



## 5. Planar Digraphs

Marcin Pilipczuk and Michał Pilipczuk

In this chapter we focus on *planar* directed graphs, that is, directed graphs that can be drawn on a plane (or, equivalently, on a sphere) without arc crossings. We will alternate between the planar and spherical embeddings, picking the more convenient for the current argumentation.

A **planar embedding** of a digraph  $D$  is a mapping  $\pi$  that assigns a distinct point in the Euclidean plane to every vertex of  $D$ , and a curve without self-intersections to every arc of  $D$  in such a manner that for every arc  $e = (u, v)$ , the curve  $\pi(e)$  has endpoints  $\pi(u)$  and  $\pi(v)$ , and the images of two arcs are disjoint (except for endpoints if the arcs in question share end vertices). A **face** in an embedding  $\pi$  is a connected component of the plane minus the image of  $\pi$ ; a face is **incident** with all vertices and arcs whose images under  $\pi$  lie in the closure of the face. A **spherical embedding** is defined analogously with the target surface being a sphere instead of a plane; intuitively, the main difference between a planar and a spherical embedding is that the first distinguishes one face as an infinite one.

After this very brief introduction, we refrain here from introducing all formal definitions and notation concerning graph embeddings, assuming instead a common intuitive understanding. In case of doubt, we refer to other monographs for formal details, e.g., to the book of Mohar and Thomassen [22].

The main goal of this chapter is to show, from multiple angles, how the planarity assumption imposes structure on digraphs and how such structure, in conjunction with topological arguments, can be used algorithmically. In other words, the main focus here is to show various algorithmic techniques used to tackle planar digraphs. Thus, instead of providing a survey of the vast number of algorithmic results concerning embedded digraphs, we highlight three of them, chosen to highlight different aspects of planar digraphs.

First, in Section 5.1 we show an example of a low polynomial-time algorithm for planar graphs, namely a near-linear algorithm for single-source and single-sink maximum flow. Second, in Section 5.2, we discuss the classic problem  $k$ -DISJOINT PATHS, where the topology assumption greatly improves the tractability of the problem. Finally, in Section 5.3 we discuss the Directed Grid Theorem for planar digraphs.

---

Marcin Pilipczuk (✉) and Michał Pilipczuk  
Institute of Informatics,  
University of Warsaw, Banacha 2, 02-097 Warsaw, Poland  
e-mail: malcin@mimuw.edu.pl

© 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\\_5](https://doi.org/10.1007/978-3-319-71840-8_5)

207

While we tried to make the description in every section as self-contained as possible, some technical details are missing in order to make the presentation clear and concise. In every section, we provide relevant references to full proofs and further reading.

## 5.1 Low Polynomial-Time Algorithms

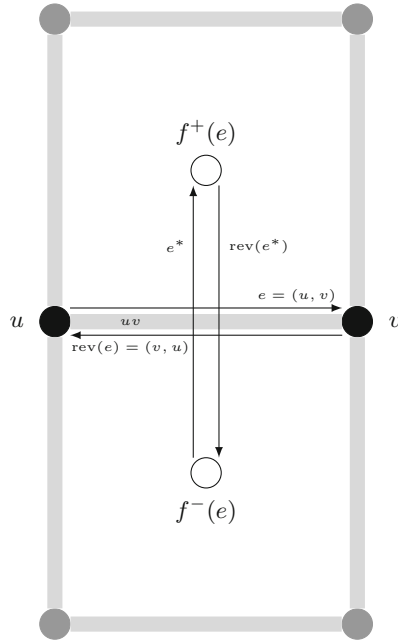
Part of the importance of planar graphs stems from the fact that many problems admit much more efficient solutions when the input graph is required to be planar. One of the areas where such improvements are particularly visible are low polynomial-time algorithms, such as algorithms for shortest paths or maximum flows. Decades of research led to linear-time or near-linear-time (e.g.,  $\mathcal{O}(n \log n)$  or even  $\mathcal{O}(n \log \log n)$ ) algorithms for problems requiring significantly larger running time in general graphs.

In this section, we do not aim at a full survey of these results for planar digraphs; the interested reader is referred to the free online book of Klein and Mozes [19]. Instead, we present one of the most elegant results in the area, namely the  $\mathcal{O}(n \log n)$ -time algorithm for finding the maximum flow between two given vertices due to Borradaile and Klein [2], with the simplified analysis due to Erickson [9]. We chose this result, as it involves a number of interesting techniques and properties of planar (di)graphs: duality of spanning trees in primal and dual graphs, duality of separators and cycles in dual graphs, as well as winding numbers analyzed via universal covers. The exposition mostly follows Chapter 10 of the book of Klein and Mozes [19], but we mainly focus on intuition, sweeping most of the technical details under the rug.

Because we will be working with residual capacities, we assume that we are given as an input a planar digraph  $D$  where for every arc  $e = (u, v)$  in  $D$  its reversed twin  $\text{rev}(e) = (v, u)$  is also in  $D$ . The input also specifies two distinguished vertices  $s$  and  $t$ , called the **source** and **sink**, and a capacity function  $u: A(D) \rightarrow \mathbb{Z}_{\geq 0}$ . If we replace every pair of arcs  $\{e = (u, v), \text{rev}(e)\}$  by an undirected edge  $uv$ , we obtain a planar undirected graph  $G$ . Without loss of generality, we can assume that  $G$  is connected. Let us fix some planar embedding of  $G$  where  $t$  lies on the outer face, denoted  $f^t$ .

In what follows, we will work with the assumed embedding of  $G$ , but also implicitly treat every undirected edge  $uv$  of  $G$  as two arcs  $(u, v)$  and  $(v, u)$  of  $D$ . Thus, for an arc  $e$  of  $D$ , we will speak about the face  $f^-(e)$  to the right (clockwise) of  $e$  and the face  $f^+(e)$  to the left (counter-clockwise) of  $e$ . Note that these notions formally refer to the faces of the embedding of  $G$ . We refer to Figure 5.1 for the basic notation of the dual graphs used in this proof.

