdivision of a prescribed digraph F ? We give a number of examples of polynomial instances, several NP-completeness proofs as well as a number of conjectures and open problems.

1. Introduction

Many interesting classes of graphs are defined by forbidding induced subgraphs, see [7] for a survey. This is why the detection of several kinds of induced subgraphs is interesting, see [15] 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-time solvable or NP-complete according to 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-time 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 undirected 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 [18] (for a short explanation of this see e.g. [3]). 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 [15] 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 $K_{2,3}$ which can be performed in $O(n^{11})$ time by Chudnovsky and Seymour’s three-in-a-tree algorithm, see [8]. 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 two triangles, and in many other cases.

¹Department of Mathematics and Computer Science, University of Southern Denmark, Odense DK-5230, Denmark (email: jbj@imada.sdu.dk).

²Most of this work was done while J. Bang-Jensen visited Projet Mascotte, I3S (CNRS, UNSA) and INRIA, Sophia Antipolis whose hospitality and financial support is gratefully acknowledged.

³Projet Mascotte, I3S (CNRS, UNSA) and INRIA, Sophia Antipolis. Partly supported by ANR Blanc AGAPE and CAPES/Brazil. (email:[frederic.havet, karol.maia]@inria.fr)

1 When we move to digraphs, the situation becomes more complicated, even for the sub-
2 digraph containment relation. In this paper, by digraph we mean a simple digraph, that is a
3 digraph with no parallel arcs nor loops. Sometimes however, multiple arcs are possible. In
4 such cases, we write multidigraph. We rely on [1] for classical notation and concepts. A few
5 things need to be stated here though. Unless otherwise stated the letters n and m will always
6 denote the number of vertices and arcs (edges) of the input digraph (graph) of the problem
7 in question. By *linear time*, we mean $O(n + m)$ time. If D is a digraph, then we denote by
8 $UG(D)$ the underlying (multi)graph of D , that is, the (multi)graph we obtain by replacing
9 each arc by an edge. A digraph D is *connected* if $UG(D)$ is a connected graph. If xy is an arc
10 from x to y , then we say that x *dominates* y . When H, H' are digraphs we denote by $H + H'$
11 the disjoint union of H and H' (no arcs between disjoint copies of these).

12 A *subdivision of a digraph F* , also called an *F -subdivision*, is a digraph obtained from F
13 by replacing each arc ab of F by a directed (a, b) -path.

14 In this paper, we consider the following problem for a fixed digraph F .

15 F -SUBDIVISION

16 Input: A digraph D .

17 Question: Does D contain a subdivision of F as a subgraph?

18
19 In [2] the problem INDUCED- F -SUBDIVISION of finding an induced subdivision of a
20 prescribed digraph F in a given digraph D was studied. It turns out that here there is a
21 big difference in the complexity of the problem depending on whether or not D is an ori-
22 ented graph or it may contain 2-cycles. In the latter case INDUCED- F -SUBDIVISION is
23 NP-complete for every oriented digraph F which is not the disjoint union of spiders (see
24 definition of these digraphs below) and it was conjectured that INDUCED- F -SUBDIVISION
25 is NP-complete unless F is the disjoint union of spiders and at most one 2-cycle.

26 Let $x_1, x_2, \dots, x_k, y_1, y_2, \dots, y_k$ be distinct vertices of a digraph D . A k -linkage from
27 (x_1, x_2, \dots, x_k) to (y_1, y_2, \dots, y_k) in D is a system of disjoint directed paths P_1, P_2, \dots, P_k such
28 that P_i is an (x_i, y_i) -path in D .

29 Similarly to the situation for undirected graphs, the D -SUBDIVISION problem is related
30 to the following k -LINKAGE problem.

31 k -LINKAGE

32 Input: A digraph D and $2k$ distinct vertices $x_1, x_2, \dots, x_k, y_1, y_2, \dots, y_k$.

33 Question: Is there a k -linkage from (x_1, x_2, \dots, x_k) to (y_1, y_2, \dots, y_k) in D ?

34 However, contrary to graphs, unless $P=NP$, k -LINKAGE cannot be solved in polynomial
35 time in general digraphs. Fortune, Hopcroft and Wyllie [10] showed that already 2-LINKAGE
36 is NP-complete. Using this result, we show that for lots of F , the F -SUBDIVISION problem
37 is NP-complete. We also give some digraphs F for which we prove that F -SUBDIVISION is
38 polynomial-time solvable. We believe that there is a dichotomy between NP-complete and
39 polynomial-time solvable instances.

40 **Conjecture 1.** For every digraph F , the F -SUBDIVISION problem is polynomial-time solv-
41 able or NP-complete.

1 To prove such a conjecture, a first idea would be to try to establish for any digraph
 2 G and subdigraph F , that if F -SUBDIVISION is NP-complete, then G -SUBDIVISION is
 3 also NP-complete, and conversely, if G -SUBDIVISION is polynomial-time solvable, then
 4 F -SUBDIVISION is polynomial-time solvable. However, these two statements are false as
 5 shown by the two digraphs depicted Figure 1. The NP-completeness of A -SUBDIVISION
 6 follows Theorem 12. The fact that B -SUBDIVISION is polynomial-time solvable is proved in
 7 Theorem 27.

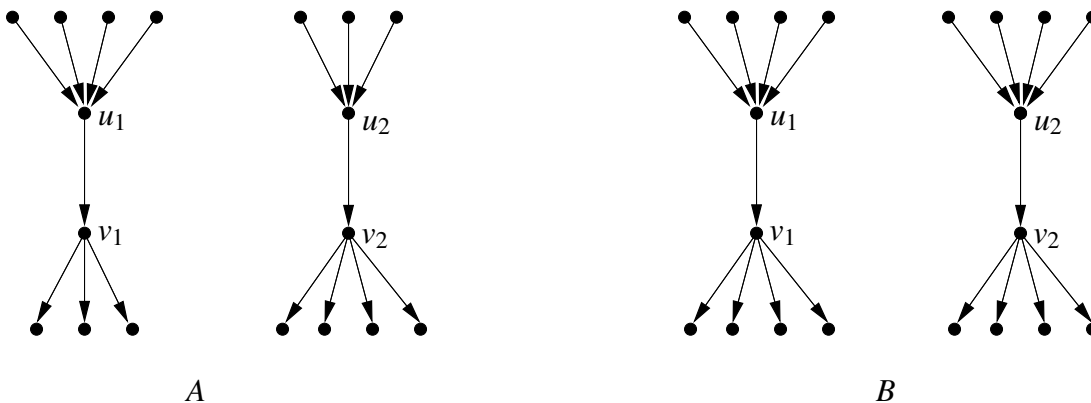


Figure 1: Digraphs A and B such that A is a subdigraph of B , A -SUBDIVISION is NP-complete, and B -SUBDIVISION is polynomial-time solvable.

8 The paper is organized as follows. We start by giving some general lemmas which al-
 9 low to extend NP-completeness results of F -SUBDIVISION for some digraphs F to much
 10 larger classes of digraphs. Next we give a powerful tool, based on a reduction from the
 11 NP-complete 2-linkage problem in digraphs, which can be applied to conclude the NP-
 12 completeness of F -SUBDIVISION for the majority of all digraphs F . We then describe
 13 different algorithmic tools for proving polynomial-time solvability of certain instances of
 14 F -SUBDIVISION. We first give some easy brute force algorithms, then algorithms based
 15 on maximum-flow calculations and finally algorithms based on handle decompositions of
 16 strongly connected digraphs. After this we give a number of classes of digraphs for which the
 17 F -SUBDIVISION is polynomial-time solvable for every F . Then we treat F -SUBDIVISION
 18 when F belongs to some special classes of digraphs such as disjoint unions of cycles, wheels,
 19 fans, transitive tournaments, oriented paths or cycles or F has at most 3 vertices. Finally, we
 20 conclude with some open problems, including an interesting conjecture due to Seymour,
 21 which if true would imply some of the polynomial cases treated in this paper.

22 2. Some general lemmas

23 **Lemma 2.** *Let F_1 and F_2 be two digraphs.*

- 24 (i) *If F_1 -SUBDIVISION is NP-complete, then $(F_1 + F_2)$ -SUBDIVISION is NP-complete.*
 25 (ii) *If $(F_1 + F_2)$ -SUBDIVISION is polynomial-time solvable, then F_1 -SUBDIVISION is po-
 26 lynomial-time solvable.*

1 *Proof.* Let D be a digraph. We shall prove that D contains an F_1 -subdivision if and only if
 2 $D + F_2$ contains an $(F_1 + F_2)$ -subdivision.

3 Clearly if D contains an F_1 -subdivision S , then $S + F_2$ is an $(F_1 + F_2)$ -subdivision in
 4 $D + F_2$.

5 Conversely, assume that $D + F_2$ contains an $(F_1 + F_2)$ -subdivision $S = S_1 + S_2$ with S_1 an
 6 F_1 -subdivision and S_2 an F_2 -subdivision. Let us consider such an $(F_1 + F_2)$ -subdivision that
 7 maximizes the number of connected components⁴ of F_2 that are mapped (in S) into F_2 again
 8 (notice that since there are no arcs between D and F_2 in $D + F_2$, in the subdivision S every
 9 component of S_2 will either be entirely inside F_2 or entirely inside D). We claim that $S_2 = F_2$.
 10 Indeed suppose that some component T of S_2 is in D . Let C be the component of F_2 of which
 11 T is the subdivision. Let $U = S \cap C$. Then T contains a subdivision U' of U (because it is a
 12 subdivision of all of C). Hence replacing U by U' and T by C in S , we obtain a subdivision
 13 with one more component mapped on itself, a contradiction.

14 Hence $S_2 = F_2$, and so D contains S_1 which is an F_1 -subdivision. □

15 **Lemma 3.** *Let F_1 and F_2 be two digraphs such that F_1 is strongly connected and F_2 contains
 16 no F_1 -subdivision. Let F be obtained from F_1 and F_2 by adding some arcs with tail in $V(F_1)$
 17 and head in $V(F_2)$.*

18 (i) *If F_1 -SUBDIVISION is NP-complete, then F -SUBDIVISION is NP-complete.*

19 (ii) *If F -SUBDIVISION is polynomial-time solvable, then F_1 -SUBDIVISION is polynomial-
 20 time solvable.*

21 *Proof.* We shall prove that a digraph D contains an F_1 -subdivision if and only if $D \mapsto F_2$
 22 contains an F -subdivision, where $D \mapsto F_2$ is obtained from $D + F_2$ by adding all possible
 23 arcs from $V(D)$ to $V(F_2)$.

24 It is easy to see that if D contains an F_1 -subdivision S , then $S + F_2$ together with some
 25 subset of the arcs from D to F_2 is an F -subdivision in $D \mapsto F_2$. Conversely, if $D \mapsto F_2$ contains
 26 an F subdivision S^* , then, since F_1 is strongly connected, the part of S^* forming a subdivision
 27 of F_1 has to lie entirely inside D or F_2 . Since F_2 contains no F_1 -subdivision, the subdivision
 28 of F_1 has to be inside D and hence we get that D has an F_1 -subdivision. □

29 It is useful to look at Figure 1 again and notice that the digraphs A, B show that we need
 30 the assumption that F_1 is strongly connected in Lemma 3 (and the analogous version where
 31 the roles of F_1 and F_2 are interchanged).

32 A digraph D is *robust* if it is strongly connected and $UG(D)$ is 2-connected.

33 **Lemma 4.** *Let F_1 and F_2 be two digraphs such that F_1 is robust and F_2 contains no F_1 -
 34 subdivision. Let F be obtained from F_1 and F_2 by identifying one vertex of F_1 with one vertex
 35 of F_2 .*

36 (i) *If F_1 -SUBDIVISION is NP-complete, then F -SUBDIVISION is NP-complete.*

37 (ii) *If F -SUBDIVISION is polynomial-time solvable, then F_1 -SUBDIVISION is polynomial-
 38 time solvable.*

⁴A connected component of a digraph H is a connected component of $UG(H)$.

1 *Proof.* Given a digraph D we form the digraph D^{F_2} by fixing one vertex x in F_2 and adding
2 $|V(D)|$ disjoint copies of F_2 such that the i th copy has its copy of x identified with the i th
3 vertex of D . It is easy to check that D^{F_2} contains an F -subdivision if and only if D contains
4 an F_1 -subdivision. This follows from the fact that F_2 contains no F_1 -subdivision and $UG(F_1)$
5 is 2-connected. \square

6 **Lemma 5.** *Let F be a digraph in which every vertex v satisfies $\max\{d^+(v), d^-(v)\} \geq 2$, and*
7 *let S be a subdivision of F .*

8 (i) *If F -SUBDIVISION is NP-complete, then S -SUBDIVISION is NP-complete.*

9 (ii) *If S -SUBDIVISION is polynomial-time solvable, then F -SUBDIVISION is polynomial-*
10 *time solvable.*

11 *Proof.* We shall prove a polynomial reduction from F -SUBDIVISION to S -SUBDIVISION.

12 Let D be an instance of F -SUBDIVISION and p be the length of a longest path in S
13 corresponding to an arc in F . Let D_p be the D -subdivision obtained by replacing every
14 arc of D by a directed path of length p . One easily checks that D has an F -subdivision
15 if and only if D_p has an S -subdivision. It follows from the fact that every vertex v corre-
16 sponding to one of F in S must be mapped onto a vertex corresponding to D in D_p because
17 $\max\{d^+(v), d^-(v)\} \geq 2$. \square

18 We believe that the condition $\max\{d^+(v), d^-(v)\} \geq 2$ for all $v \in V(F)$ is not necessary,
19 although it is in our proof.

20 **Conjecture 6.** Let F be a digraph, and let S be a subdivision of F .

21 (i) If F -SUBDIVISION is NP-complete, then S -SUBDIVISION is NP-complete.

22 (ii) If S -SUBDIVISION is polynomial-time solvable, then F -SUBDIVISION is polynomial-

23 time solvable.

24 3. General NP-completeness results

25 3.1. The tool

26 The following observations allow us to conclude that F -subdivision is “almost always”
27 NP-complete. We use an easy modification of the 2-linkage problem as the basis for these
28 proofs.

29 A vertex v is said to be *small* if $d^-(v) \leq 2$, $d^+(v) \leq 2$ and $d(v) \leq 3$. A non-small vertex
30 is called *big*.

31 **Theorem 7.** *The 2-LINKAGE problem is NP-complete even when restricted to digraphs with*
32 *no big vertices in which x_1 and x_2 are sources and y_1 and y_2 are sinks.*

33 *Proof.* Reduction from 2-LINKAGE in general digraphs.

