



## 4. Euler Digraphs

Magnus Wahlström

An Euler digraph is a connected digraph where every vertex has in-degree equal to its out-degree. The name, of course, comes from the directed version of Euler's theorem. Recall that an Euler tour in a digraph is a directed closed walk that uses each arc exactly once. Then in this terminology, by the famous theorem of Euler, a digraph admits an Euler tour if and only if it is an Euler digraph.

However, beyond this point of historical interest, Euler digraphs are also interesting since they form a class of intermediate complexity between undirected graphs and fully general digraphs for many problems. For example, consider the  $k$ -LINKAGE and WEAK  $k$ -LINKAGE problems. Recall that in these problems, the input is a digraph  $D = (V, A)$  together with  $k$ -tuples  $(s_1, \dots, s_k)$  and  $(t_1, \dots, t_k)$  of vertices, and the goal is to find internally vertex-disjoint paths (respectively, arc-disjoint paths) from  $s_i$  to  $t_i$  for every  $i \in [k]$ . For undirected graphs, both problems are famously FPT parameterized by  $k$ , as a central result of the graph minor theory of Robertson and Seymour [36]. For general digraphs, both variants are  $\mathcal{NP}$ -hard already for  $k = 2$  as shown by Fortune, Hopcroft and Wyllie [15]. For Euler digraphs, the  $k$ -LINKAGE PROBLEM is in general  $\mathcal{NP}$ -hard, but the WEAK  $k$ -LINKAGE PROBLEM is in  $\mathcal{P}$  at least up to  $k = 3$ , and it is a long open question whether the WEAK  $k$ -LINKAGE PROBLEM is in  $\mathcal{P}$  for every fixed  $k$  or even FPT. (We discuss these problems later in this chapter.)

For another example, consider the concepts regarding classes of graphs and digraphs of restricted structure, e.g., bounded width. For undirected graphs, although many alternatives have been considered, arguably the established standard width notion is **bounded treewidth**, and the related notion of **bounded pathwidth**. These width measures have several desirable properties, not least including algorithmic applications such as linear-time FPT algorithms for a multitude of problems when parameterized by the width  $k$ . On the other hand, for directed graphs, although directed analogues of these basic width notions exist, not only are the basic definitions significantly more complex, but the algorithmic implications are also typically weaker, e.g., most problems would not be FPT parameterized by directed pathwidth. We will see that in the general case there is no significant difference between the di-

---

M. Wahlström (✉)

Department of Computer Science,

Royal Holloway University, Egham Hill, Egham TW20 0EX, England

e-mail: Magnus.Wahlstrom@rhul.ac.uk

© Springer International Publishing AG, part of Springer Nature 2018

J. Bang-Jensen and G. Gutin (eds.), *Classes of Directed Graphs*, Springer Monographs in Mathematics, [https://doi.org/10.1007/978-3-319-71840-8\\_4](https://doi.org/10.1007/978-3-319-71840-8_4)

rected width measures on Euler digraphs and on general digraphs, but (unlike for general digraphs) if we additionally impose that an Euler digraph is of bounded degree, then the undirected and directed versions of pathwidth and treewidth coincide up to a constant factor.

However, interestingly, there are also a few problems which are easier to deal with on Euler digraphs than on undirected graphs (even undirected Euler graphs). We will see two main examples of this. The first is the so-called BEST theorem, which states that the number of Euler tours in a digraph can be counted efficiently; the same is not true for undirected graphs, where the corresponding problem is  $\#P$ -hard. The second, less well known example is ARC MULTIWAY CUT (see later for definitions), which is  $\mathcal{NP}$ -hard both for undirected Euler graphs and for general digraphs, but which admits a simple polynomial-time algorithm on Euler digraphs. We will cover both of these results in the following.

This chapter is structured as follows. We begin with some basic constructions and observations in Section 4.1; in Section 4.2 we consider the BEST theorem and other questions of Euler tours; in Section 4.3 we consider notions of Euler digraphs of bounded width; in Section 4.4 we review problems related to packing and hitting cycles, and in Section 4.5 we review general problems of path-packing and linkages.

## 4.1 Basic Constructions and Properties

To relax the definition of Euler digraphs slightly, let a digraph  $D$  be **balanced** if every vertex has in-degree equal to its out-degree, but with no requirement that  $D$  be connected. Most of our algorithmic results will apply to balanced digraphs as well as Euler digraphs, and it will sometimes be convenient to not have to require connectivity. In fact, we will frequently gloss over the difference between the two notions. Also note that a balanced digraph is strongly connected if and only if it is connected.

We review two basic properties of balanced digraphs. First, we note that a balanced digraph can be exhaustively decomposed into simple directed cycles. Second, we note that for every balanced digraph  $D = (V, A)$  and every vertex  $v \in V$ , there are  $d_D^+(v)$  pairwise arc-disjoint cycles through  $v$ . Both of these results follow via simple induction (in the latter case using the existence of an Euler tour for the component).

### 4.1.1 Cuts in Euler Digraphs

The following observation is the underlying source of many of the tractability results in this chapter.

**Proposition 4.1.1** *Let  $D = (V, A)$  be a balanced digraph. For any set  $S \subseteq V$ , it holds that  $d^+(S) = d^-(S)$ .*

**Proof:** Note that  $\sum_{v \in S} d^+(v) = \sum_{v \in S} d^-(v)$  since  $D$  is balanced. Thus we have

$$0 = \sum_{v \in S} d^+(v) - \sum_{v \in S} d^-(v) = (d^+(S) + |(S, S)|) - (d^-(S) + |(S, S)|),$$

so  $d^+(S) = d^-(S)$ . □

The following is an easy but important consequence.

**Proposition 4.1.2** *Let  $D = (V, A)$  be a balanced digraph and let  $G = \text{UMG}(D)$  be the underlying multigraph of  $D$ . Then for any  $S \subseteq V$ , we have  $d_D^+(S) = d_D^-(S) = d_G(S)/2$ .*

This implies that tools developed for edge cuts in undirected graphs will transfer directly to arc-cuts in Euler digraphs. In particular, for an Euler digraph  $D = (V, A)$  with two vertices  $s, t \in V$ , using the **treewidth reduction theorem** of Marx, O’Sullivan and Razgon [32] it follows that all minimal  $s$ - $t$ - and  $t$ - $s$  arc-cuts in  $D$  of size at most  $k$  are contained in a subgraph of  $D$  of treewidth bounded as a function of  $k$ . Although we will not need this result in the remainder of the chapter, it is a powerful tool for FPT algorithms in undirected graphs and worth observing for its potential applications. The same can be said for other advanced methods for producing FPT algorithms for undirected graph problems. In particular, there is a method of designing FPT algorithms via **recursive understanding**, which was pioneered by Kawarabayashi and Thorup for  $k$ -WAY CUT [28] and was developed further and made more efficient using the method of **randomized contractions** by Chitnis, Cygan, Hajiaghayi, Pilipczuk and Pilipczuk [6]. Another related work is the special tree decomposition used by Cygan, Lokshtanov, Pilipczuk, Pilipczuk and Saurabh for the MINIMUM BISECTION problem [10]. All of these cases represent advanced and successful methods for designing FPT algorithms for cut problems on undirected graphs, which a priori seem not to transfer in a useful way to general digraphs, but which may be worth considering for the case of Euler digraphs.

#### 4.1.2 Hardness Constructions

We review two simple constructions that will be useful in showing problem hardness on Euler digraphs. Recall that for an undirected graph  $G = (V, E)$ , the **complete biorientation** of  $G$  is a digraph  $\vec{G}$  with vertex set  $V$  and with a pair of arcs  $uv, vu$  for every edge  $\{u, v\} \in E$ . Clearly,  $\vec{G}$  is balanced, and Euler if  $G$  is connected. This construction can frequently be used to show that problems on Euler digraphs are “at least as hard” as the corresponding problem on undirected graphs.

The second construction is as follows. Let  $D = (V, A)$  be a connected digraph, and define  $b_D(v) = d_D^+(v) - d_D^-(v)$  as the **balance number** of  $v$

in  $D$ . The **Euler two-vertex extension** of  $D$  is the directed multigraph obtained by adding two vertices  $x, y \notin V$  to  $D$ , then for every vertex  $v \in V$  adding  $|b(v)|$  arcs  $vx$  if  $b(v) < 0$  and  $b(v)$  arcs  $yv$  if  $b(v) > 0$ , then finally adding  $d^-(x)$  arcs  $xy$ . It is clear that the resulting directed multigraph is Euler. If the situation does not allow parallel arcs, we may simply subdivide all arcs into and out of  $x$  and  $y$ . The significance of this construction is that by simply deleting  $x$  or  $y$  from  $D'$ , we eliminate all paths not present in the original graph  $D$ .

Above all, this means that for many problems which come in a “vertex version” and an “arc version”, the vertex version is usually equally difficult on an Euler digraph as on general digraphs, while the arc version can be significantly easier. For example, considering the problems mentioned in the introduction, it is trivial to show that VERTEX MULTIWAY CUT (i.e., the vertex-deletion version), the  $k$ -LINKAGE PROBLEM and the VERTEX-DISJOINT CYCLE PACKING problem are all as hard on Euler digraphs as on general digraphs (in the case of the  $k$ -LINKAGE PROBLEM increasing  $k$  by one), but ARC MULTIWAY CUT, the WEAK  $k$ -LINKAGE PROBLEM and ARC-DISJOINT CYCLE PACKING are all significantly easier on Euler digraphs, as we see in the results surveyed in this chapter.

### 4.1.3 Splitting Off and Other Operations

One frequently used operation in this chapter is the **splitting-off** operation. Let  $D = (V, A)$  be an Euler digraph and  $uv, vw$  a pair of arcs in  $D$ . Then **splitting off  $uv$  and  $vw$  in  $D$**  refers to the operation of deleting the arcs  $uv$  and  $vw$ , and creating a new arc  $uw$ . It is clear that this operation preserves balance (if not necessarily connectivity). If all vertices  $u, v, w$  are distinct, and if the result is a digraph (as opposed to a directed multigraph or pseudograph), then we refer to  $uv$  and  $vw$  as a **simple splitting pair**. Let us make a simple observation.

**Proposition 4.1.3** *Let  $D = (V, A)$  be a balanced digraph with no simple splitting pair. Then every connected component of  $D$  is a complete digraph.*

**Proof:** If  $D$  has no simple splitting pair, then for every pair of arcs  $uv$  and  $vw$  such that  $u \neq w$ , the arc  $uw$  already exists, i.e.,  $D$  is transitive. It is well known (and easy to see) that a strongly connected transitive digraph must be complete. Since every connected component of a balanced digraph is also strongly connected, the result follows.  $\square$

Other operations that preserve the balance property of a digraph include arc contractions and the removal of a balanced subgraph.

We will also occasionally need the notion of a **minor** of a digraph. There are two variants of this definition, **butterfly minors** and **topological minors**. Let  $D = (V, A)$  be a digraph and  $uv \in A$  an arc. The arc  $uv$  is **butterfly contractible** in  $D$  if either  $uv$  is the only arc out of  $u$  or the only

arc into  $v$ . A **butterfly minor** of  $D$  is obtained from a subgraph of  $D$  by contracting butterfly contractible arcs. Alternatively, a **topological minor** of  $D$  is produced from arc contractions of a subgraph of  $D$  by contracting an arc  $uv$  only if either  $u$  or  $v$  has in-degree and out-degree 1.

## 4.2 Problems Regarding Euler Tours

Let us begin as a warm-up with the so-called **BEST theorem**, showing that Euler tours of an Euler digraph can be counted in polynomial time.

The BEST theorem was implicit in the work of Tutte and Smith [41] and was shown in full by van Aardenne-Ehrenfest and de Bruijn [42]; the theorem takes its name from the authors.

**Theorem 4.2.1 (BEST theorem)** *Let  $D = (V, A)$  be an Euler digraph, and  $w \in V$  an arbitrary vertex. The number of Euler tours in  $D$  is*

$$t_D(w) \prod_{v \in V} (d(v) - 1)!,$$

where  $t_D(w)$  is the number of out-branchings of  $D$  rooted in  $w$ . In particular, there is a polynomial-time algorithm for counting the number of directed Euler tours.

**Proof:** Let  $ww' \in A$ , and let  $T$  be an Euler tour of  $D$ . Observe that  $T$  induces a permutation  $\pi_v$  of the out-arcs of  $v$  for every  $v \in V$ , according to the order in which these arcs are visited in  $T$ , starting the count from  $ww'$ . Also note that  $T$  can be recovered from this collection of permutations, and conversely, every such collection of permutations  $\{\pi_v\}_{v \in V}$  defines a closed trail in  $D$  containing the arc  $ww'$ , although not every collection of permutations induces an Euler tour. For a collection of permutations  $P = \{\pi_v\}_{v \in V}$ , let  $F(P) = \{\pi_v(d^+(v)) \mid v \in V, v \neq w\}$  be the set containing the last outgoing arc from every vertex except  $w$ . We claim that  $P$  defines an Euler tour if and only if  $F(P)$  is an in-branching in  $D$  rooted in  $w$ .