For this fixed embedding, a dual of the graph  $G$  is a graph  $G^*$  whose vertex set is the set of faces of the embedding, and where an edge  $uv \in E(G)$  corresponds to an edge joining the two faces incident to  $uv$  in the embedding of  $G$ . Clearly,  $G^*$  is a planar graph with a natural embedding induced by the embedding of  $G$ . As in the case of  $D$  and  $G$ , if we replace every edge of  $G^*$  with two arcs in both directions, we obtain a digraph  $D^*$ . If  $e = (u, v)$  is an



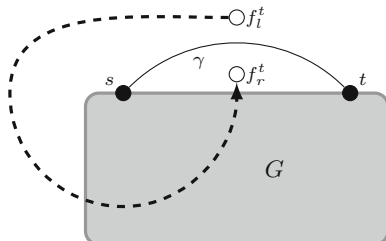
**Figure 5.1** Notation of the dual graphs, that is, graphs  $D$ ,  $G$ ,  $G^*$ , and  $D^*$ .

arc of  $D$ , then by  $e^* = (f^-(e), f^+(e))$  we denote the corresponding arc of  $D^*$ . We translate the capacities in  $D$  to *lengths* or *distances* in  $D^*$ : for an arc  $e \in A(D)$ , we assign in  $D^*$  distances  $w(e^*) = u(e)$  and  $w(\text{rev}(e^*)) = u(\text{rev}(e))$ .

Furthermore, in this section we assume that every multiset of arcs of  $D^*$  of polynomial size has a distinct sum of capacities. This property will turn out to be very helpful in the analysis. In general, this can be obtained by slightly perturbing every capacity; however, such a step would require some technical analysis of the required precision. Luckily, as we will discuss later, in our algorithm we can mimic such a property by a number of carefully chosen tie-breaking rules.

### 5.1.1 Warm-Up: Source also Lying on the Outer Face

As a warm-up, let us consider the case when the source  $s$  also lies on the outer face  $f^t$ . Draw a curve from  $s$  to  $t$  inside  $f^t$ : the curve partitions the arcs incident to  $f^t$  in  $D^*$  into two sets,  $A_l^*$  and  $A_r^*$ , to the left and to the right of the curve, respectively. Consider a graph  $D_{lr}^*$ , constructed from  $D^*$  by splitting  $f^t$  into two vertices  $f_l^t$  and  $f_r^t$ ; the first one is incident with arcs  $A_l^*$ , and the second one with  $A_r^*$ . The critical observation is that a minimum cut between  $s$  and  $t$  in  $D$  corresponds to a shortest path from  $f_l^t$  to  $f_r^t$  in  $D_{lr}^*$ ; see Figure 5.2. This can be found in  $\mathcal{O}(n \log n)$  time using Dijkstra’s



**Figure 5.2** Finding a minimum cut is equivalent to finding a shortest path in the dual in the case of  $s$  and  $t$  lying on a common face. The edge  $\gamma$  is an auxiliary edge of infinite distance that splits the face incident with  $s$  and  $t$  into two faces  $f_l^t$  and  $f_r^t$ ; a shortest path in the dual graph between these faces corresponds to a minimum cut between  $s$  and  $t$  in the primal graph.

algorithm, or in linear time using the algorithm of Henzinger, Klein, Rao, and Subramanian [14]. Both these algorithms find not only a shortest path from  $f_l^t$  to  $f_r^t$ , but also the minimum distances from  $f_l^t$  to all the vertex of  $D_{lr}^*$ .

To obtain a maximum flow, we need to work a bit harder. Let  $\text{dist}(f)$  be the (shortest path) distance from  $f_l^t$  to  $f$  in the graph  $D_{lr}^*$ . This distance has been computed already by the shortest path computation that identified a minimum cut. For an edge  $f^-(e)f^+(e)$  of  $G^*$  originating in an arc  $e$  of  $D$ , we send a flow of size  $\text{dist}(f^+(e)) - \text{dist}(f^-(e))$  along the arc  $e$  (that is, if  $\text{dist}(f^+(e)) < \text{dist}(f^-(e))$  we send a flow of  $\text{dist}(f^-(e)) - \text{dist}(f^+(e))$  along  $\text{rev}(e)$ ). Let  $x$  be the flow defined. Observe the following:

- Since  $\text{dist}(f)$  is the distance from  $f_l^t$  to  $f$ , the flow  $x$  respects capacities:  $x(e) = \text{dist}(f^+(e)) - \text{dist}(f^-(e)) \leq w(e^*) = u(e)$ .
- Since  $G^*$  is dual to  $G$ , the flow  $x$  respects the conservation property at every vertex except for  $s$  and  $t$ ; the latter is because in  $D_{lr}^*$  the face  $f^t$  has been split in two. One can view this splitting as drawing an auxiliary edge  $st$ , that is not present in  $x$ . Consequently,  $x$  is an  $(s, t)$ -flow of value  $\text{dist}(f_r^t)$ .

From the above, we can obtain the following result of [13, 14]:

**Theorem 5.1.1** *Given a planar digraph  $D$  with capacities and two distinguished vertices  $s$  and  $t$ , such that  $D$  can be embedded on a plane with  $s$  and  $t$  lying on the same face, a maximum  $(s, t)$ -flow and a minimum  $(s, t)$ -cut can be found in linear time.*

### 5.1.2 The Algorithm for the General Case

In the general case, we no longer assume that  $s$  lies on the face  $f^t$ , and hence we cannot construct a planar digraph  $D_{lr}^*$ . However, we can still rely on the crucial idea of the flow construction in the previous section: a shortest paths computation from  $f^t$  in  $D^*$  yields a distance function  $\text{dist}(\cdot)$  that can be used as a potential on faces to define a flow.

That is, similarly as in the previous case, let  $\text{dist}(f)$  be the distance of  $f$  from  $f^t$  in  $D^*$ , and define a flow  $x$  as before:  $x(e) = \text{dist}(f^+(e)) - \text{dist}(f^-(e))$  for an arc  $e$  of  $D$  with  $\text{dist}(f^+(e)) \geq \text{dist}(f^-(e))$ . Since now  $G^*$  is the actual dual of  $G$  (we do not split  $f^t$ ), with the same argument as in the previous section,  $x$  is a circulation respecting capacities.

Furthermore, let  $T^*$  be the computed shortest path tree in  $D^*$ , which is an out-branching with root  $f^t$ . Note that, since  $T^*$  is a shortest path tree, for every arc  $(f^-(e), f^+(e))$  of  $T^*$  we have  $\text{dist}(f^+(e)) = \text{dist}(f^-(e)) + w(e^*)$  and, consequently, the arc  $e$  is saturated in the flow  $x$ .