34 An *out-arborescence* is the orientation of a tree in which all vertices have in-degree 1 ex-
35 cept one special vertex, called the *root*. A *switching out-arborescence* is an out-arborescence,
36 in which the root has out-degree 1, the leaves have out-degree 0 and all other vertices have
37 out-degree 2. A *switching in-arborescence* is the dual notion to out-arborescence.

Let D be a digraph and x_1, x_2, y_1, y_2 four vertices. Let D^* be the digraph obtained from D by deleting all the arcs entering x_1 and x_2 and all the arcs leaving y_1 and y_2 . Let $S(D)$ be the digraph obtained from D^* as follows. For every vertex v , replace all the arcs leaving v by a switching out-arborescence with root v and whose leaves corresponds to the out-neighbours of v in D^* , and replace all the arcs entering v by a switching in-arborescence with root v and whose leaves corresponds to the in-neighbours of v in D^* . It is clear that $S(D)$ has no big vertices and that x_1 and x_2 are sources and y_1 and y_2 are sinks. Furthermore, one checks easily that there is a 2-linkage from (x_1, x_2) to (y_1, y_2) in D if and only if there is a 2-linkage from (x_1, x_2) to (y_1, y_2) in $S(D)$. \square

3.2. A general NP-completeness theorem

For a digraph D , we denote by $B(D)$ the set of its big vertices. A *big path* in a digraph is a directed path whose endvertices are big and whose internal vertices all have in- and out-degree one in D (in particular an arc between two big vertices is a big path). Note also that two big paths with the same endvertices are necessarily internally disjoint.

The *big paths digraph* of D , denoted $BP(D)$, is the multidigraph with vertex set $V(D)$ in which there are as many arcs between two vertices u and v as there are big (u, v) -paths in D . By the remark above $BP(D)$ is well-defined and easy to construct in polynomial time given D .

Theorem 8. *Let F be a digraph. If F contains two arcs ab and cd whose endvertices are big vertices and such that $(BP(F) \setminus \{ab, cd\}) \cup \{ad, cb\}$ is not isomorphic to $BP(F)$, then F -SUBDIVISION is NP-complete.*

Proof. Reduction from 2-LINKAGE in digraphs with no big vertices in which x_1 and x_2 are sources and y_1 and y_2 are sinks.

Let D, x_1, x_2, y_1, y_2 be an instance of this problem. Let H be the digraph obtained from the disjoint union of $F \setminus \{ab, cd\}$ and D by adding the arcs ax_1, cx_2, y_1b , and y_2d . We claim that H has an F -subdivision if and only if D has a 2-linkage from (x_1, x_2) to (y_1, y_2) .

Clearly, if there is a 2-linkage P_1, P_2 in D , then the union of $F \setminus \{ab, cd\}$ and the paths $ax_1P_1y_1b$ and $cx_2P_2y_2d$ is a F -subdivision in H .

Conversely, suppose that H contains an F -subdivision S . Observe that in H , no vertex of D is big. Hence, since S has as many big vertices as F , F and S have the same set of big vertices.

Clearly, S contains as many big paths as F and thus there must be in D two disjoint directed paths between (x_1, x_2) and (y_1, y_2) . These two paths cannot be an (x_1, y_2) - and an (x_2, y_1) -path, for otherwise $(BP(F) \setminus \{ab, cd\}) \cup \{ad, cb\} = BP(S)$ would be isomorphic to $BP(F)$ since S is an F -subdivision. Hence, there is 2-linkage from (x_1, x_2) to (y_1, y_2) . \square

Remark 9. Observe that if $BP(F)$ has two arcs ab and cd which are consecutive (i.e. $b = c$) or contains an antidirected path (a, b, c, d) of length 3, then $(BP(F) \setminus \{ab, cd\}) \cup \{ad, cb\}$ is not isomorphic to $BP(F)$. Hence, by Theorem 8, F -SUBDIVISION is NP-complete.

Corollary 10. *If F is a digraph with no small vertices, then F -SUBDIVISION is NP-complete.*

Proof. If F has no small vertices, then $BP(F) = F$. Moreover if F does not contain two consecutive arcs, then $V(F)$ can be partitionned into two sets A and B such that all arcs in F have tail in A and head in B . In this case, F contains an antidirected path of length 3. So by Remark 9, the F -SUBDIVISION problem is NP-complete. \square

1 For many digraphs F , the condition of Theorem 8 is verified and so F -SUBDIVISION is
 2 NP-complete. However, there are graphs F that do not verify this condition but for which
 3 F -SUBDIVISION is NP-complete as we shall prove in the following subsection.

4 3.3. Dumbbells

5 An *oriented path* is an orientation of an undirected path. Let $P = (x_1, \dots, x_n)$ be an
 6 oriented path. If x_1x_2 is an arc, then P is an *out-path*, otherwise P is an *in-path*. In particular,
 7 if P is a directed path then it is an out-path. The *blocks* of P are the maximal directed subpaths
 8 of P . We often enumerate them from the origin to the terminus of the path. The number of
 9 blocks of P is denoted by $b(P)$.

10 A *dumbbell* is a digraph D with exactly two big vertices u and v which are connected
 11 by an induced oriented (u, v) -path P such that removing the internal vertices of P leaves
 12 a digraph with two connected components, one L containing u and one R containing the
 13 terminus v . The subdigraph L (resp. R) is the *left* (resp. *right*) *plate* of the dumbbell, vertex
 14 u is its *left clip*, vertex v its *right clip* and P its *bar*.

15 A *dumbbell set* is a disjoint union of dumbbells. In this subsection, we shall give some
 16 necessary conditions for F -SUBDIVISION to be NP-complete, F being a dumbbell set. In
 17 Subsection 5.3, we give particular cases when F -SUBDIVISION is polynomial-time solvable.

18 A pair of oriented paths (P, Q) is a *bad pair* if one of the following holds:

- 19 • P and Q are both directed paths;
- 20 • $\{b(P), b(Q)\} = \{1, 2\}$;
- 21 • P and Q are both out-paths and $\{b(P), b(Q)\} \in \{\{2, 2\}; \{2, 4\}\}$;
- 22 • P and Q are both in-paths and $\{b(P), b(Q)\} \in \{\{2, 2\}; \{2, 4\}\}$.

23 **Lemma 11.** *Let P and Q be two oriented paths. If (P, Q) is not a bad pair, then there exists*
 24 *$ab \in A(P)$ and $cd \in A(Q)$ such that the two oriented paths P' and Q' obtained from P and Q*
 25 *by replacing ab and cd by ad and cb verify $\{b(P), b(Q)\} \neq \{b(P'), b(Q')\}$.*

26 *Proof.* Let (P, Q) be a non-bad pair of paths. Without loss of generality, we may assume that
 27 $b(Q) \geq b(P)$. In particular this implies $b(Q) \geq 3$.

28 Assume that P is an out-path (resp. in-path) and Q is an in-path (resp. out-path). If
 29 $b(P) \geq 2$, then take ab as an arc of the first block of P and cd an arc of the first block of
 30 Q . Replacing ab and cd by ad and cb results necessarily in $b(P') = 1$ and $b(Q') = b(P) +$
 31 $b(Q) - 1$. If $b(P) = 1$, take ab as an arc of the first block of P and cd an arc of the second
 32 block of Q . Then $\{b(P'), b(Q')\} = \{2, b(Q) - 1\} \neq \{b(P), b(Q)\}$.

33 So we may assume that P and Q are both out-paths or both in-paths. Observe that this
 34 in particular implies that P and Q have an even number of blocks, because the opposite path
 35 (same digraph but starting from the terminus and ending at the origin) of an out-path with an
 36 odd number of blocks is an in-path with an odd number of blocks.

37 Take an arc ab of the first block of P and an arc cd of the second block of Q . Then one of
 38 P', Q' has two blocks and the other $b(P) + b(Q) - 2$ blocks. So if $\{b(P), b(Q)\} \neq \{2, b(P) +$
 39 $b(Q) - 2\}$, we have the result. Hence we may assume that $\{b(P), b(Q)\} = \{2, b(P) + b(Q) -$
 40 $2\}$, so $b(P) = 2$ because $b(Q) \geq 3$.

1 Hence $b(Q) \geq 6$, because (P, Q) is not bad. Take ab be an arc of the first block of P
2 and cd an arc of the third block of Q . Then one of P', Q' has four blocks and the other has
3 $b(P) + b(Q) - 4$ blocks, so we have the result. \square

4 If two digraphs D and D' are isomorphic, then we write $D \cong D'$. If they are not, then we
5 write $D \not\cong D'$.

6 **Theorem 12.** *Let F be a dumbbell set. Let D_1 and D_2 be two dumbbells of F , and for $i = 1, 2$,*
7 *let L_i, R_i, u_i, v_i and P_i be the left plate, right plate, left clip, right clip and bar of D_i . If one of*
8 *the following holds*

9 (a) (P_1, P_2) is not a bad pair,

10 (b) $L_1 \not\cong L_2, L_1 \not\cong R_2, R_1 \not\cong L_2$ and $R_1 \not\cong R_2$,

11 (c) P_1 and P_2 are both directed paths, $L_1 \not\cong L_2$ and $R_1 \not\cong R_2$,

12 (d) P_1 is a directed path and P_2 is an out-path (resp. in-path) with two blocks and $L_1 \not\cong L_2$
13 or $L_1 \not\cong R_2$ (resp. $R_1 \not\cong L_2$ or $R_1 \not\cong R_2$).

14 then F -SUBDIVISION is NP-complete.

15 *Proof.* By Lemma 2, it is sufficient to prove it when $F = D_1 + D_2$. The proof is very similar
16 to the one of Theorem 8. We give a reduction from 2-LINKAGE in digraphs with no big
17 vertices in which x_1 and x_2 are sources and y_1 and y_2 are sinks.

18 Let D, x_1, x_2, y_1, y_2 be an instance of this problem. Let ab be an arc of the bar of D_1
19 and cd be an arc of the bar of D_2 . Moreover, if (P_1, P_2) is not a bad pair, we choose ab
20 and cd as described in Lemma 11. Let H be the digraph obtained from the disjoint union of
21 $F \setminus \{ab, cd\}$ and D by adding the arcs ax_1, cx_2, y_1b , and y_2d . We can then show that H has
22 an F -subdivision if and only if D has a 2-linkage from (x_1, x_2) to (y_1, y_2) .

23 Clearly, if there is a 2-linkage R_1, R_2 in D , then the union of $F \setminus \{ab, cd\}$ and the paths
24 $ax_1R_1y_1b$ and $cx_2R_2y_2d$ is an F -subdivision in H .

25 Conversely, suppose that H contains an F -subdivision S . For each vertex x of F , we
26 denote by x^* the vertex corresponding to x in S and for any subdigraph G of F , we denote by
27 G^* the subdigraph of S corresponding to the subdivision of G .

28 In H , no vertex of D is big, so the sole big vertices of D are the clips of D_1 and D_2 .
29 Hence $\{u_1^*, v_1^*, u_2^*, v_2^*\} = \{u_1, v_1, u_2, v_2\}$. Now in S , the paths P_1^* and P_2^* connect big vertices.
30 For connectivity reasons these two paths must use $P_1 \setminus ab$ and $P_2 \setminus cd$. In particular, $(L_1 +$
31 $L_2 + R_1 + R_2)^*$ is a subdigraph of $L_1 + L_2 + R_1 + R_2$. So $(L_1 + L_2 + R_1 + R_2)^* = L_1 + L_2 +$
32 $R_1 + R_2$. So for any $G \in \{L_1, L_2, R_1, R_2\}$, the digraph G^* is isomorphic to G and is one of the
33 subdigraphs L_1, L_2, R_1 and R_2 .

34 Moreover $b(P_i^*) = b(P_i)$ for $i = 1, 2$. Hence, the subpaths of $P_1^* \cap D$ and $P_2^* \cap D$ must
35 be two disjoint directed paths in D , with origins in $\{x_1, x_2\}$ and terminus in $\{y_1, y_2\}$, for
36 otherwise $b(P_1^*) + b(P_2^*) > b(P_1) + b(P_2)$.

37 Let P'_1 and P'_2 be the oriented paths obtained from P_1 and P_2 by replacing ab and cd by
38 ad and cb . By construction, if there is no 2-linkage from (x_1, x_2) to (y_1, y_2) in D , then P_1^* and
39 P_2^* consist in a P'_1 -subdivision and a P'_2 -subdivision, and so $\{b(P'_1), b(P'_2)\} = \{b(P_1^*), b(P_2^*)\}$.

- 1 (a) If (P_1, P_2) is not a bad pair, then by our choice of ab and cd , $\{b(P'_1), b(P'_2)\} \neq \{b(P_1), b(P_2)\}$.
2 Since $b(P_1^*) = b(P_1)$ and $b(P_2^*) = b(P_2)$, there is a 2-linkage from (x_1, x_2) to (y_1, y_2) in
3 D .
- 4 (b) If $L_1 \not\cong L_2$ and $L_1 \not\cong R_2$, then $L_1^* \in \{L_1, R_1\}$. Similarly, if $R_1 \not\cong L_2$ and $R_1 \not\cong R_2$, then
5 $R_1^* \in \{L_1, R_1\}$. Hence P_1^* must go from u_1 to v_1 , and so $P_1^* \cap D$ is a directed (x_1, y_1) -
6 path. Hence there is a 2-linkage from (x_1, x_2) to (y_1, y_2) in D .
- 7 (c) If P_1 and P_2 are both directed paths, then $\{u_1^*, u_2^*\} = \{u_1, u_2\}$ as there are the origin of
8 P_1^* and P_2^* . Now, since $L_1 \not\cong L_2$, we have $L_1^* = L_1$ and $L_2^* = L_2$. Similarly, $R_1^* = R_1$ and
9 $R_2^* = R_2$. Hence, $P_1^* \cap D$ and $P_2^* \cap D$ form a 2-linkage from (x_1, x_2) to (y_1, y_2) in D .
- 10 (d) Assume that P_1 is a directed path and that P_2 is an out-path with two blocks. (The
11 proof is analogous when P_2 is an in-path with two blocks.)
- 12 Assume that $L_1 \not\cong L_2$. Then we can choose cd to be an arc of the first block of P_2 .
13 Necessarily, $v_1^* = v_1$ and $R_1^* = R_1$ since v_1^* is the only clip with out-degree 0 in $P_1^* \cup P_2^*$.
14 It follows that $L_1^* \in \{L_1, L_2\}$, and so $L_1^* = L_1$ because $L_1 \not\cong L_2$. Thus $P_1^* \cap D$ is a directed
15 (x_1, y_1) -path and there is a 2-linkage from (x_1, x_2) to (y_1, y_2) in D .
- 16 If $L_1 \not\cong R_2$, we get the result similarly by choosing cd to be an arc of the second block
17 of P_2 .