On the one hand, let  $P$  be defined via an Euler tour  $T$ . The set  $F(P)$  forms a digraph where every vertex except  $w$  has out-degree 1; hence  $F(P)$  forms an in-branching rooted in  $w$  if and only if it is acyclic. We claim that for every arc  $uv \in F(P)$ ,  $v \neq w$ , the last out-arc of  $u$  is visited before the last out-arc of  $v$  in  $T$ , if we begin the counting from  $ww'$ . Indeed,  $uv$  is the last out-arc of  $u$  visited in  $T$  by definition, and clearly whatever arc follows  $uv$  in  $T$  is an out-arc of  $v$  visited after  $uv$ . Since the out-degree of  $w$  in  $F(P)$  is zero, it follows that  $F(P)$  is acyclic, and that  $F(P)$  is an in-branching rooted in  $w$ .

On the other hand, let  $P$  be a collection of permutations such that  $F(P)$  forms an in-branching, and let  $T$  be the closed tour defined by  $P$  starting

from the arc  $ww'$ . Then  $D - T$  is a balanced digraph. Let  $H$  be a connected component of  $D - T$ . Clearly, for every vertex  $v$  that is not of degree zero in  $D - T$ , the out-arc of  $v$  in  $F(P)$  is contained in  $D - T$ . But then the set  $F_H$  of out-arcs of  $F(P)$  of vertices in  $H$  forms a subgraph of  $H$  where every vertex has out-degree 1, which necessarily contains a cycle. This contradicts our assumption on  $P$ .

The formula follows from this claim. For every in-branching  $B$  of  $D$  rooted in  $w$ , there are exactly  $\prod_{v \in V} (d^+(v) - 1)!$  collections  $P$  of permutations such that  $F(P) = B$ : for every vertex  $v \neq w$ , the in-branching  $B$  fixes the last out-arc of  $v$  in  $P$ , whereas for  $w$ , the out-arc  $ww'$  is the first out-arc of  $w$  by definition. Any choice of a permutation  $\pi_v$  on the remaining arcs does not affect  $F(P)$ . Finally, it is well known that the number of rooted in-branchings can be counted in polynomial time using Tutte's matrix-tree theorem, see e.g. [2].  $\square$

Interestingly, both the Tutte–Smith paper and the van Aardenne-Ehrenfest and de Bruijn paper have as their main interests something other than counting Euler cycles. Tutte and Smith considered the problem of tracing a 4-regular undirected planar drawing without lifting the pen, in such a way that the line traced never crosses itself, and showed that this can be reduced to a question of counting Euler tours in a digraph (using arguments similar to those used in the polynomial-time algorithm for counting planar matchings). On the other hand, van Aardenne-Ehrenfest and de Bruijn arrived at the question via their study of string problems, specifically De Bruijn sequences, cyclic sequences over an alphabet  $\Sigma$  that contain every  $n$ -tuple over  $\Sigma$  exactly once.

Other questions on Euler tours include the following. We say that two Euler tours are **compatible** if they use only distinct transitions at every vertex, i.e., for every vertex  $v$  with an in-arc  $uv$  and out-arc  $vw$ , at most one of the tours contains the transition from  $uv$  to  $vw$ . Fleischner and Jackson [14] showed that every Euler digraph  $D$  of minimum degree  $2k$  contains  $\lfloor \frac{k}{2} \rfloor$  pairwise compatible Euler tours, and conjectured that the bound can be improved to  $k - 2$ .

### 4.3 Euler Digraphs of Bounded Width

The notion of **width measures** has been a highly successful approach for studying graphs of restricted structure, especially for undirected graphs. The rough idea is that a graph of simple structure can be **decomposed** recursively into pieces that interact with each other only in a limited way, where each piece is either very simple (e.g., constant size) or can itself be further decomposed. For undirected graphs, arguably the default notion of bounded structure is having bounded **treewidth**, motivated by a plethora of algorithmic results, e.g., methods such as Courcelle's theorem [9] or other dynamic

programming results for efficiently solving  $\mathcal{NP}$ -hard problems on a graph, given a decomposition of the graph that shows it has bounded treewidth. However, many other width notions also exist, both less expressive ones such as **pathwidth** or **treedepth**, and more expressive ones such as **rankwidth**.

For directed graphs, the story of bounded width measures has arguably been more limited in terms of algorithmic applications. Although we do have a growing understanding of the structure of graphs of small or large width under various natural directed width notions, compared to the undirected case, the algorithmic implications of bounded directed widths are generally weaker (see Chapter 9 for results on bounded width measures on digraphs). In the undirected case, it is a common occurrence that a problem is FPT parameterized by treewidth, in fact often with a running time such as  $O(f(k) \cdot n)$  that is linear in the order of the graph. For directed width notions, it is far more common that a parameterized problem is  $W[1]$ -hard – meaning that, while it may be polynomial-time tractable on graphs of (say) bounded directed treewidth, the running time is of the less appealing form  $O(n^{f(k)})$  and FPT algorithms are not expected to exist (see Section 1.11). In fact, one could argue that the width measure that has had the widest success for digraphs in terms of FPT algorithms is simply the undirected treewidth, i.e., the treewidth of the underlying undirected graph.

In view of this, it is natural to ask about the structure of Euler digraphs in this perspective. In particular, to what extent do the directed and undirected notions of treewidth and pathwidth differ for Euler digraphs?

We prove a simple result in this direction. First, we may observe that if a width notion is closed under taking induced subgraphs, then there is no sense in studying it in full generality for Euler digraphs, since (by the Euler vertex-extension) every digraph on  $n$  vertices is an induced subgraph of a (not necessarily simple) Euler digraph on  $n+2$  vertices. Since most width measures are closed under taking induced subgraphs and closed or approximately closed under subdividing parallel arcs, in general Euler digraphs of bounded width will not have any extra structure that is not present in other digraphs. On the other hand, we may observe that the above reduction creates a vertex of unbounded degree, and ask whether the situation changes under a combined bound of bounded width and bounded maximum degree. Indeed, it is known that an Euler digraph of maximum degree  $d$  and directed treewidth  $k$  has undirected treewidth at most  $O(dk)$  [26]. Hence for bounded-degree Euler digraphs, the difference between directed and undirected width disappears. (This is certainly not true for general digraphs, as, e.g., an acyclic grid has total degree 4, unbounded undirected treewidth, and directed pathwidth 0, as we will see below.)

For more information on digraphs of bounded width, see Chapter 9 of this book. In the following, we omit the technical details of directed treewidth, and prove a simpler result that relates the undirected and directed notions of bounded pathwidth.

### 4.3.1 Cutwidth and Bounded Pathwidth

We begin by studying the directed pathwidth of Euler digraphs. Let us first recall the notions.

We need two variants of pathwidth, undirected and directed. We begin with the undirected version. Let  $G = (V, E)$  be an undirected graph. A **path decomposition** of  $G$  is a sequence of vertex sets  $X_1, \dots, X_s$  called **bags** such that  $\bigcup_{i \in [s]} X_i = V$ , where the following hold.

1. For every edge  $uv \in E$ , there is a bag  $X_i$ ,  $i \in [s]$ , such that  $u, v \in X_i$ , and
2. for every triple of indices  $i < j < k$ ,  $i, j, k \in [s]$ , we have  $X_i \cap X_k \subseteq X_j$ .  
Equivalently,  $\{i \in [s] \mid v \in X_i\}$  forms an interval for every  $v \in V$ .

The **width** of the decomposition is  $\max_{i \in [s]} |X_i| - 1$ . The **pathwidth of  $G$**  is the minimum width of a path decomposition of  $G$ . We let the **undirected pathwidth** of a digraph  $D = (V, A)$  refer to the pathwidth of its underlying undirected graph  $UG(D)$ .

Analogously, let  $D = (V, A)$  be a digraph. A **directed path decomposition** is a sequence of vertex sets  $X_1, \dots, X_s$ , again called bags, such that  $\bigcup_{i \in [s]} X_i = V$ , where the following hold.

1. For every arc  $uv \in E$ , there are indices  $i \leq j$ ,  $i, j \in [s]$  such that  $u \in X_i$  and  $v \in X_j$ , and
2. as in the undirected case, for every triple of indices  $i < j < k$ ,  $i, j, k \in [s]$  we have  $X_i \cap X_k \subseteq X_j$ .

The **width** of the decomposition is  $\max_{i \in [s]} |X_i| - 1$  and the **directed pathwidth of  $D$**  is the minimum width of a directed path decomposition of  $D$ .

We will also need a width measure that is less commonly used in general, but which will be highly relevant for Euler digraphs. Let  $D = (V, A)$  be a digraph, and let  $\sigma = v_1 \dots v_n$  be an ordering of  $V$ . The **undirected cutwidth of  $\sigma$**  equals the maximum over  $i$  of the number of arcs between  $\{v_1, \dots, v_i\}$  and  $\{v_{i+1}, \dots, v_n\}$ . The **undirected cutwidth of  $D$**  is the minimum undirected cutwidth over all orderings of  $V$ . Note that this is by its nature an undirected width measure, i.e., we do not distinguish arcs by their direction.

We make a few simple observations.

1. For any digraph  $D = (V, A)$ , the directed pathwidth is at most the undirected pathwidth, which in turn is at most the undirected cutwidth. Indeed, the former is trivial, and given any ordering  $\sigma$  of undirected cutwidth  $k$  it is easy to produce a path decomposition of width  $k$ .
2. Both inequalities are strict. Indeed, a star has undirected pathwidth 1 but unbounded undirected cutwidth, and an acyclic grid has directed pathwidth 0 (and constant degree) but unbounded undirected pathwidth.
3. A digraph has directed pathwidth 0 if and only if it is an acyclic digraph. In particular, the only Euler digraph of directed pathwidth 0 is a single vertex.



We now observe formally (as already sketched) that there are Euler digraphs of constant directed pathwidth but unbounded undirected pathwidth.

**Lemma 4.3.1** *For every  $k \geq 1$ , there is an Euler digraph with undirected treewidth at least  $k$  but directed pathwidth 1.*

**Proof:** Let  $D$  be a  $k \times k$  acyclic grid, with all arcs oriented downwards and to the right. Add one additional vertex as in-neighbour of the top-left grid vertex, and another as out-neighbour of the bottom-right grid vertex, and observe that  $|b_D(v)| \leq 1$  for every vertex in the resulting graph  $D$ . Hence we can complete  $D$  into an Euler digraph by adding a single vertex  $x$  and for every unbalanced vertex  $v$  either an arc  $vx$  or  $xv$ , as required. Then  $UG(D)$  contains a  $k \times k$  grid and hence has treewidth at least  $k$ , whereas  $D$  has a path decomposition of width 1 formed by simply adding  $x$  to every bag in the decomposition of the acyclic digraph  $D - x$ .  $\square$

Finally, we have the following positive result.

**Lemma 4.3.2** *If  $D$  is an Euler digraph with directed pathwidth  $k$ , then  $D$  has undirected cutwidth at most  $k \cdot \Delta(D)$ .*

**Proof:** Let  $X_1, \dots, X_s$  be a directed path decomposition of  $D$  of width  $k$ , and construct a linear ordering  $\sigma$  of  $V(D)$  by first arbitrarily arranging the vertices of  $X_1$ , then the vertices of  $X_2 \setminus X_1$ , and so on until  $X_s$ . Let  $d = \Delta^+(D)$ . We claim that the ordering  $\sigma$  has undirected cutwidth at most  $2dk$ .

Let  $(L_i, R_i)$  be the vertex cut corresponding to some position  $i$  of  $\sigma$ , i.e.,  $L_i$  is the set of vertices ordered at or before position  $i$  in  $\sigma$ , and  $R_i = V(D) \setminus L_i$ . Assume that  $(L_i, R_i)$  cuts through the bag  $X_j$ ,  $j \in [s]$ , and let  $A = X_j \cap L_i$  and  $B = X_j \cap R_i$ ; assume  $A \neq \emptyset$ . We claim that every arc of  $(R_i, L_i)_D$  has either its tail in  $B$  or its head in  $A$ . Indeed, by definition every such arc has its tail in  $R_i$ , and any such arc with tail in  $R_i \setminus B$  has its head in a vertex still present in a bag  $X_{j'}$  for some  $j' > j$ ; hence the head is contained in  $X_j$ . Since there are  $|A|d$  arcs with head in  $A$  and  $|B|d$  arcs with tail in  $B$ , and  $|A| + |B| \leq k$ , there are at most  $dk$  such arcs. On the other hand, since  $D$  is Euler we have  $d^+(L_i) = d^-(L_i) \leq dk$ ; hence the undirected cutwidth of  $\sigma$  is at most  $2dk = k\Delta(D)$ .  $\square$

Since undirected pathwidth is sandwiched between undirected cutwidth and directed pathwidth, we also get that the undirected pathwidth of Euler digraphs of bounded degree and bounded directed pathwidth is bounded, as promised.