We shall now treat  $x$  as a flow from  $s$  to  $t$ . Initially the amount of the flow sent from  $s$  to  $t$  is zero, since  $x$  is a circulation at the beginning. We will gradually increase the amount of flow sent from  $s$  to  $t$  while maintaining the following invariant:

$$\begin{aligned} T^* \text{ is an out-branching with root } f^t \\ \text{and all corresponding arcs of } D \text{ are saturated by } x. \end{aligned} \quad (5.1)$$

At every step, given  $T^*$ , let  $T_G^*$  be the corresponding (undirected) spanning tree in  $G^*$ . Let  $T_G$  be the set of edges of  $G$  that are not crossed by the edges of  $T_G^*$ ; then  $T_G$  is a spanning tree of  $G$ . The tree  $T_G$  contains a unique  $s$ -to- $t$  path  $P$  in  $D$ . We augment  $x$  by sending the maximum possible amount of flow along this path (which may be zero, if one of the arcs of  $P$  is already saturated).

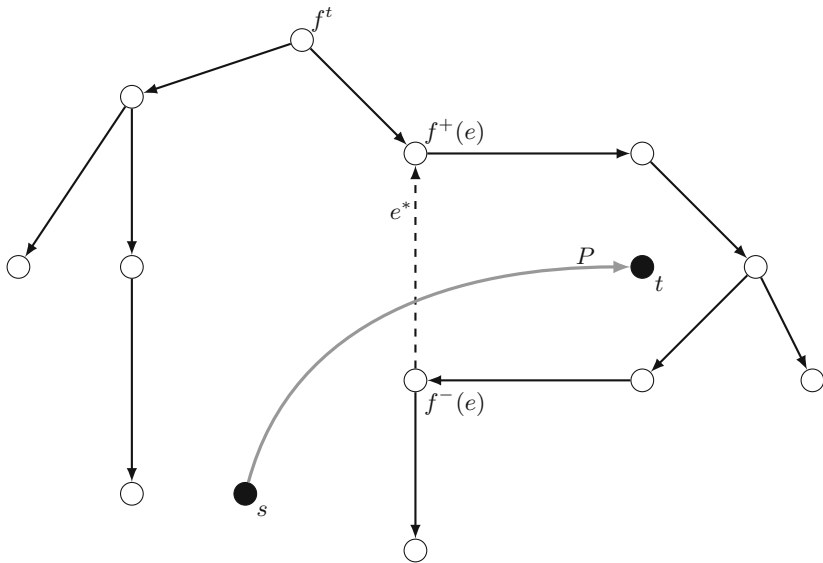
Then, we modify the out-branching  $T^*$  as follows. Let  $e$  be one of the arcs saturated on the path  $P$ . We would like to add the arc  $e^* = (f^-(e), f^+(e))$  to  $T^*$ . However, then  $T^*$  has one arc too many—it would no longer be an out-branching—and we need to fix it.

First, consider the case when  $f^-(e)$  is a descendant of  $f^+(e)$  in the out-branching  $T^*$  (see Figure 5.3). Then  $e^*$ , together with the path from  $f^+(e)$  to  $f^-(e)$  in  $T^*$ , form a directed cycle  $C^*$  in  $D^*$ . Note that the cycle  $C^*$  has the vertex  $s$  to the left and the vertex  $t$  to the right. Consequently, the arcs of  $D$  corresponding to the arcs of  $C^*$  form an  $(s, t)$ -cut that, by Invariant (5.1) and the choice of  $e$ , consists of arcs saturated by  $x$ . This cut certifies that  $x$  is a maximum  $(s, t)$ -flow and we can terminate the algorithm.

In the other case, when  $f^-(e)$  is not a descendant of  $f^+(e)$  in  $T^*$ , we replace the arc  $e'$  of  $T^*$  that has tail in  $f^+(e)$  with the arc  $e^*$ ; see Figure 5.4. Since  $f^-(e)$  is not a descendant of  $f^+(e)$ ,  $f^-(e)$  and  $f^+(e)$  lie in different connected components of  $T^* \setminus \{e'\}$  and, consequently, such an operation maintains the invariant that  $T^*$  is an out-branching. Furthermore, since we choose  $e^*$  to be saturated, Invariant (5.1) remains satisfied.

### 5.1.3 Implementing a Single Step

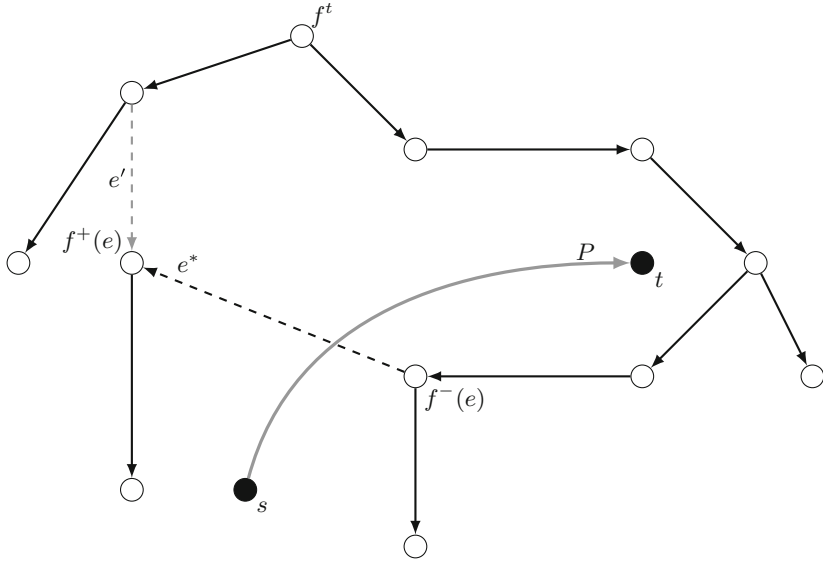
It turns out that a single step of the algorithm can be implemented very efficiently, in  $\mathcal{O}(\log n)$  time. However, since such an improvement belongs to the area of advanced data structures, we present here only the key ideas.



**Figure 5.3** When  $f^-(e)$  is a descendant of  $f^+(e)$ , then the saturated arcs  $e^*$  and of  $T^*$  form a saturated cut certifying that the current flow is a maximum one.

Let us analyze our needs. We need to maintain the trees  $T_G^*$  and  $T_G$ . In a single step, we first need to compute the minimum residual capacity on a single path in  $T_G$ , and then augment the flow  $x$  by sending this capacity along the path. Then, we modify  $T_G$  and  $T_G^*$  by switching a constant number of edges. All these operations can be performed in amortized  $\mathcal{O}(\log n)$  time per operation using one of the elaborate data structures for maintaining dynamic trees, such as the link-cut trees of Sleator and Tarjan [28]. For full details, we refer to the book of Klein and Mozes [19].