18 □

19 4. Easy polynomial-time solvable F -subdivision problems

20 There are digraphs F for which F -SUBDIVISION can be easily proved to be polynomial-
21 time solvable.

22 A *spider* is a tree obtained from disjoint directed paths by identifying one end of each
23 path into a single vertex. This vertex is called the *body* of the spider.

24 **Proposition 13.** *If F is the disjoint union of spiders, then F -SUBDIVISION can be solved in*
25 *$O(n^{|V(F)|})$ time.*

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

28 A natural question is to ask whether the problem remains polynomial-time solvable when
29 the spider F is no more fixed but specified in the input.

30 **Problem 14.** Is the following problem is polynomial-time solvable?

31 SPIDER-SUBDIVISION

32 Input: A spider F and a digraph D .

33 Question: Does D contain a subdivision of F ?

34 Similarly, one could ask if SPIDER-SUBDIVISION can be solved in FPT time when
35 parameterized by F , that is in $f(|V(F)|) \times n^c$ time, where f is a computable function and c
36 an absolute constant.

1 **Lemma 15.** *Let F_1 be a digraph and S a disjoint union of spiders. If F_1 -SUBDIVISION is*
2 *polynomial-time solvable, then $(F_1 + S)$ -SUBDIVISION is also polynomial-time solvable.*

3 *Proof.* For each set A of $|S|$ vertices, we check if the digraph $D\langle A \rangle$ induced by A contains S .
4 Then, if yes, we check if $D - A$ has an F -subdivision. \square

5 4.1. Subdivision of directed cycles

6 We denote by C_k the directed cycle of length k .

7 **Proposition 16.** *For every $k \geq 2$, C_k -SUBDIVISION can be solved in time $O(n^k \cdot m)$.*

8 *Proof.* For any $k \geq 2$, for k -tuple (x_1, x_2, \dots, x_k) , we check if (x_1, x_2, \dots, x_k) is a directed path
9 and if yes if there is a directed (x_k, x_1) -path in $D - \{x_2, \dots, x_{k-1}\}$. There are $O(n^k)$ k -tuples,
10 so this can be done in $O(n^k \cdot m)$ time. \square

11 The running time above is certainly not best possible. For example, when $k = 2$ or $k = 3$,
12 we can find linear-time algorithms.

13 **Proposition 17.** *C_2 -SUBDIVISION can be solved in linear time.*

14 *Proof.* A subdivision of the directed 2-cycle is a directed cycle. Hence a digraph has a C_2 -
15 subdivision if and only if it is not acyclic. Since one can check in linear time if a digraph is
16 acyclic or not [1, Section 2.1], C_2 -SUBDIVISION is linear-time solvable. \square

17 **Proposition 18.** *C_3 -SUBDIVISION can be solved in linear time.*

18 *Proof.* Let D be a digraph. If D has no directed 2-cycles, then D contains a C_3 -subdivision
19 if and only if it is not acyclic, which can be tested in linear time.

20 Assume now that D has some directed 2-cycles. Let H be the graph with vertex set $V(D)$
21 and edge-set $\{xy \mid (x, y, x) \text{ is a 2-cycle of } D\}$. The graph H can be constructed in linear time.
22 We first check, in linear time, if H contains a cycle. If H contains a cycle, then it has length
23 at least 3 and any if its two directed orientations is a directed cycle in D , so we return such a
24 cycle, certifying that D is a 'yes'-instance.

25 If not, then H is a forest. If there is any single arc uv (an arc which is not part of a 2-cycle)
26 in D such that both u and v belong to the same connected component of H , then it is easy to
27 produce a directed cycle of length at least 3 in D (following a path from u to v in H) so we
28 may assume that all single arcs go between different components in H . Now it is easy to see
29 that D contains a cycle of length at least 3 if and only if the digraph obtained by contracting
30 (into a vertex) each connected component of H in D has a directed cycle. In case we find
31 such a cycle, we can easily reproduce a directed cycle of length at least 3 in D . \square

32 If k is not fixed but specified in the input, it is NP-complete to decide if a digraph has a
33 directed cycle of length k because the Hamiltonian directed cycle is a particular case of it.
34 Gabow and Nie proved that it is FPT to decide if a graph has a cycle of length at least k .

35 **Theorem 19** (Gabow and Nie [11, 12]). *One can decide in $O(k^{3k} \cdot n \cdot m)$ time whether a*
36 *digraph contains a directed cycle of length at least k .*

37 **Problem 20.** For any fixed k , can we solve C_k -SUBDIVISION in linear time? In other words,
38 does there exists a computable function f such that one can decide in $O(f(k)(n + m))$ time
39 whether a digraph contains a directed cycle of length at least k ?

5. Polynomial-time solvable problems via flows

Recall that two paths are *internally disjoint* if they have no internal vertices in common. For any fixed k , there exist algorithms running in linear time that, given a digraph D and two distinct vertices x and y , returns k internally disjoint directed (x, y) -paths in D if some exist, or returns ‘no’ otherwise. Indeed, in such a particular case, any flow algorithm like Ford–Fulkerson algorithm for example, performs at most k incrementing-path searches, because it increments the flow by 1 each time, and we stop when the flow has value k , or if we find a cut of size less than k , which by Menger’s Theorem certifies that there do not exist k internally disjoint directed (x, y) -paths. Moreover each incrementing-path search consists in a search (usually Breadth-First Search) in an auxiliary digraph of the same size, and so is done in linear time. For more details, we refer the reader to the book of Ford and Fulkerson [?] or Chapter 7 of [5]. We call such an algorithm a *Menger algorithm*.

5.1. Subdivision of spindles

A (k_1, \dots, k_p) -spindle is the union of p pairwise internally disjoint directed (a, b) -paths P_1, \dots, P_p of respective length k_1, \dots, k_p . Vertex a is said to be the *tail* of the spindle and b its *head*.

Proposition 21. *If F is a spindle, then F -SUBDIVISION can be solved in $O(n^{|V(F)|(n+m)})$ time.*

Proof. Let F be a spindle with tail a and head b . Let a_1, \dots, a_p be the out-neighbours of a in F . An F -subdivision may be seen as an F -subdivision in which only the arcs aa_i , $1 \leq i \leq p$ are subdivided. The following algorithm takes advantage of this property.

Let D be a digraph. For each pair (S, a') where S is a set of $|V(F)| - 1$ vertices and a' a vertex of $D - S$, we first enumerate all the possible subdigraphs of $D \setminus S$ isomorphic to $F - a$ with a'_1, \dots, a'_p corresponding to a_1, \dots, a_p . We then check if, in $D - (S \cup \{a'_1, \dots, a'_p\})$, there exist p internally disjoint directed paths P_i , $1 \leq i \leq p$, each P_i starting in a' and ending in a'_i . This can be done using a Menger algorithm. Clearly, this algorithm decides if there is an F -subdivision in D . There are $O(n^{|V(F)|})$ possible pairs (S, a') , and for each of them we run at most $(|V(F)| - 1)!$ times a Menger algorithm. Since such an algorithm runs in linear time, the time complexity of the above algorithm is $O(n^{|V(F)|(n+m)})$. \square

The complexity given in Proposition 21 is certainly not optimal. For example, it can be improved for spindles with paths of small lengths.

Proposition 22. *If F is a (k_1, \dots, k_p) -spindle and $k_i \leq 2$ for all $1 \leq i \leq p$, then F -SUBDIVISION can be solved in $O(n^2(n+m))$ time.*

Proof. If some of the k_i , say k_1 , equals 1, then finding an F -subdivision is equivalent to find p internally disjoint directed paths from some vertex a to some other vertex b , which by Menger’s theorem is equivalent to check that the connectivity from a and b is at least p . For any pair (a, b) , this can be done in linear time by a Menger algorithm.

If $k_i = 2$ for all $1 \leq i \leq p$, then finding an F -subdivision is equivalent to find p internally disjoint directed paths of length at least two from some vertex a to some other vertex b . Such paths exist if and only if in $D \setminus ab$ there are p internally disjoint (a, b) -paths. For any pair (a, b) , this can be checked in linear time by a Menger algorithm. \square

1 A natural question is to ask about the complexity of deciding if a digraph contains a
 2 subdivision of a spindle, when the spindle is no more fixed but specified in the input.

3 **Proposition 23.** *The following problem is NP-complete*

4 SPINDLE-SUBDIVISION

5 Input: A spindle F and a digraph D .

6 Question: Does D contain a subdivision of F ?

7 *Proof.* Reduction from the (undirected) Hamiltonian cycle problem.

8 Let G be an undirected graph. Let $D(G)$ be the symmetric digraph associated to G , that is
 9 D is the digraph obtained from G by replacing every edge uv by the two arcs uv and vu . Let
 10 F be any spindle of the same order as G (and $D(G)$). For order reason, the digraph contains
 11 an F -subdivision if and only if it contains F as a subgraph, and thus if and only if G has a
 12 Hamiltonian cycle. \square

13 In view of Proposition 23, one could ask whether it is possible to solve SPINDLE-
 14 SUBDIVISION in $f(|V(F)|) \times n^c$ time, where f is a computable function and c an absolute
 15 constant? This may be formulated in FPT setting as follows.

16 **Problem 24.** Is the following problem fixed-parameter tractable?

17 PARAMETERIZED SPINDLE-SUBDIVISION

18 Input: A spindle F and a digraph D .

19 Parameter: $|V(F)|$.

20 Question: Does D contain a subdivision of F ?

21 There are many other digraphs that can be solved in the same way as spindles using a
 22 Menger algorithm. It is in particular the case of any oriented tree T such that there is a
 23 vertex r of in-degree 0 such that $T - r$ is the disjoint union of spiders. For such a tree, T -
 24 SUBDIVISION can be solved in $O\left(n^{|V(T)|-1}(n+m)\right)$ time. The polynomial-time solvability
 25 of F -SUBDIVISION of some other digraphs may also be established by using a Menger al-
 26 gorithm in a slightly different way as we show in the next two subsections.

27 5.2. Subdivision of windmills

28 A *cycle windmill* is a digraph obtained from disjoint directed cycles by taking one vertex
 29 per cycle and identifying all of these. This vertex will be called the *axis* of the windmill.

30 **Theorem 25.** *If W is a cycle windmill, then W -SUBDIVISION can be solved in $O(n^{|V(W)|}(n+m)$
 31 $m))$ time.*

32 *Proof.* Suppose W is a windmill with axis o and cycle lengths a_1, a_2, \dots, a_p . To check
 33 whether a given digraph $D = (V, A)$ contains a subdivision of W with axis at the vertex
 34 x we do the following (until success or all subsets have been tried): for all choices of dis-
 35 joint ordered subsets X_1, X_2, \dots, X_p of V such that $X_i = \{v_{i,1}, \dots, v_{i,a_i-1}\}$, $i = 1, 2, \dots, p$ check
 36 whether $Q_i = xv_{i,1}v_{i,2} \dots v_{i,a_i-1}$ is a directed (x, v_{i,a_i-1}) -path. If this holds for all i , then delete
 37 all the vertices of $X_i - v_{i,a_i-1}$, $i = 1, 2, \dots, p$ and check whether the resulting digraph contains
 38 internally disjoint paths P_1, P_2, \dots, P_p where P_i is a path from v_{i,a_i-1} to x using a Menger al-
 39 gorithm. If these paths exist, then return the desired subdivision of W formed by the union of
 40 $Q_1, Q_2, \dots, Q_p, P_1, P_2, \dots, P_p$. Otherwise continue to the next choice for X_1, X_2, \dots, X_p . Since

1 the size of $X_1 \cup X_2 \cup \dots \cup X_p$ is $|V(W)| - 1$, there are $O(n^{|V(W)|-1})$ choices for it, and there
 2 are n choices for x , hence the algorithm runs $O(n^{|V(W)|})$ times a Menger algorithm. Since a
 3 Menger algorithm runs in linear time, the overall complexity is $O(n^{|V(W)|(n+m)})$. \square

4 Clearly, given as input a windmill W and a digraph D , deciding if D contains a W -
 5 subdivision is NP-complete because the Hamiltonian directed cycle problem is a particular
 6 case of it. Theorem 25 tells us that this problem parameterized by $|W|$ is in XP. But is it
 7 fixed-parameter tractable?

8 **Problem 26.** Is the following problem fixed-parameter tractable?

9 CYCLE-WINDMILL SUBDIVISION

10 Input: A cycle windmill W and a digraph D .

11 Parameter: $|V(W)|$.

12 Question: Does D contain a subdivision of W ?

13 5.3. Subdivision of palm trees

14 A *palm tree* is a dumbbell, whose left and right plates are spiders, and whose bar is a
 15 directed path of length one. Observe that in a palm tree, the two clips must be the bodies of
 16 the spiders. A *palm grove* is a disjoint union of palm trees. For example, the two graphs A
 17 and B depicted Figure 1 are palm groves.

18 By Theorem 12(c), if F is a palm grove having two palm trees whose left spiders are
 19 not isomorphic and whose right spiders are not isomorphic, then F -SUBDIVISION is NP-
 20 complete. We shall now prove that it is indeed the only hard case. Observe that if a digraph
 21 contains a subdivision of a palm tree, then it contains a subdivision of this palm tree such
 22 that the only subdivided arc is the bar.

23 **Theorem 27.** *Let F be a palm grove. Then F -SUBDIVISION is polynomial-time solvable if
 24 and only if all its left spiders are isomorphic or all its right spiders are isomorphic.*

25 *Proof.* If there are two left spiders that are not isomorphic and there are two right spiders that
 26 are not isomorphic, then there exist two palm trees such that their left spiders are not isomor-
 27 phic and their right spiders are not isomorphic. Then, by Theorem 12(c), F -SUBDIVISION
 28 is NP-complete.

29 Assume now that all the right spiders are isomorphic to a spider R . Let L_1, \dots, L_p be the
 30 left spiders (possibly some of them are isomorphic). We shall describe an algorithm to solve
 31 F -SUBDIVISION.

32 Let D be a digraph. By the above remark, if D contains an F -subdivision, then it contains
 33 an F -subdivision such that only the bars of the palm trees are subdivided. Hence we look for
 34 such a subdivision. Observe that such a subdivision is the disjoint union of copies of each of
 35 the L_i , $1 \leq i \leq p$ and p copies of R together with p disjoint directed paths from the bodies
 36 of the copies of the L_i to the bodies of the p copies of R . Hence to decide if D contains
 37 an F -subdivision, we try all possibilities for the disjoint union of spiders L_i , $1 \leq i \leq p$, and
 38 p spiders R and for each possibility we check via a Menger algorithm if there are disjoint
 39 directed paths from the bodies of the L_i to the bodies of the copies of R .