As noted, the above result also holds for the more general notion of **directed treewidth**: If  $D$  is an Euler digraph with maximum degree  $d$  and the directed treewidth of  $D$  is at most  $k$ , then  $D$  has undirected treewidth at

most  $O(dk)$  [26]. Directed treewidth is the most general of the various width-measures that serve as directed analogues of (undirected) treewidth, and has some interesting structural properties. See, in particular, the **directed grid theorem** [27] proved by Kawarabayashi and Kreutzer, which shows that every digraph of large enough directed treewidth contains a particular canonical obstacle known as a **cylindrical grid** of large order as a butterfly minor (see Theorem 9.3.14). Hence this shows that any Euler digraph that does not contain a large cylindrical grid as a butterfly minor, and which has bounded degree, also has bounded undirected treewidth.

## 4.4 Cycle-Packing and Cycle-Hitting

In this section, we consider problems of cycle-hitting and cycle-packing in Euler digraphs. More properly, we consider the following two problems. A **feedback arc set** of a digraph  $D$  is a set  $F$  of arcs of  $D$  such that  $D - F$  is acyclic. The problem FEEDBACK ARC SET takes as input a digraph  $D$  and an integer  $k$ , and asks whether  $D$  has a feedback arc set of cardinality  $k$ . Dually to this, given a digraph  $D$  and an integer  $k$ , ARC-DISJOINT CYCLES asks whether  $D$  contains a packing of  $k$  pairwise arc-disjoint cycles. For both problems, the vertex versions (FEEDBACK VERTEX SET, respectively VERTEX-DISJOINT CYCLES) are defined in the natural way.

Before we proceed, we observe that these vertex-versions are not easier in Euler digraphs than in general graphs.

**Lemma 4.4.1** *For both FEEDBACK VERTEX SET and VERTEX-DISJOINT CYCLES, there are polynomial-time reductions from the versions on general digraphs to the versions on Euler digraphs that increase the value of  $k$  by only 1.*

**Proof:** The reduction is the same for both problems. Let  $D$  be a given digraph  $D$ , and let  $D'$  be its Euler two-vertex extension with added vertices  $x, y$ . Add an additional pair of arcs  $xy, yx$  (and if needed, subdivide parallel arcs to acquire a simple digraph). For FEEDBACK VERTEX SET, it is now easy to observe that for every  $X \subseteq V(D)$ ,  $X$  is a feedback vertex set of  $D$  if and only if  $X + x$  is a feedback vertex set of  $D'$ , and that there is a minimum feedback vertex set  $X'$  of  $D'$  for which  $X' \cap (V(D') \setminus V(D)) = \{x\}$ . For VERTEX-DISJOINT CYCLES, note that every cycle that intersects  $V(D') \setminus V(D)$  intersects both  $x$  and  $y$ . Let  $C$  be the cycle on  $x$  and  $y$ , contained in  $V(D') \setminus V(D)$ . Then there exists an optimal cycle-packing that contains  $C$ , and having included  $C$ , the cycle-packing problem on  $D' - V(C)$  is equivalent to that on  $D$ .  $\square$

Regarding the hardness of these problems, we recall that FEEDBACK VERTEX SET is  $\mathcal{NP}$ -complete but FPT on digraphs by the algorithm of Chen, Liu, Lu, O'Sullivan, and Razgon [5], whereas VERTEX-DISJOINT-CYCLES on

general digraphs is in XP, i.e., has an algorithm with running time  $O(n^{f(k)})$  for some function  $f(k)$  due to Reed, Robertson, Seymour and Thomas [35], but is W[1]-hard due to the results of Slivkins [39]. The problem also has an **FPT approximation** due to Grohe and Grüber [18], building on the results of Reed *et al.* – i.e., a parameterized algorithm running in FPT time, which either reports that  $D$  does not contain  $k$  vertex-disjoint cycles, or returns  $g(k)$  vertex-disjoint cycles, for some growing, unbounded function  $g(k)$ . By the above reduction, all these statements hold for general digraphs as well as for the restriction to Euler digraphs (for the vertex versions).

In the rest of this section, we will consider the arc-versions of these problems, which differ significantly in behavior on Euler digraphs.

#### 4.4.1 Feedback Arc Set

First, let us consider FEEDBACK ARC SET. We begin by showing  $\mathcal{NP}$ -hardness; the reduction is easy, but its correctness proof is revealing. We need the following observation. Relative to an ordering  $(v_1, \dots, v_n)$  of the vertices of an Euler digraph  $D = (V, A)$ , the **backward arcs** are arcs  $v_i v_j \in A$  with  $i > j$ . Note that for any such ordering, the backward arcs form a feedback arc set, and conversely, if  $F$  is a minimal feedback arc set of  $D$ , then  $F$  is exactly the set of backward arcs for some acyclic ordering of  $D - F$ .

**Lemma 4.4.2** *The number of backward arcs of an ordering of vertices of an Euler digraph is invariant under cyclic shifts, i.e., for any Euler digraph  $D = (V, A)$  and any ordering  $(v_1, \dots, v_n)$  of  $V$ , the orderings  $(v_1, \dots, v_n)$  and  $(v_2, \dots, v_n, v_1)$  have the same number of backward arcs.*

**Proof:** When  $v_1$  is in the first position, every in-arc of  $v_1$  is a backward arc but no out-arc is; when  $v_1$  is in the last position, the opposite statement holds. Every other arc is unaffected by the change. Since  $d^+(v_1) = d^-(v)$ , the two orderings have the same number of backward arcs.  $\square$

An important corollary is that for every vertex  $v$  of an Euler digraph  $D$ , there exists a minimum feedback arc set of  $D$  that contains all out-arcs of  $v$ . Indeed, if  $F$  is a feedback arc set, let  $v_1, \dots, v_n$  be an acyclic ordering of the vertices of  $D - F$ , and rotate the ordering until it ends with  $v$ . Then the backward arcs of the new ordering form a feedback arc set  $F'$  of  $D$ , with  $|F| = |F'|$  and where every out-arc of  $v$  is contained in  $F'$ . The  $\mathcal{NP}$ -hardness reduction is now trivial. Recall that  $\text{fas}(D)$  denotes the size of a minimum feedback arc set of  $D$ .

**Lemma 4.4.3** FEEDBACK ARC SET is  $\mathcal{NP}$ -hard on Euler digraphs.

**Proof:** We show a reduction from FEEDBACK ARC SET on general digraphs. Let  $D = (V, A)$  be a digraph, and let  $D'$  be its Euler two-vertex extension, with added vertices  $x, y$ . We claim that  $\text{fas}(D') = \text{fas}(D) + d^+(y)$ . Indeed,

let  $Y \subseteq A(D')$  be the set of out-arcs of  $y$ . By the above observation, there is a minimum feedback arc set that includes  $Y$ , and every directed cycle in  $D' - Y$  is also contained in  $D$ .  $\square$

Regarding properties of approximation and parameterized complexity, the simple reduction used above gives us no lower bounds, since the output parameter value is unbounded in  $k$ . We are also not aware of any lower bounds-preserving reduction from FEEDBACK VERTEX SET on undirected graphs to FEEDBACK ARC SET on Euler digraphs. Still, it can be interesting to compare with what is known for general digraphs and for FEEDBACK VERTEX SET on undirected graphs (recall that FEEDBACK EDGE SET can be solved in polynomial time, and therefore does not serve as a point of comparison).

For (unweighted) FEEDBACK ARC SET on general digraphs, the best approximation result is an  $O(\log \tau^* \log \log \tau^*)$ -approximation, where  $\tau^* \leq \text{fas}(D)$  is the cost of a natural LP-relaxation of the problem [13]. The problem does not admit a constant-factor approximation under the Unique Games Conjecture (see Guruswami and Lee [20]). The problem has an FPT algorithm with a running time of  $O^*(4^k k!)$  [5], and it is a famous open problem whether it has a single-exponential FPT algorithm, i.e., an FPT algorithm with a running time of  $O^*(2^{O(k)})$ , and whether it admits a polynomial kernel.

In contrast, FEEDBACK VERTEX SET on undirected graphs admits single-exponential FPT algorithms [11, 29], a polynomial kernel with  $4k^2$  vertices [40] (recently improved to  $2k^2$  vertices and linear time in [24]), and a 2-approximation [7].

Therefore, natural open questions are what the status of each of these three questions is for Euler digraphs.

**Problem 4.4.4** *Does FEEDBACK ARC SET on Euler digraphs allow (1) a single-exponential time FPT algorithm, (2) a polynomial kernel, and/or (3) a constant-factor approximation?*

Finally, although it does not serve to close any of the above-mentioned open questions, we note a few properties of FEEDBACK ARC SET on Euler digraphs that do not hold for the general FEEDBACK ARC SET problem. First, we note that instances with a small feedback arc set are structurally simple.

**Lemma 4.4.5** ([21]) *Every Euler digraph  $D$  has undirected cutwidth at most  $2\text{fas}(D)$ .*

**Proof:** Let  $F$  be a minimum feedback arc set of  $D$ , and let  $\sigma = (v_1 \dots v_n)$  be an acyclic ordering of  $D - F$ . Let  $i \in [n]$  and let  $V_i = \{v_1, \dots, v_i\}$ . Then for any  $i$ ,  $d^-(V_i) \leq \text{fas}(D)$  since every such arc is contained in the feedback arc set, and  $d^+(V_i) = d^-(V_i)$  since  $D$  is Euler.  $\square$

Recall that the undirected pathwidth is bounded by the undirected cutwidth; therefore the undirected pathwidth is also bounded. Clearly, no

such statement is possible for the general FAS problem, since acyclic digraphs can have arbitrary underlying undirected graphs.

Finally, we make a remark about the iterative compression approach to FPT algorithms. This is an important method in parameterized complexity, which among many other cases is used in the algorithm for DIRECTED FEEDBACK VERTEX SET [5] and in the currently fastest algorithm for (UNDIRECTED) FEEDBACK VERTEX SET [11, 29]. In this approach, a parameterized problem is solved by iteratively solving the problem on a sequence of subgraphs of the original graph, in each step using the solution from the previous step to produce a solution for the next step. Concretely, let COMPRESSION FEEDBACK ARC SET be the FEEDBACK ARC SET problem where the input additionally includes a feedback arc set of size  $k + 1$ . Assume that we have an FPT algorithm for COMPRESSION FEEDBACK ARC SET that either produces an output feedback arc set of size at most  $k$ , or concludes that no such solution exists. Then we can solve FEEDBACK ARC SET for general digraphs using  $|A|$  calls to this algorithm, as follows: Let  $D = (V, A)$  be a digraph, enumerate the arcs as  $A = \{a_1, \dots, a_m\}$ , and define  $D_i = D(\{a_1, \dots, a_i\})$  for  $i \in [m]$ . Compute a trivial solution for  $D_k$  (e.g., the entire arc set). Then for every  $k < i \leq m$ , if  $F_i$  is a solution for  $D_i$  with  $|F_i| \leq k$  then  $F_i \cup \{a_{i+1}\}$  is a solution for  $D_{i+1}$  of size at most  $k + 1$ , which can be fed into the compression algorithm. It remains only to observe that if any instance  $D_i$  is concluded to be negative, i.e., not to have a feedback vertex set of size  $k + 1$ , then the same is true for  $D$ .

The obstacle to the immediate application of this strategy in Euler digraphs is that the digraphs  $D_1, D_2, \dots$  are in general no longer Euler. However, we observe that the strategy can be adopted by the use of the splitting off operation.

**Lemma 4.4.6** FEEDBACK ARC SET *on an Euler digraph  $D = (V, A)$  with parameter  $k$  can be solved using polynomial-time processing and at most  $|A|$  calls to a solver for COMPRESSION FEEDBACK ARC SET on balanced digraphs, where the calls to the solver all use graphs with at most  $|V|$  vertices, at most  $|A|$  arcs, and parameter at most  $k + 1$ .*

**Proof:** We use the iterative compression approach, constructing a sequence of graphs  $D_i = (V, A_i)$  as follows. Begin with  $D = D_m$ , then for every  $i \in [m - 1]$  we attempt to identify a simple splitting pair  $uv, vw$  in  $A_{i+1}$ . If there is one, then we construct  $D_i$  from  $D_{i+1}$  by splitting off  $uv$  and  $vw$  in  $D_{i+1}$ , i.e.,  $A_i = (A_{i+1} \setminus \{uv, vw\}) \cup \{uw\}$ . If we cannot find such a pair, then by Proposition 4.1.3,  $D_{i+1}$  takes the form of a disjoint union of complete digraphs. At this point, the instance  $D_{i+1}$  can be solved in polynomial time, since every ordering of the vertex sets yields the same number of backward arcs. Therefore the sequence  $D_m, D_{m-1}, \dots$  can be constructed, yielding a sequence  $D_1, \dots, D_t$  of gradually larger Euler digraphs, where  $t \leq m$ , and where we can find a minimum solution for  $D_1$  in polynomial time.