Recall that, for the sake of further analysis, we have assumed that every polynomial-size multiset of arcs of  $D^*$  has unique total length. We remark here that this can be mimicked in the algorithm by careful tie-breaking in two places where the algorithm can make an arbitrary choice: when it chooses the initial shortest-path out-branching  $T^*$ , and when it chooses the saturated arc  $e$  in each step of the algorithm.



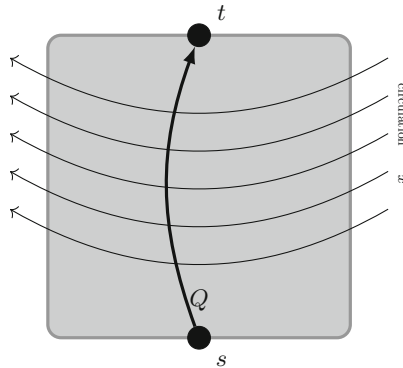
**Figure 5.4** When  $f^-(e)$  is not a descendant of  $f^+(e)$ , we replace  $e'$  with  $e^*$  in the out-branching  $T^*$ .

### 5.1.4 Bounding the Number of Steps

In this section we focus on the following question: *how many steps can the algorithm make?* We show that every arc of  $D^*$  is evicted from  $T^*$  at most once, giving an  $\mathcal{O}(n)$  bound on the number of steps, and, consequently, the promised  $\mathcal{O}(n \log n)$  bound on the running time of the algorithm.

**Winding numbers.** For the moment, it is convenient to interpret the planar embedding of  $D$  and  $D^*$  as an embedding on a sphere, where  $t$  is placed at the north pole and  $s$  is placed at the south pole; see Figure 5.5. One can think of the choice of the initial circulation  $x$  as a maximally *westbound* circulation in this embedding: we circulate as much flow as possible around the north pole in the westbound direction. Each iteration corresponds to “unwinding” some of this flow, and sending it from  $s$  to  $t$ .

To measure this “unwinding”, we need to fix some reference curve that would serve as a prime meridian between  $s$  and  $t$ . Although any  $s$ -to- $t$  path  $A$  in  $G$  would suffice, for clarity we choose  $Q$  to be the  $s$ -to- $t$  path in  $T_G$  at the first iteration of the algorithm. In the embedding, without loss of generality we can assume that  $Q$  is drawn as a straight line along the prime meridian, and we can use the notion of *west* or *east* of  $Q$ . To use  $Q$  as a reference line, we define a *winding number* of a walk  $W$  in  $D^*$  as the total number of signed crossings of  $Q$  by  $W$ . That is, we go along the walk  $W$ , and whenever we cross  $Q$  eastbound, we add 1 to the winding number, and when



**Figure 5.5** Visualizing  $t$  as the north pole,  $s$  as the south pole, the reference path  $Q$  as the prime meridian, and the initial circulation  $x$  as a maximally westbound circulation.

we cross  $Q$  westbound, we subtract 1. In the current step of the algorithm, given the current out-branching  $T^*$ , the **winding number** of a vertex  $f$  of  $D^*$  is the winding number of the unique root-to- $f$  path in  $T^*$ . Note that the choice of  $Q$  ensures that every winding number is zero at the beginning of the algorithm. We emphasize that, although  $T^*$  and  $T_G$  change in the course of the algorithm, the path (meridian)  $Q$  remains fixed.

The following critical observation due to Erickson [9] formalizes the “unwinding” nature of a single step of the algorithm.

**Lemma 5.1.2** *Assume that in a step of the algorithm, in an out-branching  $T^*$  a new arc  $e^*$  is introduced and an arc  $e'$  with tail  $f^+(e)$  is removed. Then, in the new out-branching, the winding number of every descendant of  $f^+(e)$  is increased by one, while all other winding numbers of vertices of  $D^*$  stay the same.*

**Proof:** First, note that replacing  $e'$  with  $e^*$  changes the root-to- $f$  paths in  $T^*$  only for vertices that are descendants of  $f^+(e)$  in  $T^*$ . Consequently, the winding number of every other vertex is not changed in the step of the algorithm.

For the affected vertices, consider the out-branching  $T^*$  before the step, and let  $P_-$  and  $P_+$  be the root-to- $f^-(e)$  and root-to- $f^+(e)$  paths, respectively. Let  $w$  be the last vertex in common of  $P_-$  and  $P_+$ , and let  $C$  be a closed walk in  $D^*$  that consists of  $P_-$ , the arc  $e^*$ , and the reversed path  $P_+$ . Note that during the step, for every descendant  $f$  of  $f^+(e)$  in  $T^*$  the root-to- $f$  path in  $T^*$  changes in the following manner: its prefix  $P_+$  is replaced by  $P_-$  followed by the arc  $e^*$ . Consequently, the change of the winding number of the root-to- $f$  paths equals the winding number of  $C$ .

By the choice of  $w$  and the fact that  $T^*$  is an out-branching,  $C$  is actually a simple cycle in  $D^*$ . Furthermore, by the choice of  $e^*$  in the step of the



algorithm,  $C$  has  $t$  to its left, and  $s$  to its right; in other words, it is an eastbound cycle in  $D^*$ , and thus has winding number exactly  $+1$ . This finishes the proof of the claim.  $\square$

Observe that Lemma 5.1.2 alone proves that the algorithm makes  $\mathcal{O}(n^2)$  steps, as every winding number cannot be larger than the size of  $D^*$  (every root-to- $f$  path in  $T^*$  is a simple path). We now present a more elaborate argument to show a linear bound.

**Shortest paths.** Recall that the distances  $\text{dist}(\cdot)$  in  $D^*$  have been inherited from the capacities  $u(\cdot)$  in  $D$  in a standard manner. Given a flow  $y$  in  $D$ , we can consider the residual capacities  $u_y := u - y$ , and define accordingly the residual distances  $\text{dist}_y$ .

If a flow  $y$  respects capacities—and the flow  $x$  maintained by the algorithm does respect the capacities—then no arc of  $D^*$  has negative length in  $\text{dist}_y$ . Invariant (5.1) ensures that every arc of  $T^*$  has zero length in  $\text{dist}_x$ . As a corollary, we infer that  $T^*$  is a shortest-path out-branching from  $f^t$  with respect to the distances  $\text{dist}_x$ .

Consider now a flow  $y$  that sends the same amount of flow from  $s$  to  $t$  as  $x$ , but sends all the flow along the path  $Q$ , ignoring the capacities. Although  $y$  may not respect the capacities, we can still define  $u_y$  and  $\text{dist}_y$ . Readers familiar with the potential method in designing shortest path algorithms will find the following lemma immediate.