40 Formally, the algorithm is the following. For each set of distinct vertices $\{u_1, \dots, u_p, v_1, \dots,$
 41 $v_p\}$ of D and family of disjoint subsets $\{U_1, \dots, U_p, V_1, \dots, V_p\}$ of D such that for $1 \leq i \leq p$,
 42 $u_i \in U_i$ and $v_i \in V_i$, we check if for all i , $D \langle U_i \rangle$ (resp. V_i) contains a spider isomorphic

1 to L_i (resp. R) with body u_i (resp. v_i). If not we proceed to the next case. If yes, we
2 check if there are p disjoint directed paths from $\{u_1, \dots, u_p\}$ to $\{v_1, \dots, v_p\}$ in the digraph
3 $D \setminus (\bigcup_{i=1}^p (U_i \cup V_i) \setminus \{u_i, v_i\})$ via a Menger algorithm. If there are such paths, the union of
4 them with the spiders is an F -subdivision and we return it. If such paths do not exist, we
5 proceed to the next case.

6 The number of possible cases is $O(n^{|V(F)|})$ and each run of the Menger algorithm can be
7 done in linear time. Hence the complexity of the algorithm is $O(n^{|V(F)|(n+m)})$. \square

8 6. The Fork Problem and bispindles

9 A *fork* with *bottom vertex* a , *top vertices* b and c and *centre* t is a digraph in which

- 10 • a, b and c are distinct, and t is distinct from b and c (but possibly equal to a),
- 11 • every vertex except a has in-degree 1 and a has in-degree 0, and
- 12 • all vertices except b, c and t have out-degree 1 and b and c have out-degree 0 and t has
13 out-degree 2.

14 The following problem is very useful, as it can be efficiently solved.

15 **FORK**

16 **Input:** A digraph D and three distinct vertices a, b and c .

17 **Question:** Does D contain a fork with bottom vertex a and top vertices b and c ?

18

19 **Lemma 28.** *FORK can be solved in linear time.*

20 *Proof.* Assume that a digraph D contains a fork with bottom vertex a and top vertices b and
21 c . Then, clearly, there are a directed (a, b) -path in $D - c$ and a directed (a, c) -path in $D - b$.

22 We claim that this necessary condition is also sufficient. Indeed, assume that there is a
23 directed (a, b) -path P in $D - c$ and a directed (a, c) -path Q in $D - b$. Let t be the last vertex
24 on P which also belongs to Q . Such a vertex exists because a is in P and Q . Then the union
25 of P and $Q[t, c]$ is the desired fork.

26 Since one can decide in linear time if there is a directed (u, v) -path in a digraph, FORK
27 can be solved in linear time. \square

28 The $(k_1, \dots, k_p; l_1, \dots, l_q)$ -*bispindle*, denoted $B(k_1, \dots, k_p; l_1, \dots, l_q)$, is the digraph ob-
29 tained from the disjoint union of a (k_1, \dots, k_p) -spindle with tail a_1 and head b_1 and a (l_1, \dots, l_q) -
30 spindle with tail a_2 and head b_2 by identifying a_1 with b_2 into a vertex a , and a_2 with b_1 into
31 a vertex b . The vertices a and b are called, respectively, the *left node* and the *right node*
32 of the bispindle. The directed (a, b) -paths are called the *forward paths*, while the directed
33 (b, a) -paths are called the *backward paths*.

34 We say that $(P_1, \dots, P_p; Q_1, \dots, Q_q)$ is a $(k_1, \dots, k_p; l_1, \dots, l_q)$ -bispindle if, for each $1 \leq$
35 $i \leq p$, P_i is a directed (c, d) -path of length k_i , for each $1 \leq j \leq q$, Q_j is a directed (d, c) -path
36 of length l_j and the union of the P_i and Q_j is $B(k_1, \dots, k_p; l_1, \dots, l_q)$.

37 Let F be a bispindle with p forward paths and q backward paths. Consider the big paths
38 multidigraph $BP(F)$. By Remark 9, we get the following.

1 **Proposition 29.** *Let F be a bispindle with p forward paths and q backward paths. If $p \geq 1$,
2 $q \geq 1$, and $p + q \geq 4$, then F -SUBDIVISION is NP-complete.*

3 On the other hand, if F has no backward paths or exactly one backward path and one for-
4 ward path, then it is a spindle or a directed cycle, respectively. In both cases, F -SUBDIVISION
5 can be solved in polynomial time as shown in Subsections 5.1 and 4.1, respectively.

6 We now show using Lemma 28 that, in the remaining cases, that is when F is a bispin-
7 dle with two forward paths and one backward path, F -SUBDIVISION is polynomial-time
8 solvable.

9 **Theorem 30.** *If F is a bispindle with two forward paths and one backward path, then F -
10 SUBDIVISION can be solved in $O(n^{|F|+1}(n+m))$ time.*

11 *Proof.* Let a be the left node of F and let b and c be its two out-neighbours in F .

12 For every subset S of $|F|$ vertices, we check if $D \langle S \rangle$ contains a copy of $F \setminus \{ab, ac\}$ with
13 a', b', c' corresponding to a, b, c , respectively. Then we check in $D - (S \setminus \{a', b', c'\})$ if there
14 is a fork with bottom vertex a' and top vertices b' and c' .

15 Since there are $O(n^{|F|})$ possible set S and FORK can be solved in linear time by Lemma 28,
16 our algorithm runs in $O(n^{|F|+1}(n+m))$ time. \square

17 The complexity given in Theorem 30 is certainly not best possible. Similarly to Proposi-
18 tion 23, one shows that given a digraph D and a bispindle F (with two forward paths and one
19 backward path), deciding if D contains an F -subdivision is NP-complete. It is again natural
20 to ask if it is FPT when parameterized by $|F|$.

21 **Problem 31.** Is the following problem fixed-parameter tractable?

22 **PARAMETERIZED BISPINDLE-SUBDIVISION**

23 **Input:** A bispindle F with two forward paths and one backward path and a digraph D .

24 **Parameter:** $|V(F)|$.

25 **Question:** Does D contain a subdivision of F ?

26 In the next section, we give faster algorithms to solve $B(1, 2; 1)$ - , $B(1, 2; 2)$ - and $B(1, 3; 1)$ -
27 SUBDIVISION.

28 7. Polynomial-time solvable problems via handle decomposition

29 Let D be a strongly connected digraph. A *handle* h of D is a directed path $(s, v_1, \dots, v_\ell, t)$
30 from s to t (where s and t may be identical) such that:

- 31 • for all $1 \leq i \leq \ell$, $d^-(v_i) = d^+(v_i) = 1$, and
- 32 • the digraph $D \setminus h$ obtained from D by *suppressing* h , that is removing the arcs and the
33 internal vertices of h , is strongly connected.

34 The vertices s and t are the *endvertices* of h while the vertices v_i are its *internal vertices*.
35 The vertex s is the *origin* of h and t its *terminus*. The *length* of a handle is the number of its
36 arcs, here $\ell + 1$. A handle of length one is said to be *trivial*.

37 Given a strongly connected digraph D , a *handle decomposition* of D starting at $v \in V(D)$
38 is a triple $(v, (h_i)_{1 \leq i \leq p}, (D_i)_{0 \leq i \leq p})$, where $(D_i)_{0 \leq i \leq p}$ is a sequence of strongly connected
39 digraphs and $(h_i)_{1 \leq i \leq p}$ is a sequence of handles such that:

- 1 • $V(D_0) = \{v\}$,
- 2 • for $1 \leq i \leq p$, h_i is a handle of D_i and D_i is the (arc-disjoint) union of D_{i-1} and h_i , and
- 3 • $D = D_p$.

4 A handle decomposition is uniquely determined by v and either $(h_i)_{1 \leq i \leq p}$, or $(D_i)_{0 \leq i \leq p}$.
 5 The number of handles p in any handle decomposition of D is exactly $|A(D)| - |V(D)| + 1$.
 6 The value p is also called the *cyclomatic number* of D . Observe that $p = 0$ when D is a
 7 singleton and $p = 1$ when D is a directed cycle.

8 A handle decomposition $(v, (h_i)_{1 \leq i \leq p}, (D_i)_{0 \leq i \leq p})$ is *nice* if all handles except the first
 9 on h_1 have distinct endvertices. The following proposition is well-known (see [5] Theo-
 10 rem 5.13).

11 **Proposition 32.** *Every robust digraph admits a nice handle decomposition.*

12 7.1. Subdivision of the lollipop

13 The *lollipop* is the digraph L with vertex set $\{x, y, z\}$ and arc set $\{xy, yz, zy\}$.

14 **Proposition 33.** *L-SUBDIVISION can be solved in linear time.*

15 *Proof.* If D contains a strong component of cyclomatic number greater than 1, then it con-
 16 tains a lollipop. Indeed, the smallest directed cycle C in the component is induced and is not
 17 the whole strong component. Hence there must be a vertex v dominating a vertex of C thus
 18 forming a lollipop-subdivision.

19 If not, then all the strong components are cycles. Thus D contains a lollipop if and only
 20 if one of its component is a directed cycle and is not an initial strong component (i.e some
 21 arc is entering it).

22 All this can be checked in linear time. □

23 7.2. Faster algorithm for subdivision of bispindles

24 In this subsection, using handle decomposition, we show algorithms to solve $B(1, 2; 1)$ - ,
 25 $B(1, 2; 2)$ - and $B(1, 3; 1)$ -SUBDIVISION, whose running time is smaller than the complexity
 26 of Theorem 30.

27 Recall that a digraph D is *robust* if it is strongly connected and $UG(D)$ is 2-connected.
 28 The *robust components* of a digraph are its robust subdigraphs which are maximal by inclu-
 29 sion.

30 Because bispindles are robust, a subdivision S of a bispindle is also robust, and if a
 31 digraph D contains S , then S must be in a robust component of D . Finding the robust com-
 32 ponents of a digraph can be done in linear time, by finding the strong components and the
 33 2-connected components of the underlying graphs of these. Therefore one can restrict our
 34 attention to subdivision of bispindles in robust digraphs.

1 7.2.1. Subdivision of the $(1,2;1)$ -bispindle

2 Observe that a subdivision of the $(1,2;1)$ -bispindle has cyclomatic number two. Con-
 3 versely, one can easily check that every robust digraph of cyclomatic number 2 is a subdivi-
 4 sion of the $(1,2;1)$ -bispindle. Hence, we have the following.

5 **Proposition 34.** *A digraph contains a subdivision of the $(1,2;1)$ -bispindle if and only if one*
 6 *of its robust components has cyclomatic number at least two.*

7 **Corollary 35.** *$B(1,2;1)$ -SUBDIVISION can be solved in linear time.*

8 *Proof.* Finding the robust components can be done in linear time and computing the cyclo-
 9 matic number of all of them in linear time as well. □

10 7.2.2. Subdivision of the $(1,2;2)$ -bispindle

11 In this subsection, we show that $B(1,2;2)$ -SUBDIVISION is polynomial-time solvable. In
 12 order to prove it, we characterize the robust digraphs that contain no $B(1,2;2)$ -subdivision.
 13 Let us now describe the family $\mathcal{F}_{1,2;2}$. A *double ring* is a digraph obtained from an undirected
 14 cycle by replacing every edge by two arcs, one in each direction. See Figure 2. A digraph
 15 G is in $\mathcal{F}_{1,2;2}$ if it is a double ring or it can be obtained from a (k_1, \dots, k_p) -spindle S , $p \geq 1$,
 16 with tail x and head y as follows. Add the arc yx and possibly some *back arcs*, that are, arcs
 17 vu such that $uv \in A(S)$, so that the unique directed (y,x) -path is the arc yx . See Figure 3.

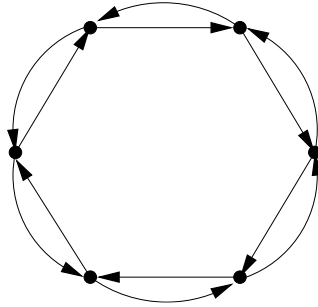


Figure 2: The double ring of order 6.

18 **Theorem 36.** *A robust digraph D contains a $B(1,2;2)$ -subdivision if and only if $D \notin \mathcal{F}_{1,2;2}$.*

19 *Proof.* Let us first prove that if $D \in \mathcal{F}_{1,2;2}$, then it contains no $B(1,2;2)$ -subdivision. Suppose
 20 for a contradiction, that there is such a subdivision S . Let a and b be the left and right nodes
 21 of a subdivision of S . Then the connectivity between a and b is at least 2 in one direction. So,
 22 by construction, either $(a,b) = (x,y)$, or (a,b) is such that ab is a back arc. But, in both cases,
 23 the unique directed (b,a) -path is (b,a) which has length less than 2, this is a contradiction.

24 Suppose now that $D \notin \mathcal{F}_{1,2;2}$. Let us prove that it contains a $B(1,2;2)$ -subdivision. Let
 25 $(v, (h_i)_{1 \leq i \leq p}, (D_i)_{0 \leq i \leq p})$ be a nice handle decomposition of D , and let i be the smallest
 26 positive integer such that $D_i \notin \mathcal{F}_{1,2;2}$. Clearly $i \geq 2$ because every directed cycle is in $\mathcal{F}_{1,2;2}$.
 27 Then D_{i-1} is in $\mathcal{F}_{1,2;2}$.

28 We shall prove that D_i contains a $B(1,2;2)$ -subdivision, and thus so does D .

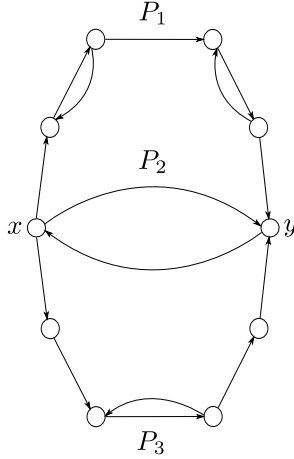


Figure 3: A digraph in $\mathcal{F}_{1,2;2}$, which is not a double ring

Suppose first that D_{i-1} is the double ring associated to a cycle $x_1x_2 \dots x_nx_1$. Without loss of generality, we may assume that the origin of h_i is x_1 and its terminus x_j for $2 \leq j \leq n$. Then $(h_i, x_1 \dots x_j; x_j \dots x_nx_1)$ is a $B(1, 2; 2)$ -subdivision. (Observe that if $j = 2$, then h_i must have length at least 2, since there are no multiple arcs.)