It remains to show that the sequence is useful for iterative compression, i.e., that  $\text{fas}(D_i) \leq \text{fas}(D_{i+1}) \leq \text{fas}(D_i) + 1$  for every  $1 \leq i < t$ . Let  $i \in [t-1]$ , and let  $F_i$  be a feedback arc set for  $D_i$  with at most  $k$  arcs. Assume that  $D_i$  was created by splitting off the arcs  $uv, vw$  in  $D_{i+1}$ . Define  $F'_i$  by replacing  $uw$  in  $F_i$  by the pair  $uv, vw$  if  $uw \in F_i$ , otherwise let  $F'_i = F_i$ . Then clearly  $|F'_i| \leq k + 1$ , and it is easy to verify that  $F'_i$  is a feedback arc set for  $D_{i+1}$ . In the other direction, let  $F_{i+1}$  be a feedback arc set for  $D_{i+1}$ , and construct  $F_i$  as  $(F_{i+1} \setminus \{uv, vw\}) \cup \{uw\}$  if  $F_{i+1} \cap \{uv, vw\} \neq \emptyset$ , and otherwise  $F_i = F_{i+1}$ . Then again it is easy to verify that  $F_i$  is a feedback arc set for  $D_i$ ; hence if  $D_i$  does not have a feedback arc set of size  $k$  then neither does  $D_j$  for any  $j > i$  and we may reject the instance  $D$ . We conclude that we can solve FEEDBACK ARC SET for Euler (or balanced) digraphs using  $t \leq m$  calls to a solver for COMPRESSION FEEDBACK ARC SET for balanced digraphs, without increasing the number of arcs or vertices or the value of  $k$ , as required.  $\square$

Finally, we note that the currently fastest algorithm for FEEDBACK VERTEX SET on undirected graphs is actually based on an algorithm with a running time of  $O^*(3^w)$  where  $w$  is the treewidth of the graph, which yields an FPT algorithm with a running time of  $O^*(3^k)$  using iterative compression [11]. Also note that the naive state space of the treewidth dynamic programming algorithm for feedback vertex set would seem to need to enumerate either induced forests on the vertices of the bag, or at the very least all partitions of the vertices of the bag, both of which number  $2^{\Theta(w \log w)}$  for  $w$  vertices. Therefore, it seems worthwhile to ask the following question separately from the above.

**Problem 4.4.7** *Can FEEDBACK ARC SET on Euler digraphs be solved in  $O^*(2^{O(w)})$  time, where  $w$  is the width of a provided undirected tree decomposition?*

#### 4.4.2 Arc-Disjoint Cycles

Next, we consider the ARC-DISJOINT CYCLES problem in Euler digraphs. As we saw, this problem is hard on general digraphs; Gutin, Jones, Sheng and Wahlström [21] showed the problem to be FPT on Euler digraphs. The strategy is based on a win-win approach, where they show that for every Euler digraph  $D$ , either  $D$  contains at least  $k$  pairwise arc-disjoint cycles, or  $D$  has undirected pathwidth at most  $f(k)$  for some function  $f(k)$ , in which case we can solve the problem by dynamic programming. Gutin *et al.* also solve a related problem called the DIRECTED  $k$ -CHINESE POSTMAN PROBLEM, but we will focus on ARC-DISJOINT CYCLES.

The basis for the strategy is the following result of Reed, Robertson, Seymour and Thomas [35], settling a conjecture by Younger [44].

**Theorem 4.4.8** ([35]) *There is a function  $f : \mathbb{N} \rightarrow \mathbb{N}$  such that every digraph  $D$  contains either  $k$  pairwise arc-disjoint cycles or has a feedback arc set of size at most  $f(k)$ .*

The vertex-version of the result also holds, i.e., every digraph  $D$  either contains at least  $k$  pairwise vertex-disjoint cycles or has feedback vertex set number of at most  $f(k)$ , but the above version will be more useful to us. We also remark that the function  $f(k)$  grows very rapidly; according to the authors,  $f(k)$  is an iterated exponential whose height is bounded by another iterated exponential. Therefore, the resulting FPT algorithm we describe, which uses this theorem, will be a purely theoretical result, showing that the problem is FPT but without giving a running time bound that would be practically useful.

At a high level, the algorithm goes as follows. Let  $D$  be an Euler digraph and  $k$  an integer. If  $D$  contains  $k$  cycles then the instance is positive; if not, then by Theorem 4.4.8  $D$  has a feedback arc set of size at most  $f(k)$ , and by Lemma 4.4.5 it has undirected cutwidth, and thereby undirected pathwidth, at most  $2f(k)$ . Therefore, a decision algorithm could compute  $f(k)$ , or an upper bound on it; use the FEEDBACK ARC SET algorithm with parameter  $f(k)$  to check whether  $D$  has a “small” feedback arc set; and use the feedback arc set, if it exists, to compute a bounded width path decomposition of  $D$ . The path decomposition can then be used as the basis for a standard dynamic programming algorithm. If the FAS algorithm instead signals that  $\text{fas}(D) > f(k)$ , then by Theorem 4.4.8 the instance must be positive.

To turn this into a constructive algorithm, i.e., an algorithm that actually produces the cycles as an output, involves some surprising subtleties. The usual approach to this problem would be via self-reducibility: Given an algorithm that can detect the existence of an object in  $D$ , we can apply it repeatedly to subgraphs of  $D$  until we find a subgraph  $D'$  of  $D$  such that  $D'$  contains the object but no strict subgraph of  $D'$  does, at which point finding the object is hopefully trivial.

This strategy has two obstacles in the current situation. First, the decision algorithm would only apply to Euler digraphs, and an arbitrary subgraph of  $D$  would in general not be Euler. Second, even the subgraph-minimal case, when  $D$  contains  $k$  cycles but no strict subgraph of  $D$  does, is not necessarily trivial.

Gutin *et al.* [21] solved the problem by using the FPT approximation algorithm of Grohe and Grüber [18] mentioned earlier. This is an FPT algorithm parameterized by  $k$  which on an input digraph  $D$  (not necessarily Euler) either concludes that  $D$  does not contain  $k$  disjoint cycles, or returns at least  $g(k)$  disjoint cycles, for some growing, unbounded function  $g(k)$ . Combining this result with Theorem 4.4.8 and with the FEEDBACK ARC SET algorithm as above yields a constructive FPT algorithm (see the paper for details). Here, instead, we show a different approach, based on the splitting-off strategy as in Lemma 4.4.6. We begin by noting the dynamic programming result from

Gutin *et al.* [21]. (We note that this algorithm was developed for a more general problem than cycle-packing, and therefore it is possible that the running time can be improved; but we will not investigate the question of fastest running time.)

**Lemma 4.4.9** ([21]) *There is an algorithm that, given a digraph  $D$ , a vertex ordering of  $D$  of undirected cutwidth  $p$  and an integer  $k$ , finds  $k$  arc-disjoint cycles in  $D$  if they exist, and runs in time  $O^*(2^{(p+2)k})$ .*

**Theorem 4.4.10** *Let  $f^*(k)$  be the smallest value such that every Euler digraph contains either at least  $k$  arc-disjoint cycles or has a feedback arc set number at most  $f^*(k)$ . There is an FPT algorithm, parameterized by  $k$ , that on input  $(D, k)$  in time  $O^*(2^{2kf^*(k)})$  either returns  $k$  arc-disjoint cycles in  $D$  or concludes that no such solution exists. The algorithm does not need to know the value of  $f^*(k)$ .*

**Proof:** We proceed as in Lemma 4.4.6, and starting from  $D_m = D$  we compute a sequence of gradually smaller digraphs, computing  $D_i$  from  $D_{i+1}$  using a simple splitting pair  $uv, vw \in A(D_{i+1})$ . Let  $D'$  be the transitive digraph resulting at the end of this process. It is trivial to find a maximum cycle packing in  $D'$ , by Proposition 4.1.3 and since the arc set of a complete digraph on  $t$  vertices decomposes into  $\binom{t}{2}$  arc-disjoint directed cycles of length 2. Note that the number of cycles produced in this is also identical to the feedback arc set number of  $D'$ . Hence, we either find at least  $k$  cycles in  $D'$  or we can construct a feedback arc set of  $D'$  of size less than  $k$ .

Now we “unroll” the splitting-off sequence above as follows. Let the current graph be  $D_i$ , created from  $D_{i+1}$  by splitting off the arcs  $uv, vw$ . As a loop invariant, for every graph  $D_i$  we have either located  $k$  arc-disjoint cycles, or  $D_i$  contains fewer than  $k$  arc-disjoint cycles and we have computed a minimum feedback arc set, necessarily of size at most  $f^*(k)$ . If we have found  $k$  arc-disjoint cycles in  $D_i$ , then it is clear that this cycle packing can be transformed to a cycle packing in the original graph  $D$ , by repeatedly undoing the splitting-off operation. Hence, we assume that the instance  $D_i$  is negative, and let  $F_i$  be a minimum feedback arc set of  $D_i$ , where  $|F_i| \leq f^*(k)$ . As in Lemma 4.4.6, we produce a feedback arc set  $F'_i$  of  $D_{i+1}$  of size at most  $f^*(k) + 1$ , by replacing  $uw$  by  $uv, vw$  in  $F_i$  if  $uw \in F_i$ ; we then use the algorithm of Chen *et al.* [5] to compress  $F'_i$  to a minimum feedback arc set  $F_{i+1}$  of  $D_{i+1}$ . This gives us a vertex ordering of  $D_{i+1}$  of undirected cutwidth  $p \leq 2|F_{i+1}| \leq 2f^*(k) + 2$ , and we can determine whether  $D_{i+1}$  contains  $k$  arc-disjoint cycles using Lemma 4.4.9, in time  $O^*(2^{2kf^*(k)})$ . If  $D_{i+1}$  contains  $k$  arc-disjoint cycles, then we can unroll the splitting-off sequence to produce  $k$  arc-disjoint cycles in  $D$ ; otherwise,  $|F_{i+1}| \leq f^*(k)$  and the process can be repeated. □

Finally, we note that the upper bound  $f(k)$  known on the relation between the cycle-packing number and the feedback arc set number is potentially very



far from being tight. In fact, even for general digraphs the best reported lower bound is  $\Omega(k \log k)$  due to Alon, as reported by Reed *et al.* [35]. However, as we have seen, the structure of cycles in Euler digraphs is much simpler than in general digraphs. It seems worthwhile to pursue a tighter bound for this special class.

**Problem 4.4.11** *Can the bound on  $f(k)$  be improved for Euler digraphs, possibly to polynomial or even  $O(k \log k)$ ?*

Finally, let us consider the existence of polynomial kernels for ARC-DISJOINT CYCLES on Euler digraphs. It is tempting to once again look at the undirected version of the problem, i.e., EDGE-DISJOINT CYCLES, to heuristically indicate whether a kernel is likely. In this case, we find that EDGE-DISJOINT CYCLES does have a polynomial kernel [3], and furthermore, by the classical Erdős–Pósa result, the corresponding function  $f(k)$  in undirected graphs has  $f(k) = O(k \log k)$  (both in the edge- and vertex-versions) [12]. However, fundamentally these results rely upon statements about short girth in undirected graphs (e.g., a graph of minimum degree 3 has girth  $O(\log n)$ ), which does not transfer to the Euler digraph case.

**Problem 4.4.12** *Does ARC-DISJOINT CYCLES have a polynomial kernel on Euler digraphs?*

#### 4.4.3 Additional Topics

Finally, we review a few additional topics regarding cycle-packings in Euler digraphs.

Questions of arc-disjoint cycle-packing have been considered in extremal graph theory. Alon, McDiarmid and Molloy [1] showed that every  $k$ -regular digraph contains a packing of  $\Omega(k^2)$  arc-disjoint cycles, and conjectured that this can be sharpened to  $\binom{k+1}{2}$  arc-disjoint cycles. They also give a construction showing that this result would be tight. Let  $C_n^k$ ,  $n \geq 2k + 1$ , be the digraph with vertex set  $\{0, \dots, n-1\}$  and all arcs  $uv$  where  $v = u+i \pmod{n}$ ,  $i \in [k]$ . Then  $\text{fas}(C_n^k) = \binom{k+1}{2}$ , witnessed by the vertex ordering  $0, \dots, n-1$ , and  $C_n^k$  contains  $\binom{k+1}{2}$  arc-disjoint cycles, since there are  $k$  arc-disjoint cycles through the vertex  $n-1$ , whose removal leave a graph isomorphic to  $C_{n-1}^{k-1}$ .

Brualdi and Shen [4] gave the following further conjectures.

**Conjecture 4.4.13** *Let  $k \geq 2$  be an integer. Every bipartite Euler digraph with partition sizes  $m$  and  $n$  and at least  $mn/(k+1)$  arcs contains a cycle of length at most  $2k$ .*

**Conjecture 4.4.14** *Every bipartite Euler digraph  $D = (V, A)$  with partition sizes  $m$  and  $n$  contains a collection of  $|A|^2/(4mn)$  arc-disjoint cycles.*

In particular, the latter conjecture specializes into the conjecture that every Euler bipartite tournament decomposes into arc-disjoint 4-cycles.

Another question is the following.

**Problem 4.4.15** *For which Euler digraphs is the cycle-packing number equal to the feedback arc set number?*

Let us say that a digraph  $D$  **packs** if  $D$  contains  $\text{fas}(D)$  arc-disjoint cycles. To what extent can we characterize Euler digraphs which pack? One important result here is by Seymour [38], who showed that it holds for Euler digraphs which can be linklessly embedded in 3-space. This can be viewed as an “Euler generalization” of the result that all planar digraphs pack, which follows from the Lucchesi–Younger theorem [31, 38].