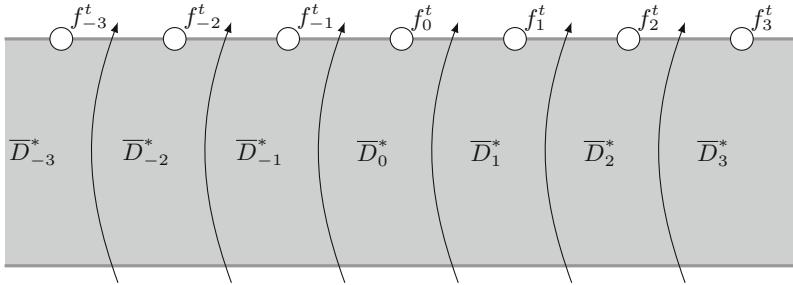
**Lemma 5.1.3**  *$T^*$  is a shortest-path out-branching from  $f^t$  with respect to the distances  $\text{dist}_y$ .*

**Proof:** The crux is that a flow  $y' := x - y$  (i.e., the flow  $x$  that additionally sends back the flow from  $t$  to  $s$  along the reversed path  $Q$ ) is a circulation (possibly not respecting the capacities).

Since  $y'$  is a circulation, we can define a potential function  $\zeta : V(D^*) \rightarrow \mathbb{R}$  such that  $y'(e) = \zeta(f^+(e)) - \zeta(f^-(e))$  for arcs  $e$  of  $D$  with  $\zeta(f^+(e)) \geq \zeta(f^-(e))$ . Indeed, we can treat the values of  $y'$  as (possibly negative) capacities of the arcs of  $D$ , translate them into a distance function  $\text{dist}'$  in  $D^*$  as before, and define  $\zeta(f)$  to be the minimum distance from  $f^t$  to  $f$  with respect to distances  $\text{dist}'$ . A direct check shows that  $\zeta$  satisfies the required properties and, since  $y'$  is a circulation, every walk from  $f^t$  to  $f$  has total length exactly  $\zeta(f)$ .

Consequently, if a path  $P$  from  $f^t$  to  $f$  has length  $\text{dist}_x(P)$  with respect to distances  $\text{dist}_x$ , then it has length  $\text{dist}_x(P) - \zeta(f)$  with respect to distances  $\text{dist}_y$ . Since  $\zeta(f)$  does not depend on the path  $P$ , but only on the endpoint  $f$ , we have that  $P$  is a shortest path from  $f^t$  with respect to  $\text{dist}_x$  if and only if it is a shortest path with respect to  $\text{dist}_y$ . The lemma follows.  $\square$

However, the simplicity of the flow  $y$  allows us to easily relate the distances in  $\text{dist}_y$  to the distances in  $\text{dist}$  that originated from the original capacities



**Figure 5.6** Universal cover of  $D^*$ .

$u$ . Indeed, if a path  $P$  has winding number  $i$  and the flow  $y$  sends  $\lambda$  amount of flow, then

$$\text{dist}_y(P) = \text{dist}(P) - \lambda \cdot i.$$

That is, the difference  $\text{dist}_y(P) - \text{dist}(P)$  depends only on the winding number of  $P$ . Consequently, we obtain the following:

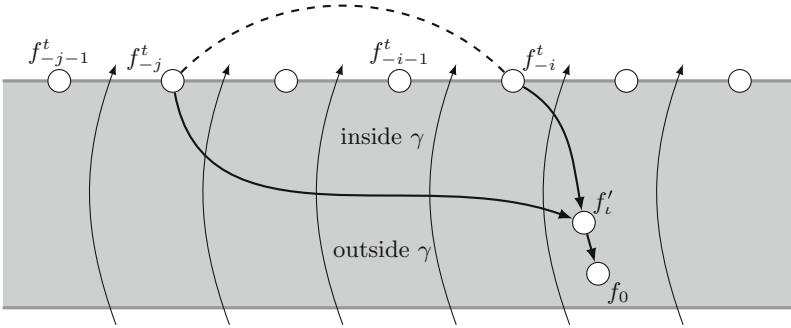
**Corollary 5.1.4** *For every vertex  $f$  of  $D^*$ , the root-to- $f$  path in  $T^*$  is the shortest  $f^t$ -to- $f$  path in  $D^*$  among the paths that have winding numbers equal to the winding number of  $f$ .*

**Universal cover.** Corollary 5.1.4 speaks about a shortest path among all paths of a given winding number. A convenient way to tackle the winding number is via **universal covers**.

In our setting, consider the following *infinite* cover  $\overline{D}^*$  of the graph  $D^*$ : we cut  $D^*$  along the path  $Q$  (which is a simple path in  $D$ , and thus corresponds to a face-edge curve of  $G^*$ ) and glue countably many copies of  $D^*$  cut along the path  $Q$ ; see Figure 5.6. The cover  $\overline{D}^*$  inherits the distances  $\text{dist}$  from  $D^*$ . We number the copies with integers, increasing in the eastbound direction. The  $i$ -th copy of  $D^*$  is denoted by  $\overline{D}_i^*$ , the  $i$ -th copy of a vertex  $f$  is denoted by  $f_i$ , etc. Since the path  $Q$  leads from  $s$  to  $t$ , the graph  $\overline{D}^*$  has a single face  $t^*$  corresponding to the vertex  $t$  (the north pole) and a single face  $s^*$  corresponding to the vertex  $s$  (the south pole). As in Figure 5.6, one can view the embedding of  $\overline{D}^*$  as an infinite strip, with  $t^*$  and  $s^*$  on its sides.

Observe that, given an integer  $i$ , every walk  $W$  in  $D^*$  can be lifted uniquely to a walk  $\overline{W}_i$  in  $\overline{D}^*$  that starts in the  $i$ -th copy of the first vertex of  $W$ , and then proceeds along the corresponding copies of the edges of  $W$ . The crux of the construction lies in the following observation: if the winding number of  $W$  is  $j$ , then the last vertex of  $\overline{W}_i$  lies in  $\overline{D}_{i+j}^*$ . In other words, when walking in  $\overline{D}^*$ , the index of the current copy reflects the winding number of the path traversed so far (when projected back to  $D^*$ ).

Consequently, if at some iteration the root-to- $f$  path in  $T^*$  has winding number  $i$ , then it corresponds to a path from  $f_{-i}^t$  to  $f_0$  and, in the other



**Figure 5.7** Final argument in the proof of the linear bound on the number of steps of the algorithm: the vertex  $f_0$  has to be inside and outside  $\gamma$  at the same time, as it needs to be reachable both from  $f_{-i-1}^t$  and  $f_{-j-1}^t$  without intersecting the closed curve  $\gamma$ .

direction, every  $f_{-i}^t$ -to- $f_0$  path in  $\overline{D}^*$  projects to a  $f^t$ -to- $f$  path in  $D^*$  of winding number  $i$ . By Corollary 5.1.4, we have the following.