Suppose now that D_{i-1} is not a double ring. Let x and y be the two vertices of D_{i-1} as in the definition of $\mathcal{F}_{1,2;2}$. In other words, D_{i-1} is obtained from a spindle (P_1, P_2, \dots, P_k) with tail x and head y by adding yx and some back arcs. We distinguish several cases according to the possible locations of the tail u and head v of h_i . Observe that $(u, v) \neq (x, y)$ for otherwise D_i would be in $\mathcal{F}_{1,2;2}$.

- (i) $u = y$ and $v = x$. Since yx is an arc of D_{i-1} and there are no multiple arcs, the handle h_i has length at least 2. Hence $(yx, h_i; P_1)$ is a $B(1, 2; 2)$ -subdivision.
- (ii) $u = x$ and v is an internal vertex of some P_j . Since there are no multiple arcs, one of the two (x, v) -paths h_i and $P_j[x, v]$ has length at least 2. Hence $(h_i, P_j[x, v]; P_j[v, y]x)$ is a $B(1, 2; 2)$ -subdivision.
- (iii) $v = y$ and u is an internal vertex of some P_j . This case is similar to the previous one by directional symmetry.
- (iv) $u = y$ and v is an internal vertex of some P_j . Then $(h_i, yP_j[x, u]; P_j[u, y])$ is a $B(1, 2; 2)$ -subdivision. Note that, since $D_i \notin \mathcal{F}_{1,2;2}$, at least one of h_i and $P_j[u, y]$ has length more than one.
- (v) $v = x$ and u is an internal vertex of some P_j . This case is similar to the previous one by directional symmetry.
- (vi) u and v are internal vertices of the same P_j and u precedes v on P_j . Since there are no multiple arcs, one of the two (u, v) -paths h_i and uP_jv has length at least 2. Hence $(h_i, P_j[u, v]; P_j[v, y]xP_j[x, v])$ is a $B(1, 2; 2)$ -subdivision.
- (vii) u and v are internal vertices of the same P_j and v precedes u on P_j . If h_i is of length one, then in D_i all the back arcs associated to arcs of P_j exist, for otherwise D_i would be in $\mathcal{F}_{1,2;2}$. These arcs induce a directed (y, x) -path R_j of length at least 2. Moreover, $k \geq 2$, for otherwise D_i would be in $\mathcal{F}_{1,2;2}$ with y as left node and x as right node. If $k = 2$ and the path of $\{P_1, P_2\} \setminus \{P_j\}$ was of length one, then D_i would be a double

ring. Hence, there is $j' \neq j$ such that $P_{j'}$ has length at least two, and we have the $B(1, 2; 2)$ -subdivision $(yx, R_j; P'_j)$
(viii) u is an internal vertex of P_j , v is an internal vertex of $P_{j'}$ and $j \neq j'$. Then $(P_j[u, y], h_i P_{j'}[v, y]; yP_j[x, u])$ is a $B(1, 2; 2)$ -subdivision.

□

Corollary 37. $B(1, 2; 2)$ -SUBDIVISION can be solved in linear time.

7.2.3. Subdivision of the $(1, 3; 1)$ -bispindle

Observe that there is a C_4 in a $(1, 3; 1)$ -bispindle. So, a digraph D that has no directed cycle of length greater than 3 contains no $B(1, 3; 1)$ -subdivision.

Let D be a robust digraph and $C = (v_1, \dots, v_\ell, v_1)$ a directed cycle in D . A handle decomposition $(v, (h_i)_{1 \leq i \leq p}, (D_i)_{0 \leq i \leq p})$ is said to be C -bad if

- (i) $D_1 = C$;
- (ii) for all $i \geq 2$, h_i has length 1 or 2, its endvertices are on C and the distance between the origin and the terminus of h_i around C is 2.
- (iii) If h_i is a $(v_k, v_k + 2)$ -path and h_j is a $(v_{k-1}, v_k + 1)$ -path (indices are taken modulo ℓ), then these two handles have length 1.
- (iv) If $\ell \geq 5$, there no k such that (v_{k-2}, v_k) , (v_{k-1}, v_{k+1}) and (v_k, v_{k+2}) are handles.

The notion of C -bad handle decomposition plays a crucial role for finding $B(1, 3; 1)$ -subdivision as shown by the next two lemmas.

Lemma 38. Let D be a digraph and C a directed cycle in D of length at least 4. Then one of the following holds:

- D contains a $B(1, 3; 1)$ -subdivision,
- C is not a longest circuit in D , or
- D has a C -bad handle decomposition.

Proof. Set $C = (v_1, \dots, v_\ell, v_1)$. Let $\mathcal{H} = (v, (h_i)_{1 \leq i \leq p}, (D_i)_{0 \leq i \leq p})$ be a nice handle decomposition of D such that $D_1 = C$.

If \mathcal{H} is not C -bad, then let k be the largest integer such that $\mathcal{H}_k = (v, (h_i)_{1 \leq i \leq k}, (D_i)_{0 \leq i \leq k})$ is a C -bad handle decomposition. One of the following occurs:

- (i) the origin s_{k+1} of h_{k+1} is the internal vertex of some h_i , $i \geq 2$. Since \mathcal{H}_k is C -bad, then necessarily $h_i = (s_i, s_{k+1}, t_i)$, and there is a directed path (s_i, v_i, t_i) of length 2 in C . Let t_{k+1} be the terminus of h_{k+1} . If t_{k+1} is on C , we set $h^* = h_{k+1}$ and $t^* = t_{k+1}$. If not, then t_{k+1} has an out-neighbour t^* on C and we let h^* be the concatenation of h_{k+1} and (t_{k+1}, t^*) . In both cases, h^* is a directed (s_{k+1}, t^*) -path with no internal vertices in C . If $t^* = v_i$, then $h^* \cup (C \setminus \{s_i v_i\}) \cup (s_i, s_{k+1})$ is a directed cycle longer than C . If $t^* = s_i$, then $(C \cup h^* \cup (s_i, s_{k+1})) - v_i$ is a $B(1, 3; 1)$ -subdivision with right node s_i and left node s_{k+1} . If $t^* = t_i$, then $C[t_i, s_i] \cup h^*$ is a directed cycle longer than C because in that case h^* has length at least 2. If $t^* \notin \{s_i, t_i, v_i\}$, then $C \cup h^* \cup (s_i, s_{k+1})$ is a $B(1, 3; 1)$ -subdivision with left node s_i and right node t^* .

- 1 (ii) the terminus of h_{k+1} is the internal vertex of some h_i , $i \geq 2$. We get the result in a
2 similar way to the preceding case.
- 3 (iii) h_{k+1} has length greater than 2 and its two endvertices are on C . Then the union of C
4 and h_{k+1} is a $B(1, 3; 1)$ -subdivision.
- 5 (iv) $h_{k+1} = (s, t)$ with s, t and $C[s, t]$ has length at least 3. Then $C \cup (s, t)$ is a $B(1, 3; 1)$ -
6 subdivision with right node s and left node t .
- 7 (v) h_{k+1} is one of the two handles h and h' , where h is a (v_{k-1}, v_{k+1}) -handle and h' is
8 a (v_k, v_{k+2}) for some k , and one of h and h' has length two. If h has length two, say
9 (v_{k-1}, x_1, v_{k+1}) , then the union of $(v_{k-1}, v_k) \cup h'$, $(v_{k-1}, x_1, v_{k+1}, v_{k+2})$ and $C[v_{k+2}, v_{k-1}]$
10 form a $B(1, 3; 1)$ -subdivision. If h' has length two, say $h' = (v_k, x_2, v_{k+2})$, then the
11 union of $h \cup (v_{k+1}, v_{k+2})$, $(v_{k-1}, v_k, x_2, v_{k+2})$ and $C[v_{k+2}, v_{k-1}]$ form a $B(1, 3; 1)$ -subdivi-
12 sion.
- 13 (vi) h_{k+1} is one of the three handles (v_{k-2}, v_k) , (v_{k-1}, v_{k+1}) , (v_k, v_{k+2}) for some k and $p \geq$
14 5. In this case, the union of $(v_{k-2}, v_{k-1}, v_{k+1}, v_{k+2})$, (v_{k-2}, v_k, v_{k+2}) and $C[v_{k+2}, v_{k-2}]$
15 form a $B(1, 3; 1)$ -subdivision.

16 □

17 **Lemma 39.** *Let D be a robust digraph and C a directed cycle in D of length at least 4. If D
18 has a C -bad handle decomposition, then it does not contain any $B(1, 3; 1)$ -subdivision.*

19 *Proof.* By induction on the number p of handles of the handle decomposition, the result
20 holding trivially if $p = 1$.

21 Set $C = (v_1, \dots, v_\ell, v_1)$ and let $\mathcal{H} = (v, (h_i)_{1 \leq i \leq p}, (D_i)_{0 \leq i \leq p})$ be a C -bad handle decom-
22 position of D .

23 By the induction hypothesis D_{p-1} does not have any $B(1, 3; 1)$ -subdivision.

24 Suppose, by way of contradiction, that D_p contains a $B(1, 3; 1)$ -subdivision S . Necessar-
25 ily, h_p is a subdigraph of S . Free to rename, we may assume that v_1 and v_3 are the origin
26 and the terminus, respectively, of h_p . If v_2 is not in S , then replacing h_p with (v_1, v_2, v_3) in S ,
27 we obtain a $B(1, 3; 1)$ -subdivision contained in D_{p-1} , a contradiction. Hence $v_2 \in V(S)$. By
28 the conditions (iii) and (iv) of a C -bad handle decomposition, there cannot be both a handle
29 ending at v_2 and a handle starting at v_2 . By directional symmetry, we may assume that v_2
30 has in-degree one, and so $v_1 v_2 \in A(S)$, and v_1 is the left node of S . Now, $v_2 v_3$ is not an arc of
31 S , for otherwise v_3 will be the right node of S , and the two directed (v_1, v_3) -paths in S have
32 length at most 2, a contradiction. But, in S , there is an arc leaving v_2 , it must be in a handle,
33 and so by (iv) and (ii) of the definition of C -bad, this arc must be $v_2 v_4$. Again by (iii) of the
34 definition of C -bad, there is no arc leaving v_3 except $v_3 v_4$. Hence $v_3 v_4 \in A(S)$. Then v_4 is the
35 right node of S , and the two directed (v_1, v_4) -paths in S have length 2, a contradiction. □

36 **Theorem 40.** $B(1, 3; 1)$ -SUBDIVISION can be solved in $O(n \cdot m)$ time.

37 *Proof.* Given a digraph D , we compute the robust components of D and solve the problem
38 separately on each of them.

39 For each robust component, we first search for a directed cycle C_0 of length at least 4.
40 This can be done in $O(n \cdot m)$ time by Theorem 19. If there is no such cycle, then we return
41 ‘no’. If not, then we build a handle decomposition starting from $C := C_0$. Each time, we add a
42 new handle, one can mimick the proof of Lemma 38, we either find a $B(1, 3; 1)$ -subdivision
43 which we return, or a C -bad handle decomposition, or a directed cycle C' longer than the

1 current C . Observe that in this case, it is easy to derive a C' -bad handle decomposition
 2 containing the vertices added so far from the C -bad one. This can be done in $O(n \cdot m)$ time
 3 because an arc has to be considered only when it is added in a handle, and we just need to
 4 keep a set of at most m handles.

5 At the end of this process, if no $B(1, 3; 1)$ -subdivision has been returned, we end up with
 6 a C -bad decomposition of D . So, by Lemma 39, D has no $B(1, 3; 1)$ -subdivision, and we can
 7 proceed to the next robust component, or return ‘no’ if there is none. \square

8. Classes of digraphs for which F -SUBDIVISION is polynomial-time solvable for all F

9 **Lemma 41.** *Let \mathcal{D} be a class of digraphs which is closed under the operation which takes
 10 as input a digraph $D \in \mathcal{D}$, a bounded set of vertices $x_1, x_2, \dots, x_r \in V(D)$ and integers
 11 $i_1, i_2, \dots, i_r, o_1, o_2, \dots, o_r$, all between 0 and r and outputs the digraph D' that is obtained
 12 as follows: For $j = 1, 2, \dots, r$ replace x_j and all arcs incident to it by two sets of ver-
 13 tices $I_j = \{v_{j,1}, \dots, v_{j,i_j}\}, O_j = \{w_{j,1}, \dots, w_{j,o_j}\}$ (if $i_j = 0$ or $o_j = 0$ the corresponding set
 14 is empty), all possible arcs from $N_D^-(x_j)$ to I_j and from O_j to $N_D^+(x_j)$. If k -LINKAGE is
 15 polynomial-time solvable for all fixed k for digraphs in \mathcal{D} , then, for each digraph F , F -
 16 SUBDIVISION is polynomial-time solvable on digraphs in \mathcal{D} .*

17 *Proof.* Let F be a digraph with vertex set $\{1, 2, \dots, r\}$ and let D belong to \mathcal{D} . It is sufficient
 18 to show that we can decide in polynomial time whether a fixed one-to-one mapping of $V(F)$
 19 to $V(D)$ extends to a subdivision of F in D . So we assume below that a one-to-one mapping
 20 of $V(F)$ to $V(D)$ is given.

21 For each vertex $\alpha \in V(F)$, fix an ordering of the arcs entering α and an ordering of
 22 the arcs leaving α : We label the $d_F^-(\alpha)$ in-neighbours of α by $i_{\alpha,1}, i_{\alpha,2}, \dots, i_{\alpha,d_F^-(\alpha)}$ and we
 23 label the $d_F^+(\alpha)$ out-neighbours of α by $o_{\alpha,1}, o_{\alpha,2}, \dots, o_{\alpha,d_F^+(\alpha)}$. For a given arc $e = \alpha\beta \in$
 24 $A(F)$ this gives two labels $l_{\alpha\beta}^+$ and $l_{\alpha\beta}^-$ (the number it has in α 's out-labelling and in β 's
 25 in-labelling). Given the one-to-one mapping $f : V(F) \rightarrow V(D)$ we make a new digraph D_F
 26 from D by replacing each vertex $f(\alpha)$, $\alpha \in V(F)$ by two sets $I_{f(\alpha)} = \{i_{\alpha,1}, i_{\alpha,2}, \dots, i_{\alpha,d_F^-(\alpha)}\}$
 27 and $O_{f(\alpha)} = \{o_{\alpha,1}, o_{\alpha,2}, \dots, o_{\alpha,d_F^+(\alpha)}\}$ and joining every in-neighbour x of $f(\alpha)$ in D to every
 28 vertex y in $I_{f(\alpha)}$ by an arc $x \rightarrow y$ and every vertex p of $O_{f(\alpha)}$ to every out-neighbour q of
 29 $f(\alpha)$ in D (it is possible that one of the sets $I_{f(\alpha)}, O_{f(\alpha)}$ is empty in which case we add no
 30 arcs corresponding to that set).