This result also carries over to the weighted version, as follows. Let  $D = (V, A)$  be a digraph (not necessarily Euler) that can be linklessly embedded in 3-space, and let  $w : A \rightarrow \mathbb{Z}_+$  be a balanced set of arc-weights, i.e., for every vertex  $v \in V$ ,  $\sum_{uv \in A} w(uv) = \sum_{vw \in A} w(vw)$ . Then the arc-disjoint cycle-packing number of  $D$ , with arc capacities  $w$ , and the weighted feedback arc set number, with arc costs  $w$ , are the same.

But for the more general question, presumably asking which individual digraphs  $D$  pack is too ambitious to expect a good answer. For general digraphs, a natural option is to restrict attention to a hereditary class of digraphs. For the non-Euler case, Guenin and Thomas [19] characterized the class of digraphs  $D$  such that for every subdigraph  $H$  of  $D$ , the feedback vertex set number and the vertex-disjoint cycle-packing number of  $H$  agree. The characterization is in terms of a list of forbidden butterfly minors for the class. Naturally, if the line graph of  $D$  belongs to this class, then  $D$  and every subdigraph of  $D$  pack in the above sense. However, this does not take into account the restriction that we are only concerned with Euler digraphs. For example, consider a digraph  $D$  on six vertices  $a_i, b_i, i \in [3]$ , with arcs  $a_i b_i$  and  $b_i a_j, i \neq j, i, j \in [3]$ . It is easily verified that this graph does not pack; it has arc-disjoint cycle-packing number 1 but feedback arc set number 2. On the other hand, consider the graph  $D'$  which instead contains two copies of the arcs  $a_i b_i, i \in [3]$ . Then  $D'$  is an Euler digraph, which decomposes into three arc-disjoint 4-cycles, and with feedback arc set number three (for example, the two arcs  $a_1 b_1$  and the arc  $b_2 a_3$ ). Thus  $D'$  packs, and it can be verified that every Euler subdigraph of  $D'$  also packs. Thus in particular, a question one could ask is, what is the class of Euler digraphs  $D$  such that every Euler subdigraph of  $D$  packs? Regarding the nature of the characterization, consider the following. Let  $H$  be an Euler digraph that does not pack. If  $D$  is a digraph which has  $H$  as a topological minor, then it follows that  $D$  has an Euler subdigraph that does not pack. The same does not appear to be true for butterfly minors (although we have no counterexample, the “minor model” of a butterfly minor is usually a non-Euler subdigraph, even when the minor itself is Euler).

Yet further options include considering richer operations than subgraphs, e.g., considering graph classes closed under splitting-off operations (and removal of loops, i.e., taking Euler subgraphs). This may be easier to answer, on account of having a more powerful containment operation.

We refrain from explicitly singling out one of these questions as a main open problem, as it is not clear to us whether any one of them will be more productive or feasible than the others.

## 4.5 Linkages and Cut Problems

We now move on to variants of linkages and multiflow-type problems. The results in this section are of mainly two variants. First, in Sections 4.5.1 and 4.5.2 we consider path-packing and unsplittable multi-commodity flow type problems in the style of WEAK  $k$ -LINKAGE, where the exact endpoints of the paths we are asked to pack are specified. We will in particular show that TWO-COMMODITY FLOW admits a polynomial-time algorithm, and recall the long-open question of whether WEAK  $k$ -LINKAGE is FPT parameterized by  $k$ .

Then, in Sections 4.5.3 and 4.5.4 we consider an alternative setup, where we are asked to find a maximum path-packing on a set of terminals according to some condition, but it is not specified exactly how many paths of each type we are required to pack. A main result here is the classical result that the so-called FREE MULTIFLOW problem is in  $P$  on Euler digraphs. We observe (as Frank did in the 1980s [16]) that, remarkably, this implies that ARC MULTIWAY CUT is in  $\mathcal{P}$  on Euler digraphs, a result that does not carry over even to undirected Euler graphs. We also review some weighted generalizations of this result.

### 4.5.1 Two-Commodity Flow

In this section, we consider problems related to TWO-COMMODITY FLOW and MULTI-COMMODITY FLOW.

We will use the following formulation. Let  $D = (V, A)$  and  $H = (V, F)$  be digraphs (not necessarily Euler), where  $D$  is referred to as the **supply graph** and  $H$  as the **demand graph**. The goal is to find a set of arc-disjoint paths in  $D$  meeting the demand of  $H$ , or equivalently, find a collection of pairwise arc-disjoint cycles in  $D + H$  such that every cycle uses exactly one arc from  $F$  and all arcs of  $F$  are covered by the cycles. We refer to the non-isolated vertices of  $H$  as the **terminals**. Slightly abusing notation, we refer to this as the WEAK  $k$ -LINKAGE problem, although with input represented as a pair of directed multigraphs  $(D, H)$  as above rather than in the equivalent alternative representation previously defined in Section 1.6.

In general digraphs, this problem is  $\mathcal{NP}$ -hard even when  $H$  consists of just two arcs. In fact, for general digraphs, the only polynomial-time solvable

cases of the problem reduce to standard  $s$ - $t$  cuts via Menger’s theorem; see the end of this subsection.

In this section, we show that when the graph  $D + H$  is Euler, then, effectively, we can handle the case where  $H$  consists of two pairs of terminals. The same result also extends to the arc-capacitated two-commodity flow problem, in the case when the capacities of  $D + H$  are balanced; see later.

These results are due to Frank [16], who extended similar results from undirected Euler graphs to the directed case. The proof below is essentially from Frank.

The central result is that given a pair of digraphs  $(D, H)$  on vertex set  $V$ , such that  $D + H$  is Euler and  $H$  is acyclic and consists of two pairs of parallel arcs, the instance  $(D, H)$  of WEAK  $k$ -LINKAGE is positive if and only if it satisfies the **directed cut criterion**:

$$d_D^+(X) \geq d_H^-(X) \text{ for all } X \subseteq V. \tag{4.1}$$

Clearly, this is a necessary condition. We show that it is also sufficient. We say that a set  $X \subseteq V$  is a **tight set** if equality holds for  $X$  in the statement above, i.e.,  $d_D^+(X) = d_H^-(X)$ . For the duration of this proof, for any digraph  $D$  we will define

$$d_D^*(A, B) = |(A \setminus B, B \setminus A)_D| + |(B \setminus A, A \setminus B)_D|,$$

i.e.,  $d_D^*(A, B)$  counts the number of arcs with one end in  $A \setminus B$  and one in  $B \setminus A$ , regardless of orientation. We begin with some statements about tight sets.

**Lemma 4.5.1** *Let  $(D = (V, A), H = (V, F))$  be an instance of WEAK  $k$ -LINKAGE such that  $D + H$  is Euler and the directed cut criterion (4.1) holds for  $(D, H)$ . Then the following hold.*

1. *If  $X \subseteq V$  is a tight set, then so is  $V \setminus X$ .*
2. *If  $X, Y \subseteq V$  are tight sets, then  $d_H^*(X, Y) \geq d_D^*(X, Y)$  and if equality holds then  $X \cap Y$  and  $X \cup Y$  are both tight sets.*
3. *If  $X, Y \subseteq V$  are tight sets, then  $d_H^*(X, V \setminus Y) \geq d_D^*(X, V \setminus Y)$  and if equality holds then  $X \setminus Y$  and  $Y \setminus X$  are both tight sets.*

**Proof:** 1. Since  $D + H$  is Euler, we have

$$d_D^+(X) + d_H^+(X) = d_D^-(X) + d_H^-(X)$$

for all  $X \subseteq V$ , from which the statement follows.

2. Recall that for any digraph and any vertex set  $X$ , we have

$$d^+(X) + d^+(Y) = d^+(X \cup Y) + d^+(X \cap Y) + d^*(X, Y),$$

and similarly for  $d^-(X)$  and  $d^-(Y)$ . Therefore, rearranging terms we get

$$\begin{aligned}
 & (d_D^+(X) - d_H^-(X)) + (d_D^+(Y) - d_H^-(Y)) = \\
 & (d_D^+(X \cap Y) + d_D^+(X \cup Y) + d_D^*(X, Y)) - \\
 & \quad - (d_H^-(X \cap Y) + d_H^-(X \cup Y) + d_H^*(X, Y)) = \\
 & (d_D^+(X \cap Y) - d_H^-(X \cap Y)) + (d_D^+(X \cup Y) - d_H^-(X \cup Y)) + \\
 & \quad + d_D^*(X, Y) - d_H^*(X, Y) = 0,
 \end{aligned}$$

where the whole expression equals 0 since  $X$  and  $Y$  are tight. Rewriting the last line, we have

$$(d_D^+(X \cap Y) - d_H^-(X \cap Y)) + (d_D^+(X \cup Y) - d_H^-(X \cup Y)) = d_H^*(X, Y) - d_D^*(X, Y),$$

where the left-hand side is non-negative.

3. This statement follows by combining the two previous ones. □

**Theorem 4.5.2** *Let  $(D = (V, A), H = (V, F))$  be an instance of WEAK  $k$ -LINKAGE such that  $D + H$  is Euler, and  $H$  is acyclic and consists of two sets of parallel arcs. Then  $(D, H)$  is a ‘Yes’-instance if and only if it satisfies the directed cut criterion (4.1).*

**Proof:** Let  $T = \{s_1, t_1, s_2, t_2\}$  be the terminals of  $H$ , and assume that  $F$  consists of  $k_1 > 0$  arcs  $t_1s_1$  and  $k_2 > 0$  arcs  $t_2s_2$ . Hence the task is equivalent to packing  $k_1 + k_2$  arc-disjoint paths in  $D$  so that  $k_1$  of these are from  $s_1$  to  $t_1$  and  $k_2$  are from  $s_2$  to  $t_2$ . We may assume that  $D$  has no isolated vertices in  $V \setminus T$  and every connected component of  $D + H$  intersects  $T$ . Also observe that the result follows from Menger’s theorem if  $T$  intersects more than one connected component of  $D + H$ ; hence we assume that  $D + H$  is connected.

Assume for a contradiction that the theorem is false, and let  $(D, H)$  be a minimum counterexample with respect to  $|A(D + H)|$ . Since it is clear that the directed cut criterion is a necessary condition, this implies that  $(D, H)$  is a negative instance that meets the directed cut criterion, and that the theorem holds for every instance  $(D', H')$  where  $|A(D' + H')| < |A(D + H)|$ .

In particular, consider a pair of arcs  $xy, yz \in A(D)$  such that  $|\{x, y, z\}| = 3$ , i.e., when  $x, y, z$  are distinct. Let  $D'$  be the result of splitting of  $xy, yz$  in  $D$ . Then  $D' + H$  is Euler, and either  $(D', H)$  fails to meet the directed cut criterion, or the instance  $(D', H)$  is positive. But in the latter case, the paths packed in  $D'$  also exist in  $D$ , by expanding one arc  $xz$  (if used) into the two arcs  $xy, yz$ ; hence we conclude that for every such pair of arcs  $xy, yz \in A(D)$ , splitting off  $xy$  and  $yz$  will break the directed cut criterion. Now note first that this is possible if and only if there is a tight set  $X$  with either  $X \cap \{x, y, z\} = \{y\}$  or  $X \cap \{x, y, z\} = \{x, z\}$ , and second, by Lemma 4.5.1 in fact both these tight sets would exist. Furthermore, since  $d_D^+(X), d_D^+(V \setminus X) > 0$  (as evidenced by the arcs  $xy, yz$ ), we also have  $d_H^-(X), d_H^-(V \setminus X) > 0$ . Therefore, we find that for every set  $X$  such that splitting off a pair of arcs breaks the directed cut criterion at  $X$ , we have  $X \cap T = \{s_i, t_j\}, i \neq j$ .

We proceed with the proof. First consider the case when  $V = T$ . If  $D$  contains an arc  $s_i t_i$ ,  $i \in \{1, 2\}$ , then we can remove that arc together with a copy of  $t_i s_i$  from  $H$ , to produce a smaller instance  $(D', H')$ . It is easy to see that the directed cut condition holds in  $(D', H')$ , therefore there is a path-packing in  $D'$ , which can be extended by the additional arc  $s_i t_i$  to a solution for  $(D, H)$ , and we are done. We claim that we can find a pair of arcs  $su, uv$  in  $D$  where  $s \in \{s_1, s_2\}$  and  $|\{s, u, v\}| = 3$ . Indeed, by the cut criterion we have  $d_D^+(s_i) \geq k_i$  for  $i = 1, 2$ , and  $|(\{s_1, s_2\}, \{t_1, t_2\})_D| \geq k_1 + k_2$ . The only case where you cannot find the pair  $su, uv$  is if  $A$  consists entirely of arcs  $s_1 t_2$  and  $s_2 t_1$ , but in such a case we have  $d_D^+(\{s_1, t_2\}) = 0 < d_H^-(\{s_1, t_2\}) = k_1$ . Hence  $su, uv$  exist, and as above, splitting off the pair will break the cut criterion, showing that there are tight sets  $X, V \setminus X$  such that  $X \cap \{s, u, v\} = \{u\}$  and  $X \cap T = \{s_i, t_j\}$ ,  $i \neq j$ . Without loss of generality we may assume that  $s = s_1$ , so that  $X = \{s_2, t_1\}$  and  $\{s, v\} = \{s_1, t_2\}$ . But then there is no way to select the arcs  $su, uv$  without using an arc  $s_i t_i$ ,  $i \in \{1, 2\}$ . Hence if there is a counterexample, it has  $T \subset V$ .