**Lemma 5.1.5** *If at some iteration the root-to- $f$  path in  $T^*$  has winding number  $i$ , then it corresponds to a shortest path from  $f_{-i}^t$  to  $f_0$  in  $\overline{D}^*$ .*

Recall now that we have assumed that every nonempty multiset of arcs in  $D^*$  of polynomial size has unique total cost. This implies that a shortest path from  $f_{-i}^t$  to  $f_0$  is unique for any vertex  $f$  of  $D^*$  and any  $i$  bounded polynomially in the size of  $D$ . Furthermore, if we draw all these shortest paths for a fixed vertex  $f$  and  $|i| \in \mathcal{O}(n^2)$ , they do not cross, that is, we obtain an in-branching in  $\overline{D}^*$  with root  $f_0$ .

Aiming at a contradiction, consider now an arc  $e$  of  $D^*$  that was evicted twice from the tree  $T^*$ . Assume that the head of  $e$  is  $f$  and the tail is  $f'$ , and assume that the winding number of  $f$  just before the first eviction is  $i$ , and before the second is  $j$ . Due to Lemma 5.1.2, the winding number of  $f$  increased by one in both considered steps of the algorithm (when  $e$  is evicted from  $T^*$ ), which implies that  $i < j$ . Furthermore, it cannot hold that  $i+1 = j$ , as a arc from  $T^*$  different than  $e$  has its head in  $f$  immediately after the first of the considered steps, and thus the root-to- $f$  path in  $T^*$  needs to change at least once between the considered steps. Thus, we have  $j - i \geq 2$ .

As we discussed, the root-to- $f$  paths in  $T^*$  in the two considered steps correspond to two paths in  $\overline{D}^*$ , one from  $f_{-i}^t$  to  $f_0$  (henceforth denoted  $P_i$ ) and one from  $f_{-j}^t$  to  $f_0$  (henceforth denoted  $P_j$ ). Let  $P'_i$  and  $P'_j$  be the paths  $P_i$  and  $P_j$  with the last arc removed; note that the endpoint of  $P'_i$  and  $P'_j$  is  $f'_i$  for some  $i \in \{-1, 0, 1\}$ . If we connect  $f_{-i}^t$  with  $f_{-j}^t$  by a curve inside the face  $t^*$ , together with  $P'_i$  and  $P'_j$  we obtain a closed curve  $\gamma$ .

Since  $P_i$  and  $P_j$  are simple paths, we have that  $f_0$  does not lie on  $\gamma$ . Since  $P_i$  and  $P_j$  do not intersect (by the uniqueness assumption), we can speak

about vertices or arcs of  $\overline{D}^*$  **inside** and **outside** the curve  $\gamma$  (see Figure 5.7). The main question now is: where does the vertex  $f_0$  lie: inside or outside  $\gamma$ ?

Consider the first discussed iteration. After the iteration, the root-to- $f$  path in  $T^*$  corresponds to an  $f_{-i-1}^t$ -to- $f_0$  path  $P_{i+1}$  in  $\overline{D}^*$ . Since  $j - i \geq 2$ , the vertex  $f_{-i-1}^t$  is *inside*  $\gamma$  and, as  $P_{i+1}$  cannot cross  $P_i$  or  $P_j$ , the vertex  $f_0$  also needs to lie *inside*  $\gamma$ .

After the second discussed iteration, the root-to- $f$  path in  $T^*$  corresponds to an  $f_{-j-1}^t$ -to- $f_0$  path  $P_{j+1}$  in  $\overline{D}^*$ . However, now  $f_{-j-1}^t$  lies *outside*  $\gamma$  and, by a similar argument, implies that  $f_0$  also lies *outside*  $\gamma$ . This is the desired contradiction. Thus, every arc can be evicted from  $T^*$  at most once, giving an  $\mathcal{O}(n)$  bound on the number of steps and, consequently, the claimed  $\mathcal{O}(n \log n)$  running time bound for the algorithm.

### 5.1.5 Perspective

We have presented an algorithm for finding maximum single-source single-sink flows in planar digraphs running in near-linear time  $\mathcal{O}(n \log n)$ . While this result definitely does not cover the vast literature on algorithms in planar digraphs that run in low-polynomial time, we have chosen it to present key properties of planar digraphs that allow such running times. For a more exhaustive picture of related algorithms, as well as a presentation of the above algorithm from a different angle, we refer to the free textbook of Klein and Mozes [19].

## 5.2 The Disjoint Paths Problem

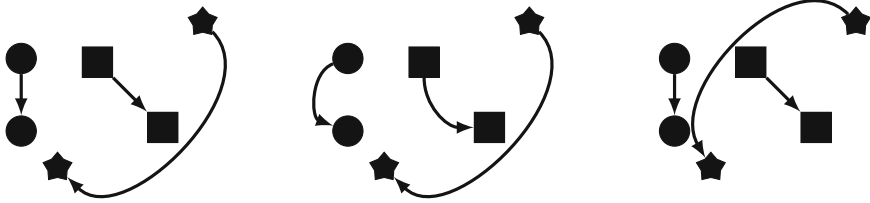
Let us consider the following problem:

$k$ -DISJOINT PATHS

**Input:** A digraph  $D$  with  $k$  pairs of terminals  $(s_1, t_1), \dots, (s_k, t_k)$ .

**Question:** Does  $D$  have vertex-disjoint directed paths  $P_1, \dots, P_k$  such that each  $P_i$  leads from  $s_i$  to  $t_i$ ?

In the undirected setting, the fixed-parameter tractability of this problem is one of the main algorithmic corollaries of the Graph Minors project of Robertson and Seymour: they gave an algorithm for it with running time  $f(k) \cdot n^3$  [24]. In directed graphs, however, the problem is completely intractable, as it is already NP-hard for  $k = 2$ , as shown by Fortune, Hopcroft, and Wyllie [11]. Some tractability can be retained in certain subclasses of digraphs. For instance, the problem can be solved in time  $n^{k+\mathcal{O}(1)}$  in acyclic digraphs by a simple dynamic programming algorithm, but it remains W[1]-hard in this setting, as shown by Slivkins [29], which means that the existence of a fixed-parameter algorithm with running time of the form  $f(k) \cdot n^{\mathcal{O}(1)}$  is



**Figure 5.8** Three solutions to  $k$ -DISJOINT PATHS on three terminal pairs, marked by different shapes. The first two are homotopic to each other, but not to the third.

unlikely. In this context, planar digraphs seem to be a setting where tractability is plausible, due to the inherent topological character of the  $k$ -DISJOINT PATHS problem. Indeed, in this section we will sketch the following result of Schrijver [26].