31 Now it is easy to check that f can be extended to a subdivision of F in D if and only if
 32 D_F contains vertex-disjoint paths $\{P_{\alpha\beta} \mid \alpha\beta \in A(F)\}$ where $P_{\alpha\beta}$ starts in $o_{\alpha,l_{\alpha\beta}^+}$ and ends in
 33 $i_{\beta,l_{\alpha\beta}^-}$. Since D_F is in \mathcal{D} we can check the existence of the desired paths in polynomial time.
 34 Doing this for (at most) all possible one-to-one mappings of $V(F)$ to $V(D)$ we can decide in
 35 polynomial time (since $|V(F)|$ is constant) whether D contains an F -subdivision. \square

36 **Theorem 42** (Fortune, Hopcroft and Wyllie [10]). *For every fixed k the k -LINKAGE problem
 37 is polynomial-time solvable for acyclic digraphs.*

38 Clearly the class of acyclic digraphs is closed under the operation given in Lemma 41
 39 and hence we have the following.

40 **Corollary 43** (Fortune, Hopcroft and Wyllie [10]). *For every digraph F , F -SUBDIVISION
 41 is polynomial-time solvable for acyclic digraphs.*

1 The algorithm given by Fortune, Hopcroft and Wyllie to solve k -LINKAGE problem has a
 2 running time in $O(k!n^{k+2})$. Hence a natural question is to ask if it can be solved in $O(f(k)n^c)$
 3 time for some absolute constant c and computable function f . In the FPT setting, it can be
 4 phrased as follows.

5 **Problem 44.** Is the following parameterized problem FPT?

6 **PARAMETERIZED ACYCLIC k -LINKAGE**

7 **Input:** An acyclic digraph D and $2k$ distinct vertices $x_1, x_2, \dots, x_k, y_1, y_2, \dots, y_k$.

8 **Parameter:** k .

9 **Question:** Is there a k -linkage from (x_1, x_2, \dots, x_k) to (y_1, y_2, \dots, y_k) in D ?

10 **Theorem 45** (Johnson et al. [14]). *For every fixed k , k -LINKAGE is polynomial-time solvable*
 11 *on digraphs of bounded directed tree-width.*

12 We will not give the definition of directed tree-width here as it is rather technical, but it
 13 suffices to say that the class of digraphs with bounded directed tree-width is closed on the
 14 operation of Lemma 41 so we have.

15 **Theorem 46** (Johnson et al. [14]). *For every digraph F , F -SUBDIVISION is polynomial-time*
 16 *solvable on digraphs of bounded directed tree-width.*

17 **Theorem 47** (Chudnovsky et al. [6]). *For any digraph F , F -SUBDIVISION is polynomial-*
 18 *time solvable when restricted to the class of tournaments.*

19 Let $D = (V, A)$ be a digraph. We say that $W \subseteq V$ guards $V' \subseteq V$ in D if $N^+(V') \subseteq W$,
 20 that is, all out-neighbours of V' are in W . A *DAG-decomposition* of a digraph D is a pair
 21 (H, χ) where H is an acyclic digraph and $\chi = \{W_h : h \in V(H)\}$ is a family of subsets of
 22 $V(D)$ satisfying the following three properties:

23 (i) $V(D) = \bigcup_{h \in V(H)} W_h$,

24 (ii) for all $h, h', h'' \in V(H)$, if h' lies on a directed (h, h'') -path, then $W_h \cap W_{h''} \subseteq W_{h'}$, and

25 (iii) if $(h, h') \in A(H)$, then $W_h \cap W_{h'}$ guards $W_{\geq h'} \setminus W_h$, where $W_{\geq h'}$ is the union of all $W_{h''}$
 26 for which there exists a directed (h', h'') -path in H .

27 The *width* of a DAG-decomposition (H, χ) is $\max_{h \in V(H)} |W_h|$. The *DAG-width* of a digraph
 28 D ($\text{dagw}(D)$) is the minimum width over all possible DAG-decompositions of D . It is easy
 29 to see that a digraph D is acyclic if and only if it has DAG-width 1 (and then we can use D
 30 itself as H).

31 **Theorem 48** (Berwanger et al. [4], Johnson et al. [14]). *For every fixed k , k -LINKAGE is*
 32 *polynomial-time solvable on digraphs of bounded DAG-width.*

33 Digraphs of bounded DAG-width are closed under the operation in Lemma 41 so we
 34 have.

35 **Corollary 49.** *For any digraph F , F -SUBDIVISION is polynomial-time solvable when re-*
 36 *stricted to the class of digraphs of bounded DAG-width.*

1 More generally, the property of having an F -subdivision can be defined in MSO1 monadic
 2 second order logic with vertex-set quantifications) and so can be solved in polynomial time
 3 on the class of digraphs with bounded directed clique-width. If F is not fixed, but specified
 4 in the input, it can also be solved in FPT-time when parameterized by $V(F)$. See Theorem
 5 1.24 of [9].

6 A *feedback vertex set* or *cycle transversal* in a digraph D is a set of vertices S such that
 7 $D - S$ is acyclic. The minimum number of vertices in a cycle transversal of D is the *cycle-*
 8 *transversal number* and is denoted by $\tau(D)$.

9 **Corollary 50.** *For any digraph F , F -SUBDIVISION is polynomial-time solvable when re-*
 10 *stricted to the class of digraphs with bounded cycle-transversal number.*

11 *Proof.* Let X be a cycle-transversal of D . Then $D' = D - X$ is acyclic and it is easy to
 12 see that D has DAG-width at most X , since we can take $H = D'$ and $W_h = \{h\} \cup X$ for all
 13 $h \in V(D')$ to obtain a DAG-decomposition of D whose width is $|X|$. Now the result follows
 14 from Corollary 49. \square

15 The maximum number of disjoint directed cycles in a digraph D is called the *cycle-*
 16 *packing number* and is denoted by $\nu(D)$. Clearly, $\nu(D) \leq \tau(D)$. Conversely, proving the
 17 so-called Gallai-Younger Conjecture, Reed et al. [17] proved that $\tau(D)$ is bounded above by
 18 a function of $\nu(D)$.

19 **Theorem 51** (Reed et al. [17]). *For every k , there is an integer $f(k)$ such that every digraph*
 20 *has either k disjoint directed cycles or a feedback vertex set of size at most $f(k)$.*

21 The function f constructed by Reed et al. [17] grows very quickly. It is a multiply
 22 iterated exponential, where the number of iterations is also a multiply iterated exponential.
 23 The correct value of $f(2)$ is 3 as shown by McCuaig [16] who also gave a polynomial-time
 24 algorithm for finding two disjoint directed cycles in a digraph or showing that it has $\tau(D) \leq 3$.

25 **Corollary 52.** *For any digraph F , F -SUBDIVISION is polynomial-time solvable when re-*
 26 *stricted to the class of digraphs with bounded cycle-packing number.*

27 9. F -SUBDIVISION for some special classes of digraphs

28 In this section the focus is on the structure of F rather than the method for solving F -
 29 SUBDIVISION or proving it NP-complete. For several of the classes we can provide (almost)
 30 complete characterizations in terms of complexity of F -SUBDIVISION .

31 9.1. Disjoint union of directed cycles

32 Since C_k -SUBDIVISION can be solved in polynomial time for any fixed k , a natural ques-
 33 tion is to ask for the complexity of F -SUBDIVISION when F is the disjoint union of directed
 34 cycles. This is not a simple problem as can be seen from the observation that a digraph D
 35 contains k disjoint directed cycles if and only if it contains an F -subdivision where F is the
 36 disjoint union of k directed 2-cycles.

37 Hence, if F is the disjoint union of k directed 2-cycles, F -SUBDIVISION is equivalent to
 38 deciding if $\nu(D) \geq k$ for a given digraph D . Using Theorem 51, Reed et al. [17] proved that
 39 this can be done in polynomial time.

1 **Theorem 53** (Reed et al. [17]). *For any fixed k , deciding if a digraph D has k disjoint*
 2 *directed cycles is polynomial-time solvable. Equivalently, if F is the disjoint union of directed*
 3 *2-cycles, then F -SUBDIVISION is polynomial-time solvable.*

4 **Remark 54.** Determining $v(D)$ is NP-hard. Indeed, given a digraph D and an integer k ,
 5 deciding whether D has at least k disjoint cycles is NP-complete. See Theorem 13.3.2 and
 6 Exercise 13.25 of [1]. As observed in [13], the problem parameterized with k is hard for the
 7 complexity class $W[1]$ (this follows easily from the results of [19]). This means that, unless
 8 $FPT = W[1]$, there is no algorithm solving the problem with a $f(k) \cdot n^{O(1)}$ running time.

9 **Problem 55.** Let F be the disjoint union of p directed cycles of lengths k_1, k_2, \dots, k_p , respec-
 10 tively. Is F -SUBDIVISION polynomial-time solvable?

11 **Theorem 56.** $(C_2 + C_3)$ -SUBDIVISION is polynomial-time solvable.

12 *Proof.* Let D be a digraph. If D has no 2-cycles, then D has a $C_2 + C_3$ -subdivision if and only
 13 if it contains two disjoint cycles. This can be checked in polynomial time by Theorem 51.

14 Assume now that D contains 2-cycles. For each 2-cycle (x, y, x) , we check if $D - \{x, y\}$
 15 has a directed cycle of length at least 3. This can be done in linear time according to Theo-
 16 rem 18. If the answer is ‘yes’ for one of them, then we return ‘yes’.

17 Suppose now that the answer is ‘no’ for all 2-cycles. Let D' be the digraph obtained from
 18 D by deleting the arcs of all the 2-cycles.

19 **Claim 56.1.** D contains a $(C_2 + C_3)$ -subdivision if and only if D' contains two disjoint di-
 20 rected cycles.

21 *Subproof.* Suppose that D contains a $(C_2 + C_3)$ -subdivision S . No cycle of S can contain two
 22 vertices x and y in a 2-cycle because $D - \{x, y\}$ contains no directed cycle of length at least
 23 3. In particular, all the arcs of S are in D' .

24 Conversely, if D' contains two disjoint directed cycles, they form a $(C_2 + C_3)$ -subdivision
 25 since D' has no 2-cycles. \diamond

26 Hence we check if D' has two disjoint directed cycles, which can be done in polynomial
 27 time according to Theorem 51. \square

28 9.2. Subdivisions of wheels and fans

29 The *fan* F_k is the graph obtained from the directed path P_k by adding a vertex, called
 30 the *centre*, dominated by every vertex of P_k . The *wheel* W_k is the graph obtained from the
 31 directed cycle C_k by adding a vertex, called the *centre*, dominated by every vertex of C_k . The
 32 path P_k (resp. cycle C_k) is called the *rim* of F_k (resp. W_k) and the arcs incident to the centre
 33 are called the *spokes*. Similarly, if D' is a subdivision of a wheel or a fan D , the *centre* of
 34 D' is the vertex corresponding to the centre of D , the *rim* of D' is the directed path or cycle
 35 corresponding to the rim of D , and the *spokes* of D' are the directed paths corresponding to
 36 the spokes of D .

37 **Proposition 57.** *A digraph D contains a W_2 -subdivision if and only if it contains some vertex*
 38 *z such that $D - z$ has a strong component S and two directed (S, z) -paths having only z in*
 39 *common.*

1 *Proof.* Suppose D contains a subdivision of W_2 with centre z and cycle C . Then the strong
 2 component of $D - z$ which contains C satisfies the required property.

3 Conversely, assume z is a vertex and S is a strong component of $D - z$ such that there are
 4 two directed (S, z) -paths P and Q having only z in common. Let x and y be the origins of P
 5 and Q respectively.

6 Let R be a directed (x, y) -path in S and R' a directed (y, x) -path in S . (Such paths exist
 7 since S is a strong component.) If R and R' form a cycle we are done, with this cycle as rim
 8 and P, Q as spokes. Otherwise let q be the last vertex in $R' \setminus \{x, y\}$ which is also on R . Then
 9 we have a W_2 -subdivision with rim $R[x, q]R'[q, x]$ and spokes P and $R[q, y]Q$. \square

10 **Corollary 58.** W_2 -SUBDIVISION is solvable in $O(n \cdot (n + m))$ time.

11 **Theorem 59.** For all $k \geq 4$, W_k -SUBDIVISION is NP-complete.

12 *Proof.* We give the proof for $k = 4$ (the case for larger k is very similar). We show a reduction
 13 from 2-LINKAGE in digraphs with no big vertices in which x_1 and x_2 are sources and y_1 and
 14 y_2 are sinks.

15 Let D, x_1, x_2, y_1, y_2 be an instance of this problem. Let D' be the graph obtained by adding
 16 five new vertices z, a, b, c, d and the arcs $az, bz, cz, dz, ab, cd, y_2a, bx_1, y_1c$, and dx_2 .

17 Let us prove that D' has a W_4 -subdivision if and only if D has a 2-linkage from (x_1, x_2)
 18 to (y_1, y_2) .

19 If P_1, P_2 form the desired 2-linkage in D , then we take $P_1y_1cdP_2abx_1$ as the rim and the
 20 four arcs az, bz, cz, dz as the spokes.

21 Conversely, suppose W is a subdivision of W_4 in D' and let C be its rim. The centre of W
 22 must be z as this is the only vertex of in-degree 4 in D' . Thus the four paths ending in z will
 23 end in the arcs az, bz, cz, dz , respectively. Now observe that a (and similarly c) must belong
 24 to C since otherwise the path containing az cannot be disjoint from the path containing bz
 25 (they will meet in a). Thus a is on C and then b is on C since it is the only out-neighbour of
 26 a different from z . Similarly d is on C . Hence C contains the arcs ab and cd and this implies
 27 that C contains disjoint paths from x_1 to y_1 and x_2 to y_2 respectively. \square

28 **Remark 60.** It is not difficult to modify the proof above to a proof that F -SUBDIVISION is
 29 NP-complete whenever F is any digraph obtained from a W_k with $k \geq 4$ by reorienting one
 30 or more of the spokes. E.g. if the arc dz is reversed, then we replace the arcs ab and cd by
 31 arcs ax_1, y_1b, cx_2, y_2d . We leave the details to the interested reader.

32 From this remark and Lemmas 2, 3 and 4 we get the following corollary. Notice that the
 33 resulting digraphs may still have only one big vertex so the conclusion does not follow from
 34 Theorem 8.