Now, let  $u \in V \setminus T$  such that there is an arc  $ux \in A$  for some  $x \in T$ ; clearly such a vertex exists, and there is also at least one arc  $vu \in A$ ,  $v \neq x$ . As before, we conclude that there must exist tight sets  $X, V \setminus X$  such that  $X \cap T = \{s_i, t_j\}$ ,  $\{i, j\} = \{1, 2\}$ , and  $\{v, u, x\} \cap X = \{u\}$ . Also note that since  $x \in T$  in fact  $X \cap T$  is fixed by the condition that  $x \notin X$ . For every arc  $vu \in A$ , let  $X_v$  be a tight set proving that we cannot split off  $vu$  and  $ux$ . Then for every pair of such sets  $X_a$  and  $X_b$ , since  $X_a \cap T = X_b \cap T$  we have  $d_H^*(X_a, X_b) = 0$ , so by Lemma 4.5.1 both  $X_a \cup X_b$  and  $X_a \cap X_b$  are tight. Let  $X$  be the intersection of all such sets. We claim that there must be an arc  $v'u \in A$  such that  $v' \in X$ ,  $v' \neq u$ . Assume not, and consider the set  $X' = X \setminus \{u\}$ . Then this set has  $d_H^-(X') = d_H^-(X)$  and  $d_D^+(X') < d_D^+(X)$ , contrary to the directed cut criterion. But now splitting off  $v'u$  and  $us$  cannot break the directed cut criterion, since  $X \subseteq X_{v'}$  and  $v' \in X$ . This gives us a path packing in  $(D', H)$  which can easily be converted to a path packing in  $(D, H)$  of the same size.  $\square$

As Frank observes, this proof suggests an algorithm for finding such a path-packing: if the directed cut criterion does not apply, reject the instance. If there is an arc  $s_i t_i$  in  $D$  and  $t_i s_i$  in  $H$ , then remove this pair of arcs and continue. Otherwise, find a pair of arcs  $uv, vw$  that can be split off without breaking the directed cut criterion and recursively find a path packing in the resulting instance  $(D', H)$ ; then finally, if the new arc  $uw$  is used in one of the paths in  $D'$ , replace it by the old arcs  $uv, vw$ . This process will run in polynomial time, assuming the ability to test the directed cut criterion.

**Lemma 4.5.3** *The directed cut criterion for the instances  $H$  considered here can be tested with three max-flow computations.*

**Proof:** Let  $X \subseteq V$ , and consider the cases for  $d_H^-(X)$ . If  $d_H^-(X) = 0$ , then the cut criterion holds for  $X$ . If  $d_H^-(X) = k_i$  but  $d_D^+(X) < k_i$ , then  $X$  represents

an arc-cut of size less than  $k_i$  from  $s_i$  to  $t_i$  for  $i \in \{1, 2\}$ , which can be tested via max-flow computations. Finally, if  $d_{\overline{H}}(X) = k_1 + k_2$ , then  $X$  represents an arc-cut of size less than  $k_1 + k_2$  from  $\{s_1, s_2\}$  to  $\{t_1, t_2\}$ , which can be tested with a max-flow computation by adding a new meta-source  $s$  and a meta-sink  $t$ .  $\square$

Finally, Frank shows a strongly polynomial time solution for the weighted version of the problem, TWO-COMMODITY FLOW. In this problem, the input is a digraph  $D = (V, A)$  with an **arc capacity** function  $c : A \rightarrow \mathbb{Z}_+$  as well as ordered request pairs  $(s_i, t_i)$  with demand values  $k_i$ ,  $i \in \{1, 2\}$ , with the condition that at every vertex, the sums of incoming and outgoing capacities and demands are equal; equivalently, replacing each arc  $a$  by  $c(a)$  parallel copies and adding  $k_i$  copies of the arc  $t_i s_i$  defines an Euler digraph. We refer to this as a **balanced** two-commodity flow instance. Note that  $D + H$  itself does not need to be Euler. The algorithm uses the same strategy as above, using a weighted version of the splitting operation, with some additional work required to prove that the number of steps is bounded by a polynomial. The proof, as above, is based on reasoning about the structure of tight sets, and shows that any sequence of weighted splitting operations has polynomially bounded length.

**Theorem 4.5.4** ([16]) *The TWO-COMMODITY FLOW problem can be solved in strongly polynomial time for balanced instances.*

We will consider the more general WEAK  $k$ -LINKAGE question later in this section, but for now we wrap up by recalling a characterization by Frank of when the directed cut criterion is a necessary and sufficient condition.

Let a **star** be a directed multigraph where there is either a common tail  $s$  to all arcs, or a common head  $t$  to all arcs. It is not hard to see that  $(D, H)$ -PATH PACKING can be solved via a max-flow computation if  $H$  is a star, hence the problem is in  $\mathcal{P}$ . Fortune, Hopcroft and Wyllie [15] showed that for general digraphs, the converse is true – for any fixed digraph  $H$  which is not a star,  $(D, H)$ -PATH PACKING is  $\mathcal{NP}$ -hard. For Euler digraphs, Frank showed the following.

**Theorem 4.5.5** ([16]) *Say that the directed cut criterion solves  $H$  for a directed multigraph  $H$  if, for every digraph  $D$  such that  $D + H$  is Euler, the instance  $(D, H)$  of WEAK  $k$ -LINKAGE is positive if and only if the directed cut criterion is met. Then for any  $H$ , the directed cut criterion solves  $H$  if and only if  $H$  is the (arc-disjoint) union of two stars.*

**Proof:** Assume first that  $H$  is the union of two stars. We convert  $(D, H)$  into an equivalent instance of two-commodity flow. For each star with a common tail  $t_i$  to the arcs, introduce a new vertex  $s_i$ , and replace every arc  $t_i v$  in  $H$  by an arc  $t_i s_i$  in  $H$  and an arc  $s_i v$  in  $D$ . For a star with a common head  $s_i$ , instead introduce a new vertex  $t_i$  in the same way. This reduces to the case

where  $H$  consists of  $k_i$  arcs  $t_i s_i$  for  $i = 1, 2$ . If the resulting graph  $H$  is not acyclic, we can additionally introduce vertices  $s'_i, t'_i$  with  $k_i$  arcs  $s'_i s_i$  and  $t_i t'_i$  in  $D$ , and replace the arcs  $t_i s_i$  in  $H$  by  $t'_i s'_i$ . It is clear that this preserves the property of  $D + H$  being Euler and the existence of a solution, and therefore the resulting instance can be solved by the algorithm above.

In the other direction, it is easy to observe that if  $H$  is solved by the directed cut criterion, then so is every subgraph of  $H$ . Hence, it is sufficient to identify a constant number of digraphs  $H$  which will occur as a subgraph in every graph  $H$  that cannot be decomposed into two stars, and for each such  $H$  show an instance where the directed cut criterion is insufficient. Such a list of instances is provided by Frank [16].  $\square$

### 4.5.2 General Arc-Disjoint Paths Problems

We now consider the more general question of when WEAK  $k$ -LINKAGE is tractable for Euler digraphs. This question has several variants. One may consider the case when  $D + H$  is Euler, or when  $D$  is already Euler (or, indeed when both of  $D$  and  $H$  are Euler separately); one may consider either the basic WEAK  $k$ -LINKAGE problem or the weighted multi-commodity flow variant (where arcs of  $D$  and  $H$  have capacities, respectively demands); and one may consider  $H$  to be one-time fixed or provided as problem input. There have also been several investigations into the complexity of the problem under various restrictions, including Ibaraki and Poljak [23], Vygen [43], Naves and Sebö [33], and Frank [17].

For the negative cases, we begin by noting that the three-commodity flow problem is easily seen to be  $\mathcal{NP}$ -hard when  $D$  is Euler but  $H$  is not. Let  $(D, H)$  be the input of WEAK  $k$ -LINKAGE where  $H$  consists of two arcs; as noted, this is an  $\mathcal{NP}$ -hard problem on general digraphs. Let  $D'$  be the Euler two-vertex extension of  $D$ , with new vertices  $x$  and  $y$ , and create a demand graph  $H'$  from  $H$  by adding the vertices  $x$  and  $y$  as well as  $\mu_{D'}(x, y)$  copies of the arc  $yx$ . Then clearly,  $(D, H)$  is positive if and only if  $(D', H')$  is positive, and  $D'$  is Euler (although  $D' + H'$  is not). To reach a situation where  $D' + H'$  is Euler, we can use a slight variation of this (used by Ibaraki and Poljak [23]).

**Lemma 4.5.6** *WEAK  $k$ -LINKAGE is  $\mathcal{NP}$ -hard when  $D + H$  is Euler and the underlying digraph of  $H$  (where all arc multiplicities are reduced to 1) has three arcs.*

**Proof:** Recall that WEAK 2-LINKAGE is  $\mathcal{NP}$ -hard even when  $A(H) = \{st, ts\}$ . We show a reduction from this problem to an instance  $(D', H')$  of WEAK  $k$ -LINKAGE where  $D' + H'$  is Euler. Let  $(D, H)$  with  $H$  as above be an instance of WEAK 2-LINKAGE, and let  $D'$  be the Euler two-vertex extension of  $D + H$ , adding vertices  $x$  and  $y$ . Now let  $H'$  be  $H$  plus all copies of the arc  $xy$  in  $D'$ , and remove the arcs  $xy$  from  $D'$ . We claim that  $(D', H')$  is



a positive instance of WEAK  $k$ -LINKAGE if and only if  $(D, H)$  is a positive instance of WEAK 2-LINKAGE.

First, assume that  $(D', H')$  is positive, and let  $\mathcal{P}$  be the corresponding collection of paths. Then  $\mathcal{P}$  contains an  $st$ -path  $P_1$  and a  $ts$ -path  $P_2$ , both of which are contained in  $D'$ . But since  $x$  and  $y$  are a sink, respectively a source, in  $D'$ , both  $P_1$  and  $P_2$  must exist in  $D$  as well.

In the other direction, assume that  $D$  contains an arc-disjoint pair of an  $st$ -path  $P_1$  and a  $ts$ -path  $P_2$ . Then  $P_1 + P_2$  form an Euler digraph, and therefore  $D'' := (D' + H') - (P_1 + P_2 + st + ts)$  is balanced. It follows that there are  $\mu_{D''}(x, y)$  arc-disjoint cycles through  $y$  in  $D''$ , and each of these cycles must use an arc  $xy$  from  $H'$ . Removing the arc  $xy$  from each of these cycles yields a collection of  $yx$ -paths in  $D''$  that together with  $P_1$  and  $P_2$  forms an arc-disjoint path-packing in  $D'$ .  $\square$

Hence, in particular, the results on TWO-COMMODITY FLOW cannot be extended to three or more commodities. Vygen [43] shows that the problem is still  $\mathcal{NP}$ -hard if  $D$  is additionally assumed to be acyclic.

For the case when  $H$  is fixed, say  $|A(H)| = k$ , the complexity of the problem is notoriously open. The case when  $k = 3$  was solved by Ibaraki and Poljak [23]. Specifically, they take the following approach. Let  $H$  be an Euler digraph with disjoint arcs  $t_i s_i$ ,  $i = 1, 2, 3$ . Then we can reduce the instance  $(D, H)$  of WEAK 3-LINKAGE to an instance where  $H = C_3$  as follows. Add three terminals  $x, y, z$  to  $D$ , and arcs  $x s_1, t_1 y, y s_2, t_2 z, z s_3, t_3 x$ . Then the original instance has a weak 3-linkage if and only if the resulting graph has arc-disjoint  $xy$ -,  $yz$ - and  $zx$ -paths. We then find that the problem has a solution if there is an Euler trail of  $D$  that, starting from  $x$ , visits the new terminals in the order  $x, y, z$  (i.e., a trail such that the first visit to  $y$  after the start at  $x$  comes before the last visit to  $z$ ). They are then able to solve the problem in polynomial time by carefully investigating the structure of minimal negative instances. Thus WEAK 3-LINKAGE is in  $\mathcal{P}$  if  $D + H$  is Euler.

For general  $H$ , as far as we know, the possibilities range from the problem being  $\mathcal{NP}$ -complete for  $k = 4$  to the problem being FPT parameterized by  $k$ .