**Theorem 5.2.1** ([26]) *The  $k$ -DISJOINT PATHS problem can be solved in time  $n^{\mathcal{O}(k^2)}$  when the input digraph is planar.*

Take an instance  $(D, ((s_i, t_i))_{i=1, \dots, k})$  of the problem where  $D$  is planar, and suppose there is a solution  $P_1, \dots, P_k$ . Imagine each path  $P_i$  as a piece of string in the plane; vertex-disjointness means that the strings neither cross nor touch each other. Now abstract away the embedding of the graph and examine the picture consisting only of the strings. In the problem we do not care how long the paths are or which vertices they exactly traverse. We are content with a solution as long as the paths are vertex-disjoint and connect respective terminal pairs. Hence, we could consider two solutions as **homotopy equivalent** if one can be transformed into the other by a continuous transformation where terminals stay fixed and strings are not allowed to jump over terminals. More formally, for each  $i = 1, 2, \dots, k$ , the  $i$ th paths in both solutions are required to be homotopic on the sphere with the other terminals pierced out; see Figure 5.8.

The intuition is that the number of such string pictures, or rather of the equivalence classes of homotopy equivalence, that can be realized in the input digraph should not be too large. If we were able to quickly search for a solution within any such class, then the whole problem could be solved efficiently. Even though this is not what will actually happen in the algorithm, as it will rely on a weaker notion than homotopy equivalence, this intuition is a good first approximation of how the problem should be attacked.

More precisely, we will consider the **homology equivalence** for solutions, because for this notion of equivalence we are able to efficiently look for a solution within a fixed equivalence class. Homotopy equivalent solutions are always homologous, but the converse direction is not necessarily true. In

order to study homology equivalence, we need to introduce a certain mathematical language. In particular, we first look at the notion of **cohomology equivalence**, which intuitively is the same as homology equivalence, but in the dual digraph. While cohomology equivalence can be defined in any digraph, the translation between homology and cohomology relies on the relation between an embedded graph and its dual, and thus makes sense only for surface-embedded graphs.

### 5.2.1 Cohomology Equivalence and Feasibility

Cohomology equivalence is defined for digraphs with arcs labeled by elements of some fixed group. Let us fix  $\Lambda$  to be a free group on  $k$  generators  $g_1, \dots, g_k$ . That is, the support of  $\Lambda$  is the set of all finite words over symbols  $g_1, g_1^{-1}, \dots, g_k, g_k^{-1}$  that are **reduced**: symbols  $g_i$  and  $g_i^{-1}$  standing next to each other cancel out. The product of two elements  $x, y$  in  $\Lambda$ , denoted  $x \cdot y$ , is defined as the concatenation of  $x$  and  $y$  followed by an exhaustive application of reductions as above. The neutral element of  $\Lambda$  is the empty word, denoted by  $\varepsilon$ . For a digraph  $D$ , a  $\Lambda$ -**labeling** of  $D$  is any function  $\phi: A(D) \rightarrow \Lambda$  that assigns elements of  $\Lambda$  to the arcs of  $D$ .

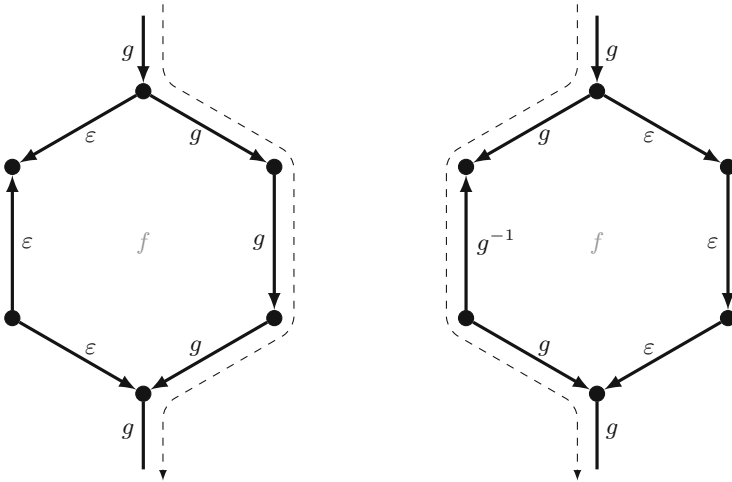
**Definition 5.2.2** *A pair of  $\Lambda$ -labelings  $\phi$  and  $\psi$  of a digraph  $D$  is called **cohomologous** if there exists a function  $\rho: V(D) \rightarrow \Lambda$  such that for each arc  $(u, v) \in A(D)$ ,*

$$\psi((u, v)) = (\rho(u))^{-1} \cdot \phi((u, v)) \cdot \rho(v).$$

*We say that  $\psi$  is cohomologous to  $\phi$  **via**  $\rho$ .*

It is clear that each  $\Lambda$ -labeling is cohomologous to itself by taking  $\rho(u) = \varepsilon$  for each vertex  $u$ . Also, the relation of being cohomologous is symmetric and transitive: if  $\phi$  is cohomologous to  $\psi$  via  $\rho$  and  $\psi$  is cohomologous to  $\zeta$  via  $\mu$ , then  $\psi$  is cohomologous to  $\phi$  via  $\rho^{-1}$  and  $\phi$  is cohomologous to  $\zeta$  via  $\nu$  defined as  $\nu(u) = \rho(u) \cdot \mu(u)$ .