35 **Corollary 61.** Let $W'_k, k \geq 4$ be a strongly connected digraph obtained from W_k by reversing
 36 between one and $k - 1$ spokes and let G be any digraph not containing a subdivision of W'_k .
 37 Then F -SUBDIVISION and F' -SUBDIVISION are NP-complete, where F is obtained from W'_k
 38 and G by adding zero or more arcs from $V(W'_k)$ to $V(G)$ and F' is obtained from W'_k and G
 39 by identifying the big vertex of W'_k with an arbitrary vertex of G .

40 Corollary 58 and Theorem 59 determine the complexity of W_k -SUBDIVISION for all k
 41 except 3. So we are left with the following problem.

1 **Problem 62.** What is the complexity of W_3 -SUBDIVISION ?

2 We now turn to fans. Notice that F_k is W_k where one arc of the rim is deleted. Observe
 3 that F_2 is the $(1, 2)$ -spindle. Thus F_2 -SUBDIVISION can be solved in $O(n^2(n + m))$ time by
 4 Proposition 22. The next result shows that F_3 -SUBDIVISION is polynomial.

5
 6 Let z be a vertex in a digraph D . A triple (x_1, x_2, x_3) is F_3 -nice with respect to z in D if
 7 the following holds:

- 8 • x_1, x_2, x_3 are distinct vertices of $D - z$;
- 9 • x_3z is an arc;
- 10 • in $D - x_3$, there exist a directed (x_1, z) -path P_1 and a directed (x_2, z) -path P_2 which
 11 intersect only in z ;
- 12 • in $D - \{x_3, z\}$, there is a directed (x_1, x_2) -path Q_1 , and in $D - \{x_1, z\}$, there is a directed
 13 (x_2, x_3) -path Q_2 .

14 **Theorem 63.** A digraph contains an F_3 -subdivision with centre z if and only if there is an
 15 F_3 -nice triple with respect to z . In particular F_3 -SUBDIVISION is polynomial-time solvable.

16 *Proof.* Trivially, if D contains an F_3 -subdivision with centre z , then it contains an F_3 -nice
 17 triple (x_1, x_2, x_3) with respect to z .

18 Conversely, assume that D contains an F_3 -nice triple (x_1, x_2, x_3) with respect to z . Let
 19 P_1, P_2, Q_1 and Q_2 be the directed paths as defined in the definition of F_3 -nice triple. We
 20 may assume that (x_1, x_2, x_3) is an F_3 -nice triple (x_1, x_2, x_3) with respect to z that minimizes
 21 $\ell = \ell(P_1) + \ell(P_2) + \ell(Q_1) + \ell(Q_2)$, that is the sum of the lengths of these paths.

22 We shall prove that P_1, P_2, Q_1 and Q_2 are internally disjoint, implying that these paths
 23 and the arc x_3z form an F_3 -subdivision with centre z .

24 a) Let us prove that Q_2 and P_1 are internally disjoint. Suppose not. Then let x'_2 be the
 25 last vertex on Q_2 which also belongs to P_1 . Then (x_2, x'_2, x_3) is F_3 -nice by the choice
 26 of paths $P'_1 = P_2, P'_2 = P_1[x'_2, z], Q'_1 = Q_2[x_2, x'_2]$ and $Q'_2 = Q_2[x'_2, x_3]$. Indeed, P'_1 and
 27 P'_2 are internally disjoint because P_1 and P_2 were, Q'_1 does not go through x_3 nor z ,
 28 because Q_2 is a directed (x_2, x_3) -path in $D - z$, and Q'_2 does not go through x_2 nor z , for
 29 the same reason. This contradicts the minimality of ℓ .

30 b) Let us prove that Q_2 and P_2 are internally disjoint. Suppose not. Then let x'_2 be the last
 31 vertex on Q_2 which also belongs to P_2 . One easily verifies that (x_1, x'_2, x_3) is F_3 -nice
 32 by the choice of paths $P'_1 = P_1, P'_2 = P_2[x'_2, z], Q'_1$ a directed (x_1, x'_2) -path included in
 33 $Q_1[x_1, x_2]Q_2[x_2, x'_2]$ (which can be a walk), and $Q'_2 = Q_2[x'_2, x_3]$. This contradicts the
 34 minimality of ℓ .

35 c) Let us prove that Q_1 and P_1 are internally disjoint. Suppose not. Then let x'_1 be the last
 36 vertex on Q_1 which also belongs to P_1 . The path Q_2 does not go through x'_1 because
 37 Q_2 and P_1 are internally disjoint. Thus (x'_1, x_2, x_3) is F_3 -nice with associated paths
 38 $P'_1 = P_1[x'_1, z], P'_2 = P_2, Q'_1 = Q_1[x'_1, x_2]$, and $Q'_2 = Q_2$. This contradicts the minimality
 39 of ℓ .

- 1 d) Let us prove that Q_1 and P_2 are internally disjoint. Suppose not. Then let x'_2 be the
2 last internal vertex on Q_1 which also belongs to P_2 . Then (x_1, x'_2, x_3) is F_3 -nice with
3 associated paths $P'_1 = P_1$, $P'_2 = P_2[x'_2, z]$, $Q'_1 = Q_1[x_1, x'_2]$, and Q'_2 a directed (x_1, x'_2) -path
4 included in $Q_1[x'_2, x_2]Q_2$ (which can be a walk). This contradicts the minimality of ℓ .
- 5 e) Let us prove that Q_1 and Q_2 are internally disjoint. Suppose not. Then let x'_2 be the
6 last internal vertex on Q_2 which also belongs to Q_1 . Then (x_1, x'_2, x_3) is a good triple
7 with associated paths $P'_1 = P_1$, $P'_2 = Q_1[x'_2, x_2]P_2$, $Q'_1 = Q_1[x_1, x'_2]$, and $Q'_2 = Q_2[x'_2, x_3]$.
8 Indeed, since P_2 and Q_1 are internally disjoint, P'_2 is a path, and since P_1 and Q_1 are
9 internally disjoint, the paths P'_1 and P'_2 are also internally disjoint.

10 \square

11 **Theorem 64.** *For all $k \geq 5$, F_k -SUBDIVISION is NP-complete.*

12 *Proof.* Reduction from 2-LINKAGE in digraphs with no big vertices in which x_1 and x_2 are
13 sources and y_1 and y_2 are sinks.

14 Let D , x_1 , x_2 , y_1 and y_2 be an instance of this problem. Let us denote by z the centre of
15 F_k and by (v_1, v_2, \dots, v_k) the directed path $F_k - z$. Let D_k be the digraph obtained from the
16 disjoint union of D and F_k by removing the arcs v_1v_2 and v_3v_4 and adding the arcs v_1x_1 , y_1v_2 ,
17 v_3x_2 and y_2v_4 .

18 We claim that D_k has an F_k -subdivision if and only if D has a linkage from (x_1, x_2) to
19 (y_1, y_2) .

20 Clearly, if there is a linkage (P_1, P_2) from (x_1, x_2) to (y_1, y_2) in D , then D_k contains an
21 F_k -subdivision, obtained from F_k by replacing the arc v_1v_2 and v_3v_4 by the directed paths
22 $(v_1, x_1) \cup P_1 \cup (y_1, v_2)$ and $(v_3, x_2) \cup P_2 \cup (y_2, v_4)$, respectively.

23 Suppose now that D_k contains an F_k -subdivision S in D_k . Since z is the unique vertex with
24 in-degree k , the centre of S' is necessarily z . For $1 \leq i \leq k$, let v'_i be the vertex corresponding
25 to v_i in S , and P_i be the directed (v'_i, z) -path in S .

26 Since z has in-degree exactly k in D_k , the v_i 's are the penultimate vertices of the P_j 's,
27 each v_i on a different P_j . Since v_1 is a source in D_k , then $v_1 = v'_1$. Moreover, for $i = 3$ and
28 $i \geq 5$, the path P'_i containing v_i must start at v_i because the unique in-neighbour of v_i is v_{i-1} .
29 Hence $v_i = v'_i$. Furthermore, necessarily $v_{i-1} = v'_{i-1}$. Now, because v_k is a sink in $D_k - z$,
30 then necessarily $v'_k = v_k$ and so for all $1 \leq i \leq k$, we have $v'_i = v_i$.

31 Let Q_1 and Q_2 be the directed (v_1, v_2) - and (v_3, v_4) -paths, respectively. Necessarily, the
32 second vertex of Q_1 (resp. Q_2) is x_1 , (resp. x_2) and its penultimate vertex is y_1 (resp. y_2).
33 Hence $(Q_1[x_1, y_1], Q_2[x_2, y_2])$ is a linkage from (x_1, x_2) to (y_1, y_2) in D . \square

34 Proposition 22 and Theorems 63 and 64 determine the complexity of F_k -SUBDIVISION
35 for all k except 4. So we are left with the following problem.

36 **Problem 65.** What is the complexity of F_4 -SUBDIVISION ?

37 9.3. Subdivisions of transitive tournaments

38 Denote by TT_k the transitive tournament on k vertices. For $k \leq 3$, TT_k -SUBDIVISION is
39 polynomial-time solvable because TT_1 and TT_2 are spiders and TT_3 is the $(1, 2)$ -spindle. On
40 the other hand, for all $k \geq 5$, TT_k -SUBDIVISION is NP-complete by Corollary 10. We shall
41 now prove that TT_4 -SUBDIVISION is polynomial-time solvable.

1 In fact we will prove it for some classes of graphs constructed from TT_4 . For any non-
2 negative integer p , let $TT_4(p)$ be the digraph obtained from TT_4 with source u and sink v by
3 adding p new vertices dominated by u and dominating v . In particular, $TT_4(0) = TT_4$. We
4 denote by $TT_4^*(p)$, the digraph obtained from $TT_4(p)$ by deleting the arc from its source u
5 to its sink v . For simplicity, we abbreviate $TT_4^*(0)$ in TT_4^* .

6 We need the following definitions. Let X be a set of vertices in a digraph D . The *out-*
7 *section* generated by X in D is the set of vertices y to which there exists a directed path
8 (possibly restricted to a single vertex) from $x \in X$; we denote this set by $S_D^+(X)$. For sim-
9 plicity, we write $S_D^+(x)$ instead of $S_D^+(\{x\})$. The dual notion, the *in-section*, is denoted by
10 $S_D^-(X)$. Note that the out-section and the in-section of a set may be found in linear time by
11 any tree-search algorithm.

12 **Theorem 66.** *For every non-negative integer p , one can solve $TT_4(p)$ -SUBDIVISION in*
13 *$O(n^3(n+m))$ -time.*

14 *Proof.* Let D be a digraph and let u and v be two distinct vertices of D . We shall describe a
15 $O(n(n+m))$ -time algorithm for finding a $TT_4(p)$ -subdivision in D with source u and sink v ,
16 if one exists.

17 Observe that all vertices in such a subdivision are in $S_D^+(u) \cap S_D^-(v)$, hence we first restrict
18 to the graph D' the digraph induced by this set.

19 Then, using a maximum flow algorithm, we can find in D' a set of internally disjoint
20 directed (u, v) -paths of maximum size in $O(n(n+m))$ -time. Let (P_1, \dots, P_k) denote this set.
21 If $k < p+3$, then return ‘no’, because in any $TT_4(p)$ -subdivision with source u and sink v ,
22 there are $p+3$ internally disjoint directed (u, v) -paths. Hence, we now assume that $k \geq 3$.

23 For $1 \leq i \leq k$, set $Q_i = P_i - \{u, v\}$, and set $H = D' - \{u, v\}$. For every vertex x in $V(H)$,
24 we compute $S(x) = S_H^-(x) \cup S_H^+(x)$, and deduce $I(x) = \{i \mid V(Q_i) \cap S(x) \neq \emptyset\}$. If there exists
25 x , such that $|I(x)| \geq 2$, then return ‘yes’. Otherwise return ‘no’.

26 The validity of this algorithm is proved by Claim 66.2.

27 **Claim 66.1.** *For all $x \in V(H)$, $I(x) \neq \emptyset$.*

28 *Subproof.* In D' , there are directed (u, x) - and (x, v) -paths, whose concatenation contains a
29 directed (u, v) -path R . Since (P_1, \dots, P_k) is a set of internally disjoint directed (u, v) -paths
30 of maximum size, $R - \{u, v\}$ must intersect one of the Q_i 's, say Q_{i_0} . By definition, $V(R) \setminus$
31 $\{u, v\} \subseteq S(x)$, so $i_0 \in I(x)$. \diamond

32 **Claim 66.2.** *D' contains a $TT_4(p)$ -subdivision with source u and sink v if and only if there*
33 *exists $x \in V(H)$ such that $|I(x)| \geq 2$.*

34 *Subproof.* Assume that $|I(x)| \geq 2$. Without loss of generality, $\{1, 2\} \subset I(x)$. We shall prove
35 that D' contains a $TT_4(p)$ -subdivision with source u and sink v .

- 36 • Suppose first that $S_H^-(x) \cap Q_1 \neq \emptyset$ and $S_H^+(x) \cap Q_2 \neq \emptyset$. Then there is a directed (Q_1, x) -
37 path and a directed (x, Q_2) - path whose concatenation contains a directed (Q_1, Q_2) -
38 path R . Let y be the first vertex on R in $\bigcup_{i=2}^k Q_i$. Free to swap the names of Q_2 and the
39 path Q_l containing y and taking the subpath of R from its origin to y instead of R , we
40 may assume that y is the last vertex of R . Now the union of P_1, \dots, P_{p+3} , and R form a
41 $TT_4(p)$ -subdivision.

- 1 • If $S_H^-(x) \cap Q_2 \neq \emptyset$ and $S_H^+(x) \cap Q_1 \neq \emptyset$, the proof is similar to the previous case.
- 2 • Suppose now that $S_H^+(x) \cap Q_1 \neq \emptyset$ and $S_H^+(x) \cap Q_2 \neq \emptyset$. We may assume that $S_H^-(x) \cap$
3 $\bigcup_{i=1}^k Q_i = \emptyset$, otherwise we are in one of the previous case, and we get the result. Let R
4 be a shortest (u, x) -path in D' . Then every vertex in $R - u$ is a vertex of $H - \bigcup_{i=1}^k Q_i$.
5 Let S_1 be a shortest directed (x, Q_1) -path and S_2 be a shortest directed (x, Q_2) -path.
6 For $i = 1, 2$, let z_i be the terminus of S_i . We may assume that all the internal vertices
7 of S_1 and S_2 are in $H - \bigcup_{i=1}^k Q_i$ for otherwise one vertex z among z_1 and z_2 satisfies
8 the condition of one of the previous cases (up to a permutation of the labels). Then the
9 union of paths $P_2, \dots, P_{p+3}, R, S_1, S_2$ and $P_1[z_1, v]$ form a $TT_4(p)$ -subdivision.
- 10 • If $S_H^-(x) \cap Q_1 \neq \emptyset$ and $S_H^-(x) \cap Q_2 \neq \emptyset$, the proof is similar to the previous case by
11 directional symmetry.