**Problem 4.5.7** *What is the status of WEAK  $k$ -LINKAGE for inputs  $(D, H)$  where  $D + H$  is Euler, parameterized by  $k$ ? Is it FPT or in XP?*

A slight variant was considered by Frank, Ibaraki and Nagamochi [17]. They consider the problem variant where the input is an Euler digraph  $D$  and an undirected graph  $H$ , say  $E(H) = \{ab, cd\}$ , and the task is to find a pair of arc-disjoint paths  $P_1, P_2$  in  $D$  where  $P_1$  is either an  $ab$ -path or a  $ba$ -path, and  $P_2$  either a  $cd$ -path or a  $dc$ -path. They show that this problem can be solved in polynomial time, via an extensive investigation into the structure of minimal infeasible instances.

They note that this problem generalizes the result of Ibaraki and Nagamochi, as follows. Let  $(D, H)$  be an input to WEAK 3-LINKAGE where  $D + H$  is Euler and  $A(H) = \{t_i s_i \mid i \in [3]\}$ . Create a graph  $D'$  from  $D$  by adding four new vertices  $a, b, c, d$  and arcs  $t_1 c, c s_2, t_2 d, d a, a s_3, t_3 b, b s_1$ . Observe that  $D'$  is Euler. Now we may observe the following. Any  $ba$ -path in  $D'$  will exhaust all arcs incident with  $d$  in  $D'$ , and similarly a  $dc$ -path will exhaust  $a$ . Thus if the instance is positive, then  $P_1$  is an  $ab$ -path and  $P_2$  is a  $cd$ -path. Then  $P_2 + da + P_1$  form a directed  $cb$ -path, hence since  $D'$  is Euler,  $D' - (P_2 + da + P_1)$  contains a directed  $bc$ -path. It is clear that these paths together must form a weak 3-linkage for  $D$ .

We end with a different question, again via Frank [16].

**Problem 4.5.8** *Is WEAK  $k$ -LINKAGE with input  $(D, H)$  in  $\mathcal{P}$  if  $D + H$  is Euler and planar?*

The undirected version of this question is known to hold, i.e., EDGE-DISJOINT PACKING for undirected graphs, with input  $(G, H)$ , is in  $\mathcal{P}$  if  $G + H$  is planar. In fact, for this version, the problem is solved by the undirected version of the cut criterion (4.1). (The corresponding statement does not hold for the directed version [16].)

### 4.5.3 Free Multiflow and Arc Multiway Cut

We now turn to a slightly different model of path-packing problems, and in the process we will cover a less well known, but very interesting result due to Frank [16] on a polynomial-time solvable multicut problem on Euler digraphs.

The general setup here is as follows. We have a digraph  $D = (V, A)$  and a set of terminals  $T \subseteq V$ , but instead of having an *exact* set of path requests (encoded as a digraph  $H$  over  $T$ , as in the previous section), we have a notion of *allowed* or *disallowed* terminal-terminal paths, and we are looking for a maximum arc-disjoint path-packing that consists entirely of “allowed” paths. Somewhat more generally, we can also introduce *weights* for paths, depending on their type, and ask for an arc-disjoint path-packing of maximum weight.

Let us begin with the FREE MULTIFLOW problem, where every simple terminal-terminal path is “allowed” under the above notion. More concretely, the input to FREE MULTIFLOW is a digraph  $D = (V, A)$  and a set of terminals  $T \subseteq V$ , and the task is to find a maximum arc-disjoint packing of directed paths in  $D$  where each path goes between distinct terminals in  $T$ . Frank showed that if  $D$  is Euler, then the problem has a simple min-max formula, as follows. For disjoint sets  $X, Y \subset V(D)$  in a digraph  $D$ , let  $\lambda_D(X, Y)$  denote the maximum number of arc-disjoint paths from  $X$  to  $Y$  in  $D$ . We will slightly abuse notation and use single vertices in place of singleton sets (e.g., we write  $t$  rather than  $\{t\}$ ). Then the maximum number of paths in the free multiflow packing equals

$$\sum_{t \in T} \lambda_D(t, T - t),$$

the sum of the sizes of isolating cuts in  $D$ .

For general digraphs, the FREE MULTIFLOW problem is  $\mathcal{NP}$ -hard for  $|T| \geq 2$  as it generalizes the WEAK 2-LINKAGE problem, but for undirected graphs there are several classical polynomial-time results in this direction. The most immediately corresponding result is due to Lovász [30], who proved the corresponding statement for undirected Euler graphs, and on which Frank's result is based. However, multiple more general results exist, including Mader's theorem on packing internally vertex-disjoint terminal-terminal paths in general undirected graphs (see Schrijver [37]), as well as generalizations in terms of packing paths in group-labelled graphs (see Chudnovsky, Geelen, Gerards, Goddyn, Lohman, and Seymour [8]) and in general permutation-labelled graphs (see Pap [34]).

However, there is one unique feature of the Euler digraph result that is not mirrored in any of the variants of the problem on undirected graphs (even Euler graphs). For each of the above packing problems, one can define a natural dual **cut problem** (or, alternatively expressed, a **path-hitting problem**) of finding a minimum set  $X$  of edges (or arcs, or vertices) such that removing every element of  $X$  leaves a graph with no allowed paths of the respective type. Concretely, the dual to the FREE MULTIFLOW problem would be the classical problem ARC MULTIWAY CUT, of finding a minimum set  $X$  of arcs in a digraph  $D$  such that  $D - X$  contains no terminal-terminal path. Whereas the undirected version, EDGE MULTIWAY CUT, is  $\mathcal{NP}$ -hard on undirected Euler graphs for 3 terminals, the min-max theorem for Euler digraphs directly implies that for this graph class, ARC MULTIWAY CUT is in  $\mathcal{P}$ .

For the rest of this section, let  $D = (V, A)$  be an Euler digraph and  $T \subseteq V$  a set of terminals. Let  $T = \{t_1, \dots, t_p\}$  for  $p = |T|$ , and recall that for each  $i \in [p]$ ,  $d_i = \lambda_D(t_i, T - t_i)$  denotes the maximum number of arc-disjoint paths from  $t_i$  to the remaining terminals.

We will prove the following theorem, from which an algorithm for ARC MULTIWAY CUT will follow easily.

**Theorem 4.5.9** *Let  $D = (V, A)$  be an Euler digraph and  $T \subseteq V$  a set of terminals. There is an arc-disjoint packing  $\mathcal{P}$  of terminal-terminal paths in  $D$  such that each terminal  $t_i \in T$  is the starting vertex of  $d_i$  paths in  $\mathcal{P}$ .*

Theorem 4.5.9 will in turn follow from the following statement.

**Theorem 4.5.10** *Let  $vw \in A$  be an arbitrary arc where  $v \notin T$ . Then there is an arc  $uv \in A$  such that splitting off  $uv$  and  $vw$  will not change the value of  $d_i$  for any  $i \in [p]$ .*

**Proof:** Recall that  $d_H^+(S)$  denotes the number of arcs out of  $S$  in a digraph  $H$ .

Let  $uv, vw \in A$  be an arbitrary pair of arcs, and assume that splitting them off in  $D$  decreases the value of  $d_i$  for some  $i \in [p]$ . Let  $D'$  be the result of the splitting-off operation. Then there exists a set  $V_i \subseteq V$  such that  $V_i \cap T = \{t_i\}$ , and  $d_{D'}^+(V_i) < d_i$  in  $D'$ , which is true if and only if  $d_D^+(V_i) = d_i$  and  $V_i \cap \{u, v, w\} \in \{\{v\}, \{u, w\}\}$ . We say that the set  $V_i$  *blocks* the splitting off  $uv, vw$  in  $D$ . Therefore, similarly to the concept of tight sets in the previous section, let us refer to a set  $V_i \subseteq V$  as *i-critical* if  $V_i \cap T = \{t_i\}$  and  $d_D^+(V_i) = d_i$ . A set  $U \subseteq V$  is *critical* if it is *i-critical* for some  $i \in [p]$ . We make a few observations about critical sets.

*Observation 1:* For every  $i \in [p]$ , the *i-critical* sets are closed under union and intersection, and for two *i-critical* sets  $V_i, V'_i$ , there is no arc between  $V_i \setminus V'_i$  and  $V'_i \setminus V_i$ . This follows from the well-known submodularity of cuts that for any  $X, Y \subseteq V$  we have

$$d_D^+(X) + d_D^+(Y) \geq d_D^+(X \cap Y) + d_D^+(X \cup Y),$$

with equality only if there are no arcs between their symmetric differences.

*Observation 2:* If  $V_i$  is *i-critical* and  $V_j$  is *j-critical*,  $i \neq j$ , then  $V_i \setminus V_j$  is *i-critical* and  $V_j \setminus V_i$  is *j-critical*, and there is no arc between  $V_i \cap V_j$  and  $V \setminus (V_i \cup V_j)$ . This follows similarly as above, with the additional ingredient that for an Euler digraph  $D$ , we have  $d_D^+(S) = d_D^+(V \setminus S)$ .

The proof now proceeds in two cases. In the first case, assume that there is a critical set  $V_i$ , without loss of generality *i-critical*, such that  $v \in V_i$  and  $w \notin V_i$ . Let  $V_i$  be a minimal such set, and pick an arc  $uv \in A$  such that  $u \in V_i$ . Such an arc must exist, as otherwise  $d^+(V_i - v) < d^+(V_i) = d_i$ . Let  $U$  be a critical set that blocks the splitting off of  $uv$  and  $vw$ . First, assume that  $v \in U$ ; hence  $u, w \notin U$ . Then  $U$  is not *i-critical*, since otherwise  $U \cap V_i \subset V_i$ , contradicting the choice of  $V_i$ ; but if  $U$  is *j-critical*,  $j \neq i$ , then  $v \in U \cap V_i$  while  $w \notin U \cup V_i$ , which contradicts observation 2. Thus  $U \cap \{u, v, w\} = \{u, w\}$ . But  $U$  cannot be *i-critical*, by observation 1 and the arc  $uv$ , and it cannot be *j-critical*,  $i \neq j$ , since  $V_i \setminus U \subset V_i$ , contradicting the choice of  $V_i$ . Thus the first case is handled.

In the second case, let  $V_i$  be a maximal critical set with  $w \in V_i, v \notin V_i$ . Assume that  $V_i$  is *i-critical*. Let  $uv \in A$  such that  $u \notin V_i$ ; this exists, since otherwise  $d^+(V_i + v) < d^+(V_i) = d_i$ . Let  $U$  be a critical set that blocks the splitting of  $uv$  and  $vw$ . Then  $U \cap \{u, v, w\} = \{u, w\}$ , as otherwise the set  $U$  brings us back to case 1. If  $U$  is *i-critical*, then  $U \cup V_i$  is an *i-critical* set contradicting the choice of  $V_i$ , but if not, then  $w \in U \cap V_i$  and  $v \notin U \cup V_i$ , which is a contradiction by observation 2. Therefore, in both cases we find that there exists an arc  $uv$  such that  $uv$  and  $vw$  can be split off, and clearly if neither case applies then there cannot exist a critical set blocking the splitting off of  $uv$  and  $vw$  for any arc  $uv$ .  $\square$

The proof of the path-packing statement (Theorem 4.5.9) follows from this, by first repeatedly splitting off arcs until every vertex except  $T$  is

isolated, then unrolling these operations while maintaining a path-packing. Frank notes that a capacitated version can also be shown to be solvable in strongly polynomial time.

The following is an easy corollary of the path-packing result.

**Theorem 4.5.11** *The ARC MULTIWAY CUT problem on Euler digraphs is polynomial-time solvable.*

**Proof:** Clearly, the solution must have cardinality at least  $\sum_{i=1}^p d_i$ , by the existence of a path packing. But it is not difficult to produce a solution of exactly this size. For  $i = 1, \dots, p$ , let  $V_i$  be the  $i$ -critical set of minimum cardinality. By observation 1 of the previous proof these sets are unique, and by observation 2 the sets are also pairwise disjoint. Thus the set  $\bigcup_i (V_i, V \setminus V_i)_D$  is an arc multiway cut of cardinality matching the size of a path-packing, and therefore clearly optimal.  $\square$

As noted, the remarkable aspect of this result is that no comparable statement can be found for undirected graphs. If  $G$  is undirected and Euler, then even though there is a min-max result corresponding to Theorem 4.5.9 for the size of a path packing, there is no corresponding way to find a cut that intersects every path only once. In particular, the collection of closest min-cuts would hit some paths twice.

Finally, we have the following variant of Theorem 4.5.10, which generalizes a result of Lovász [30] for undirected Euler graphs, and has been shown independently by Frank [16] and by Jackson [25]. The proof is in the same spirit as Theorem 4.5.10.

**Theorem 4.5.12** *Let  $D = (V, A)$  be an Euler digraph and let  $vw \in A$ . There exists an arc  $uv \in A$  such that splitting off  $uv$  and  $vw$  does not affect the value of  $\lambda(s, t)$  for any vertices  $s, t \in V - v$ .*

#### 4.5.4 General Integral Weighted Path Packings

We now review a weighted generalization of FREE MULTIFLOW due to Hirai and Koichi [22]. To present the result, we need to introduce several notions.