12 Assume now that $|I(x)| < 2$ for all $x \in V(H)$. Then, by Claim 66.1, $|I(x)| = 1$ for all
13 $x \in V(H)$. For $1 \leq i \leq k$, let $V_i = \{x \mid I(x) = \{i\}\}$. Then (V_1, \dots, V_k) is a partition of $V(H)$.
14 Moreover, by definition, there is no arc between two distinct parts of this partition. In ad-
15 dition, in $D' \langle X_i \cup \{u, v\} \rangle$, there cannot be two internally disjoint directed (u, v) -paths, for
16 otherwise it would contradicts the maximality of (P_1, \dots, P_k) . Hence, D' contains no TT_4^* -
17 subdivision, and so no $TT_4(p)$ -subdivision. \diamond

18 This finishes the proof of Theorem 66. \square

19 **Corollary 67.** *For all non-negative integer p , the $TT_4^*(p)$ -SUBDIVISION problem can be*
20 *solved in $O(n^3(n+m))$.*

21 *Proof.* Observe that a graph D contains a $TT_4^*(p)$ -subdivision with source u and sink v , if
22 and only if the graph $D \cup \{uv\}$ contains a $TT_4(p)$ -subdivision. Hence by just adding the arc
23 uv to D if it does not exists in the above algorithm, we obtain a polynomial-time algorithm
24 for $TT_4^*(p)$ -SUBDIVISION. \square

25 9.4. Subdivisions of digraphs with three vertices

26 Let us denote by \vec{K}_n the complete digraph on n vertices, in which there is an arc uv for
27 any two distinct vertices u and v . Let D_3 be the digraph obtained from \vec{K}_3 by removing an
28 arc.

29 **Theorem 68.** *Let F be a digraph on three vertices. Then F -SUBDIVISION is polynomial-*
30 *time solvable unless $F = \vec{K}_3$ in which case it is NP-complete.*

31 *Proof.* If F is neither D_3 nor \vec{K}_3 , then it is either a disjoint union of spiders, or a spindle, or
32 a bispindle, or the lollipop (or its converse), or a windmill, and so F -SUBDIVISION can be
33 solved in polynomial time by virtue of the results of the previous sections. If $F = \vec{K}_3$, then
34 F -SUBDIVISION is NP-complete by Corollary 10.

35 It remains to prove that D_3 -SUBDIVISION is polynomial-time solvable.

36 The *bulky vertex* of a D_3 -subdivision S is the unique vertex of S with degree 4. We
37 now give a procedure that given a vertex v , two of its out-neighbours s_1, s_2 and two of its
38 in-neighbours t_1, t_2 check if there is a D_3 -subdivision S in which v is the bulky vertex and
39 $\{vs_1, vs_2, t_1v, t_2v\} \in A(S)$. Such a subdivision will be called *suitable*.

1 Applying a Menger algorithm, check if in $D - v$ there are two disjoint directed paths P_1
2 and P_2 from $\{s_1, s_2\}$ to $\{t_1, t_2\}$. If not, then D certainly does not contain any suitable D_3 -
3 subdivision. If yes, then check if there is a directed path Q from P_1 to P_2 or from P_2 to P_1 .
4 If such a Q exists, then P_1, P_2, Q together with v and the arcs vs_1, vs_2, t_1v, t_2v form a suitable
5 D_3 -subdivision. If not, then no suitable D_3 -subdivision using the chosen arcs exists, because
6 there is no vertex $s \in \{s_1, s_2\}$ such that there exists in $D - v$ both a directed (s, t_1) -path and a
7 directed (s, t_2) -path.

8 A D_3 -subdivision is clearly suitable with respect to its bulky vertex and its neighbours
9 in this subdivision. Hence checking if there is a suitable D_3 -subdivision for every 5-tuple
10 (v, s_1, s_2, t_1, t_2) such that s_1, s_2 are out-neighbours of v and t_1, t_2 are out-neighbours yields a
11 polynomial-time algorithm to decide if there is a D_3 -subdivision in a digraph. \square

12 9.5. Subdivision of oriented paths and cycles

13 **Conjecture 69.** If F is an oriented path or cycle, then F -SUBDIVISION is polynomial-time
14 solvable.

15 **Proposition 70.** If P is an oriented path with at most four blocks, then P -SUBDIVISION is
16 polynomial-time solvable.

17 An *antidirected path* is an oriented path in which every vertex has either in-degree 0 or
18 out-degree 0.

19 **Theorem 71.** If P is an antidirected path, then P -SUBDIVISION is polynomial-time solvable.

20 *Proof.* Let $P = (a_1, \dots, a_p)$ be an antidirected path. By directional symmetry, we may as-
21 sume that a_i has indegree 0 in P if and only if i is odd.

22 Let D be a digraph. For a p -tuple of vertices (v_1, \dots, v_p) , we shall describe a procedure
23 that either returns a P -subdivision, or returns that there exists no P -subdivision in which each
24 v_i is the image of a_i . Then applying this procedure for all p -tuples of vertices, we obtain the
25 desired algorithm to finding a P -subdivision.

26 The procedure is as follows: For all odd (resp. even) i , we remove all the arcs entering v_i
27 (resp. leaving v_i) in D . Let D' be the resulting digraph. Clearly, D contains a P -subdivision in
28 which each v_i is the image of a_i if and only if D' does. In $UG(D')$, we check if there is a path
29 \tilde{Q} going through v_1, \dots, v_p in this order. This can be done by checking for a linkage from
30 $(v_1, v_2, \dots, v_{p-1})$ to (v_2, v_3, \dots, v_p) and thus in polynomial time by Robertson and Seymour
31 algorithm [18].

32 If no such \tilde{Q} is found, then D' (and thus D) contains certainly no P -subdivision in which
33 each v_i is the image of a_i .

34 If such a \tilde{Q} is found, let Q be the oriented path corresponding to \tilde{Q} in D' . Since v_i is a
35 source in D' when i is odd, and a sink in D' when i is even, the path Q has at least $p - 1$
36 blocks, and so contains a subdivision of P . \square

37 **Remark 72.** Using the same technique, one can show that if P is an oriented path, all blocks
38 of which have length one except possibly two consecutive blocks, then P -SUBDIVISION is
39 polynomial-time solvable.

10. Concluding remarks

The following conjecture, due to Seymour (private communication, 2011) would imply a number of the results on polynomial instances in the previous sections.

Conjecture 73 (Seymour). F -SUBDIVISION is polynomial-time solvable when F is a planar digraph with no big vertices.

This conjecture would indeed be implied by the following conjecture. An arc uv in a digraph is *contractible* if $\min\{d^+(u), d^-(v)\} = 1$. A *minor* of a digraph D is any subdigraph \tilde{D} of D which can be obtained from a subdigraph H of D by contracting zero or more contractible arcs of H . For $k = 1, 2, \dots, k$ the digraph J_k is obtained from the union of k directed cycles (each of length $2k$) C_1, C_2, \dots, C_k , where $C_i = u_{i,1}v_{i,1}u_{i,2}v_{i,2} \dots u_{i,k}v_{i,k}u_{i,1}$, for $i = 1, 2, \dots, k$ and paths P_i, Q_i , $i = 1, 2, \dots, k$, where $P_i = u_{1,i}u_{2,i} \dots u_{k,i}$ and $Q_i = v_{k,i}v_{k,i-1} \dots v_{k,1}$ for $i = 1, 2, \dots, k$.

Conjecture 74 (Johnson et al. [14]). For every positive integer k there exists $N(k)$ such that the following holds: If a digraph D has directed treewidth more than $N(k)$, then D contains a minor isomorphic to J_k .

If the directed tree-width of D is bounded, then, by Theorem 46, F -SUBDIVISION can be solved in polynomial time. If, on the other hand, the directed tree-width of D is unbounded, then (if the algorithmic version of the conjecture also holds) we can find a minor isomorphic to J_k for a sufficiently large k and presumably use this to realize the desired subdivision using the fact the F is planar and has no big vertices.

Conjecture 75. F -SUBDIVISION is NP-complete for every non-planar digraph F .

For any positive integer p , let us denote by C_p , the class of digraphs in which all directed cycles have length at most p . Then C_1 may be seen as the class of acyclic digraphs.

Problem 76. Is k -LINKAGE polynomial-time solvable on C_p ?

Thomassen proved [20] that for every natural number p there exists a p -strongly connected digraph D_p which is not 2-linked, that is, there exists no linkage from (s_1, s_2) to (t_1, t_2) for some choice of distinct vertices s_1, s_2, t_1, t_2 of D_p .

Problem 77. Let F be a fixed digraph. Does there exist k_F such that every k_F -strongly connected digraph contains an F -subdivision or at least such that F -SUBDIVISION is polynomial-time solvable when restricted to k_F -strongly connected digraphs?

Note that if F_1 -SUBDIVISION and F_2 -SUBDIVISION are both polynomial-time solvable, then $(F_1 + F_2)$ -SUBDIVISION is sometimes polynomial-time solvable and sometimes NP-complete. For example, if F_1 is the disjoint union of spiders and F_2 -SUBDIVISION is polynomial-time solvable, then $(F_1 + F_2)$ -SUBDIVISION is polynomial time solvable. On the other hand, assume that F_1 and F_2 are $(1, 2, 2)$ -spindles. Then by Proposition 22, F_1 -SUBDIVISION and F_2 -SUBDIVISION are both polynomial-time solvable, but according to Theorem 8, $(F_1 + F_2)$ -SUBDIVISION is NP-complete.

Hence for every two digraphs F_1 and F_2 such that F_1 -SUBDIVISION and F_2 -SUBDIVISION have been proved to be polynomial-time solvable, it is natural to ask for the complexity of $(F_1 + F_2)$ -SUBDIVISION. In particular, the following problem is one of the first to study.

1 **Problem 78.** Let F_1 and F_2 be two $(1,2)$ -spindles, i.e. transitive tournaments of order 3.
2 What is the complexity of $(F_1 + F_2)$ -SUBDIVISION?

3 **Acknowledgement**

4 The authors would like to thank Nicolas Trotignon for stimulating discussion and sug-
5 gesting them the FORK problem.

- 6 [1] J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications*.
7 Springer Monographs in Mathematics. Springer Verlag, London, 2008.
- 8 [2] J. Bang-Jensen, F. Havet and N. Trotignon. Finding an induced subdivision of a di-
9 graph. *Theoretical Computer Science*, 2012 to appear.
- 10 [3] J. Bang-Jensen and M. Kriesell. Disjoint directed and undirected paths and cycles in
11 digraphs *Theoretical Computer Science* 410, 5138–5144, 2009.
- 12 [4] D. Berwanger, A. Dawar, P. Hunter, and S. Kreutzer. DAG-width and parity games.
13 In STACS 2006, Proc. 23rd Symposium on Theoretical Aspects of Computer Science,
14 Volume 3884 of Lect. Notes Comput. Sci., pages 524-536. Springer-Verlag, 2006.
- 15 [5] J. A. Bondy and U. S. R. Murty. *Graph theory, vol. 244 of Graduate Texts in Mathe-*
16 *matics*. Springer, New York, 2008.
- 17 [6] M. Chudnovsky, A. Scott and P.D. Seymour. Disjoint paths in tournaments.
18 Manuscript 2012.
- 19 [7] M. Chudnovsky and P.D. Seymour. Excluding induced subgraphs. In *Surveys in*
20 *Combinatorics*, volume 346, pages 99–119. London Mathematical Society Lecture
21 Notes Series, 2007.
- 22 [8] M. Chudnovsky and P.D. Seymour. The three-in-a-tree problem. *Combinatorica*,
23 Volume 30, pages 387–417, 2010.
- 24 [9] B. Courcelle and J. Engelfriet. *Graph structure and monadic second-order logic. A*
25 *language-theoretic approach. Encyclopedia of Mathematics and its Applications*, 138.
26 Cambridge University Press, Cambridge, 2012.
- 27 [10] S. Fortune, J.E. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism
28 problem. *Theoretical Computer Science*, 10:111–121, 1980.
- 29 [11] H. N. Gabow and S. Nie. Finding a long directed cycle. In *Proceedings of the fif-*
30 *teenth annual ACM-SIAM symposium on Discrete algorithms (SODA '04)*. Society
31 for Industrial and Applied Mathematics, Philadelphia, PA, USA, 49–58, 2004.
- 32 [12] H. N. Gabow and S. Nie. Finding a long directed cycle. *ACM Trans. Algorithms*,
33 4(1):Article 7, 21 pages, 2008.
- 34 [13] M. Grohe and M. Grüber Parameterized approximability of the disjoint cycle problem.
35 In *Proceedings of the 34th International Colloquium on Automata, Languages and*
36 *Programming (ICALP'07)*, volume 4596 of LNCS, pages 363–374, 2007.

- 1 [14] T. Johnson, N. Robertson, P.D. Seymour and R. Thomas Directed tree-width. *J.*
2 *Combinatorial Theory, Ser. B* **82** (2001)
- 3 [15] B. Lévêque, D. Lin, F. Maffray, and N. Trotignon. Detecting induced subgraphs.
4 *Discrete Applied Mathematics*, **157** (2009) 3540–3551.
- 5 [16] W. McCuaig, “Intercyclic digraphs”, Graph structure theory (Seattle, WA, 1991), Con-
6 temp. Math. 147, Amer. Math. Soc., Providence, RI (1993), 203–245.
- 7 [17] B. Reed, N. Robertson, P.D. Seymour, and R. Thomas. Packing directed circuits.
8 *Combinatorica*, 16(4):535–554, 1996.
- 9 [18] N. Robertson and P.D. Seymour. Graph minors. XIII. The disjoint paths problem.
10 *Journal of Combinatorial Theory, Series B*, 63:65–110, 1995.
- 11 [19] A. Slivkins. Parameterized tractability of edge-disjoint paths on directed acyclic
12 graphs. In G. D. Battista and U. Zwick, editors, *Proceedings of the 11th Annual Eu-*
13 *ropean Symposium on Algorithms, ESA 03*, volume 2832 of *LNCS*, pages 482–493,
14 2003.
- 15 [20] C. Thomassen. Highly connected non-2-linked digraphs. *Combinatorica*, 11(4):393-
16 395, 1991.