First, we define networks. We will use a slightly different notion of a network than that given in Section 1.9, so to avoid ambiguity we introduce a different term. A **terminal network** is a triple  $(D, T, c)$  consisting of a digraph  $D = (V, A)$ , a set of terminals  $T \subseteq V$ , and a set of integer arc capacities  $c : A \rightarrow \mathbb{Z}_+$ . We say that the network is **balanced at  $v$**  for a vertex  $v \in V$  if

$$\sum_{uv \in A} c(uv) = \sum_{vw \in A} c(vw).$$

A **balanced terminal network** (respectively **inner balanced terminal network**) is a network which is balanced at every vertex  $v$  (respectively at

every vertex  $v \in V \setminus T$ ). This definition of a network differs from the usual one in that instead of an explicit balance vector, we have a set of terminals over which a flow is to be maximized. A **multiflow over  $T$**  is a pair  $(\mathcal{P}, \lambda)$  where  $\mathcal{P}$  is a collection of directed paths with all endpoints in  $T$ ,  $\lambda : \mathcal{P} \rightarrow \mathbb{R}_+$  a set of **flow values** for the paths in  $\mathcal{P}$ , and  $(\mathcal{P}, \lambda)$  satisfy the capacity constraints, i.e.,

$$\sum_{P \in \mathcal{P}: a \in P} \lambda(P) \leq c(a)$$

for every arc  $a \in A$ . Finally, a **directed distance** on  $T$  is a function  $\mu : T \times T \rightarrow \mathbb{R}_+$  such that  $\mu(x, x) = 0$  for every  $x \in T$ . Note that the triangle inequality is not required to hold. For a directed path  $P$ , starting and ending at terminals  $s$  and  $t$  in  $T$ , we let  $\mu(P) = \mu(s, t)$ . For a directed distance  $\mu$  on  $T$  and a multiflow  $(\mathcal{P}, \lambda)$  over  $T$ , the  **$\mu$ -weighted flow value** of  $(\mathcal{P}, \lambda)$  equals

$$\sum_{P \in \mathcal{P}} \lambda \cdot \mu(P).$$

The  $\mu$ -WEIGHTED MAXIMUM MULTIFLOW PROBLEM ( $\mu$ -MFP) is then defined as the problem where the input is a terminal network  $(D, T, c)$  and a directed distance  $\mu$  on  $T$ , and the task is to find a multiflow  $(\mathcal{P}, \lambda)$  over  $T$  which maximizes the  $\mu$ -weighted flow value.

This problem will in general have a fractional optimum, but for some directed distances  $\mu$ , the system will have an integral optimum for every balanced terminal network – for example, if we fix a directed distance  $\mu$  where  $\mu(s, t) = 1$  for all  $s, t \in T$ ,  $s \neq t$ , then  $\mu$ -MFP corresponds to the FREE MULTIFLOW problem, and the statement would follow from Theorem 4.5.9. The results of Hirai and Koichi imply a characterization of all directed distances  $\mu$  such that  $\mu$ -MFP has an integral optimum for every balanced terminal network.

The characterization is as follows. Let  $\Gamma$  be an oriented tree, and  $\alpha$  a set of non-negative (real-valued) edge weights of  $\Gamma$ . We define a directed metric  $d_{\Gamma, \alpha}$  on  $V(\Gamma)$  by letting  $d_{\Gamma, \alpha}(u, v)$  for  $u, v \in V(\Gamma)$  be the sum of  $\alpha(e)$  over all edges  $e$  of  $E(\Gamma)$  that are oriented from  $u$  to  $v$  in the path  $P_{uv}$  from  $u$  to  $v$  in  $\Gamma$ . An **oriented tree realization** of a directed distance  $\mu$  on  $T$  is a triple  $(\Gamma, \alpha, \{F_t\}_{t \in T})$  where  $\Gamma$  is an oriented tree,  $\alpha : E(\Gamma) \rightarrow \mathbb{R}_+$  a set of non-negative edge lengths of  $\Gamma$ , and  $\{F_t\}_{t \in T}$  a collection of subtrees of  $\Gamma$ , such that

$$\mu(s, t) = \min_{a \in F_s, b \in F_t} d_{\Gamma, \alpha}(a, b)$$

for all pairs  $s, t \in T$ .

**Theorem 4.5.13** ([22]) *Let  $\mu$  be a directed distance on a set of terminals  $T$ . Then the  $\mu$ -MFP is integral for every balanced network  $(D, T, c)$  if and only if  $\mu$  has an oriented tree realization.*

Let us consider a few examples. First, let  $\Gamma$  consist of a single arc  $st$ , with  $\alpha(st) > 0$ . Then  $d_\Gamma(s, t) = \alpha(st) > 0$ , while  $d_{\Gamma, \alpha}(t, s) = 0$  since the arc  $st$  is traversed in the wrong direction. Hence  $\mu$ -MFP reduces to the usual max-flow problem for any directed distance  $\mu$  realized by  $\Gamma$ . For another example, the FREE MULTIFLOW problem can be realized by a unit-weighted star  $\Gamma$ , with all arcs oriented into the root and with the collection  $\{F_t\}_{t \in T}$  being a bijection between  $T$  and the leaves of  $\Gamma$ .

Hirai and Koichi also give a matching min-max theorem for the positive cases, in terms of packing cuts in the oriented tree that realizes  $\mu$ ; we omit the details here.

## References

1. N. Alon, C. McDiarmid, and M. Molloy. Edge-disjoint cycles in regular directed graphs. *J. Graph Theory*, 22(3):231–237, 1996.
2. J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer-Verlag, London, 2nd edition, 2009.
3. H.L. Bodlaender, S. Thomassé, and A. Yeo. Kernel bounds for disjoint cycles and disjoint paths. *Theor. Comput. Sci.*, 412(35):4570–4578, 2011.
4. R.A. Brualdi and J. Shen. Disjoint cycles in eulerian digraphs and the diameter of interchange graphs. *J. Combin. Theory Ser. B*, 85(2):189–196, 2002.
5. J. Chen, Y. Liu, S. Lu, B. O’Sullivan, and I. Razgon. Directed Feedback Vertex Set is Fixed-Parameter Tractable. In *STOC 2008: 40th Annual ACM Symposium on Theory of Computing*. ACM Press, 2008.
6. R. Chitnis, M. Cygan, M. Hajiaghayi, M. Pilipczuk, and M. Pilipczuk. Designing FPT algorithms for cut problems using randomized contractions. *SIAM J. Comput.*, 45(4):1171–1229, 2016.
7. F.A. Chudak, M.X. Goemans, D.S. Hochbaum, and D.P. Williamson. A primal-dual interpretation of two 2-approximation algorithms for the feedback vertex set problem in undirected graphs. *Oper. Res. Lett.*, 22(4-5):111–118, 1998.
8. M. Chudnovsky, J. Geelen, B. Gerards, L. A. Goddyn, M. Lohman, and P.D. Seymour. Packing non-zero A-paths in group-labelled graphs. *Combinatorica*, 26(5):521–532, 2006.
9. B. Courcelle. The monadic second-order logic of graphs. I: Recognizable sets of finite graphs. *Inform. Comput.*, 85:12–75, 1990.
10. M. Cygan, D. Lokshtanov, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. Minimum bisection is fixed parameter tractable. In *STOC 2014: 46th annual ACM Symposium on Theory of Computing*, pages 323–332. ACM, 2014.
11. M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J.M.M. van Rooij, and J.O. Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In R. Ostrovsky, editor, *FOCS 2011: 52nd Annual Symposium on Foundations of Computer Science*, pages 150–159. IEEE Computer Society, 2011.
12. R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.
13. G. Even, J. Naor, B. Schieber, and M. Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2):151–174, 1998.

14. H. Fleischner and B. Jackson. Compatible Euler tours in Eulerian digraphs. In *Cycles and rays (Montreal, PQ, 1987)*, volume 301 of *NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci.*, pages 95–100. Kluwer, Dordrecht, 1990.
15. S. Fortune, J.E. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theor. Comput. Sci.*, 10:111–121, 1980.
16. A. Frank. On connectivity properties of Eulerian digraphs. In *Graph theory in memory of G. A. Dirac (Sandbjerg, 1985)*, volume 41 of *Ann. Discrete Math.*, pages 179–194. North-Holland, 1989.
17. A. Frank, T. Ibaraki, and H. Nagamochi. Two arc-disjoint paths in Eulerian digraphs. *SIAM J. Discrete Math.*, 11(4):557–589 (electronic), 1998.
18. M. Grohe and M. Grüber. Parameterized approximability of the disjoint cycle problem. In *ICALP 2007: 34th Int. Colloquium on Automata, Languages and Programming*, volume 4596 of *Lect. Notes Comput. Sci.*, pages 363–374. Springer-Verlag, 2007.
19. B. Guenin and R. Thomas. Packing directed circuits exactly. *Combinatorica*, 31(4):397–421, 2011.
20. V. Guruswami and E. Lee. Simple proof of hardness of feedback vertex set. *Theory Comput.*, 12(6):1–11, 2016.
21. G. Gutin, M. Jones, B. Sheng, and M. Wahlström. Parameterized directed  $k$ -chinese postman problem and  $k$  arc-disjoint cycles problem on euler digraphs. In *WG 2014: 40th International Workshop on Graph-Theoretic Concepts in Computer Science*, volume 8747 of *Lect. Notes Comput. Sci.*, pages 250–262, 2014.
22. H. Hirai and S. Koichi. On duality and fractionality of multicommodity flows in directed networks. *Discrete Optim.*, 8(3):428–445, 2011.
23. T. Ibaraki and S. Poljak. Weak three-linking in Eulerian digraphs. *SIAM J. Discrete Math.*, 4(1):84–98, 1991.
24. Y. Iwata. Linear-time kernelization for feedback vertex set. *CoRR*, abs/1608.01463, 2016.
25. B. Jackson. Some remarks on arc-connectivity, vertex splitting, and orientation in graphs and digraphs. *J. Graph Theory*, 12(3):429–436, 1988.
26. T. Johnson, N. Robertson, P.D. Seymour, and R. Thomas. Directed Tree-Width. *J. Combin. Theory Ser. B*, 82(1):138–154, 2001.
27. K. Kawarabayashi and S. Kreutzer. The directed grid theorem. In *STOC 2015: 47th Annual ACM on Symposium on Theory of Computing*, pages 655–664. ACM, 2015.
28. K. Kawarabayashi and M. Thorup. The minimum  $k$ -way cut of bounded size is fixed-parameter tractable. In R. Ostrovsky, editor, *FOCS 2011: 52nd Annual Symposium on Foundations of Computer Science*, pages 160–169. IEEE Computer Society, 2011.
29. T. Kociumaka and M. Pilipczuk. Faster deterministic feedback vertex set. *Inform. Process. Lett.*, 114(10):556–560, 2014.
30. L. Lovász. On some connectivity properties of Eulerian graphs. *Acta Math. Acad. Sci. Hung.*, 28(1–2):129–138, 1976.
31. C.L. Lucchesi and D.H. Younger. A minimax theorem for directed graphs. *J. London Math. Soc. (2)*, 17(3):369–374, 1978.
32. D. Marx, B. O’Sullivan, and I. Razgon. Finding small separators in linear time via treewidth reduction. *ACM Trans. Algor.*, 9(4):30:1–30:35, 2013.
33. G. Naves and A. Sebő. Multiflow feasibility: An annotated tableau. In W.J. Cook, L. Lovász, and J. Vygen, editors, *Research Trends in Combinatorial Optimization, Bonn Workshop on Combinatorial Optimization, 2008*, pages 261–283. Springer, 2008.
34. G. Pap. Packing non-returning  $A$ -paths. *Combinatorica*, 27(2):247–251, 2007.



35. B. Reed, N. Robertson, P.D. Seymour, and R. Thomas. Packing directed circuits. *Combinatorica*, 16(4):535–554, 1996.
36. N. Robertson and P.D. Seymour. Graph minors. XIII: The disjoint paths problem. *J. Combin. Theory Ser. B*, 63:65–110, 1995.
37. A. Schrijver. A short proof of Mader’s S-paths theorem. *J. Combin. Theory Ser. B*, 82(2):319–321, 2001.
38. P.D. Seymour. Packing circuits in Eulerian digraphs. *Combinatorica*, 16(2):223–231, 1996.
39. A. Slivkins. Parameterized tractability of edge-disjoint paths on directed acyclic graphs. In *ESA 2003: 11th Annual Eur. Symp. Algorithms*, volume 2832 of *Lect. Notes Comput. Sci.*, pages 482–493, 2003.
40. S. Thomassé. A  $4k^2$  kernel for feedback vertex set. *ACM Trans. Algor.*, 6(2), 2010.
41. W.T. Tutte and C.A.B. Smith. On unicursal paths in a network of degree 4. *Am. Math. Monthly*, 48(4):233–237, 1941.
42. T. van Aardenne-Ehrenfest and N.G. de Bruijn. Circuits and trees in oriented linear graphs. *Simon Stevin*, 28:203–217, 1951.
43. J. Vygen. NP-completeness of some edge-disjoint paths problems. *Discrete Appl. Math.*, 61(1):83–90, 1995.
44. D.H. Younger. Graphs with interlinked directed circuits,. In *Midwest symposium on circuit theory 2*, pages XVI 2.1–XVI 2.7, 1973.