Before we continue, let us discuss the intuition behind this notion. It is easy to see that a  $\Lambda$ -labeling  $\phi$  together with  $\rho: V(D) \rightarrow \Lambda$  uniquely define the labeling  $\psi$  cohomologous to  $\phi$  via  $\rho$ . Consider now changing the value of such  $\rho$  in one vertex  $u$  from  $\rho(u)$  to, say,  $\rho(u) \cdot g_1$ , where  $g_1$  is the first generator of  $\Lambda$ . It is easy to see that this triggers the following modification to  $\psi$ : for each arc  $a$  with  $u$  as the head,  $\psi(a)$  gets right-multiplied by  $g_1$ , while for each arc  $a'$  with  $u$  as the tail,  $\psi(a')$  gets left-multiplied by  $g_1^{-1}$ . Intuitively, this can be seen as “pulling” the group element  $g_1$  over  $u$  from the arcs outgoing from it to the arcs incoming to it, and  $\Lambda$ -labelings cohomologous to  $\phi$  are exactly those that can be obtained from  $\phi$  by a sequence of such “pulls”. If now  $D$  was the dual of some digraph  $D^*$ , then  $u$  corresponds to some face of  $D^*$ , and the pull can be seen as “dragging” the generator  $g_1$  over the face;



**Figure 5.9** Illustration of the “dragging” intuition. On the left panel, the values  $g$  on the arcs in the dual graph correspond to a directed dashed path in the depicted primal graph. By dragging the value  $g$  over the face  $f$ , one obtains the dashed path on the right panel; note that now the value on the middle arc is  $g^{-1}$  as it is traversed in the reverse direction.

see also Figure 5.9. This models a continuous modification of a solution to the  $k$ -DISJOINT PATHS problem by shifting some path by one face.

As we discussed, the main point of the approach is to show that we can efficiently search for a solution within a class of candidate solutions which are considered topologically equivalent. The main engine for this will be a polynomial-time algorithm for the COHOMOLOGY FEASIBILITY problem, defined as follows. Suppose we are given a digraph  $D$  and a  $\Lambda$ -labeling  $\phi$ . Suppose further that for each arc  $a \in A(D)$ , we are given a set  $H(a) \subseteq \Lambda$  that is **hereditary** in the following sense: if  $x \in H(a)$ , then every subword of the word  $x$  also belongs to  $H(a)$ . These sets are given by an oracle, that is, we assume there is a polynomial-time algorithm that given a word  $x$  and an arc  $a$ , checks whether  $x \in H(a)$ . Finally, we are also given a set  $S \subseteq V(D)$  of *fixed* vertices. The goal is to determine whether there exists a  $\Lambda$ -labeling  $\psi$  that is cohomologous to  $\phi$  via  $\rho$  satisfying the following conditions:

- $\psi(a) \in H(a)$  for each arc  $a \in A(D)$ ; and
- $\rho(u) = \varepsilon$  for each vertex  $u \in S$ .

The intuition for the  $k$ -DISJOINT PATHS problem is as follows. The digraph  $D$  is the dual of the original digraph. The initial labeling  $\phi$  corresponds to a crude picture of the solution, where the paths can touch or even share some subpaths, but they cannot cross in the plane. We are looking for a solution that is homologous (that is, cohomologous in the dual) and respects the disjointness conditions. By appropriately defining the dual and setting sets  $H(a)$ , the first property of  $\psi$  will be equivalent to the disjointness of the

paths. The second property will be used to ensure that the paths are not allowed to jump over terminals.

The backbone of the result of Schrijver is the following algorithmic result for COHOMOLOGY FEASIBILITY.

**Theorem 5.2.3** ([26]) *The COHOMOLOGY FEASIBILITY problem for free finitely generated groups is polynomial-time solvable.*

The proof of Theorem 5.2.3 is very technical, but the crux can be explained in modern terms as follows. We think of COHOMOLOGY FEASIBILITY as a constraint satisfaction problem (CSP) where vertices  $u \in V(D)$  are to be labeled by elements  $\rho(u)$  from the domain  $\Lambda$  such that some specific constraints are satisfied. It appears that the CSP problems constructed in this way are polynomial-time solvable, because they have certain *persistence* properties. Very roughly speaking, if some part of the problem can be solved without breaking any constraint, then one can greedily fix this solution on this part; this is the same phenomenon that leads to polynomial-time solvability of the 2-SAT problem. Stating and verifying the persistence, however, requires a lot of technical work. An interesting by-product of this approach is that if the algorithm of Theorem 5.2.3 reports failure, it also provides a combinatorial certificate for the non-existence of a solution, which can be exploited algorithmically. We refer to the notes of Schrijver for details [27].

## 5.2.2 Homology Equivalence and Duals

Having understood cohomology equivalence and the COHOMOLOGY FEASIBILITY problem, we now move to homology. Suppose we are given a planar digraph  $D$ , say embedded on a sphere with a fixed orientation. For each arc  $a \in A(D)$ , let  $f^-(a)$  and  $f^+(a)$  be the faces incident to  $a$  on the clockwise and counter-clockwise side, respectively. Similarly as in the previous section, we define the dual  $D^*$  of  $D$  as follows; see Fig. 5.10 for an example. The vertex set of  $D^*$  is the set  $F(D)$  of the faces of  $D$ . For each arc  $a$  of  $D$ , we add the *dual arc*  $a^* = (f^-(a), f^+(a))$  to the arc set of  $D^*$ . A sphere embedding of  $D$  naturally gives rise to a sphere embedding of  $D^*$ , where each arc crosses its dual at one point.

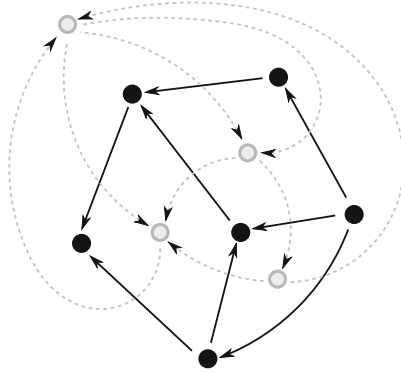
Now homology is defined as a dual notion to cohomology, hence we are allowed to pull over faces instead of vertices.

**Definition 5.2.4** *A pair of  $\Lambda$ -labelings  $\phi$  and  $\psi$  of a sphere-embedded digraph  $D$  is called **homologous** if there exists a function  $\rho: F(D) \rightarrow \Lambda$  such that for each arc  $a \in A(D)$ ,*

$$\psi(a) = (\rho(f^-(a)))^{-1} \cdot \phi(a) \cdot \rho(f^+(a)).$$

*We say that  $\psi$  is homologous to  $\phi$  via  $\rho$ .*





**Figure 5.10** A planar digraph (black) and its dual (grey).

Thus, the COHOMOLOGY FEASIBILITY problem in the dual  $D^*$  naturally translates to the analogous problem in  $D$ , where we are looking for a homologous  $\Lambda$ -labeling satisfying certain constraints. For instance, if in the COHOMOLOGY FEASIBILITY problem on  $D^*$  we put  $H(a^*) = \{\varepsilon, g_1, g_2, \dots, g_k\}$  for each arc  $a \in A(D)$ , then we are effectively looking for a  $\Lambda$ -labeling  $\psi$  of  $D$  homologous to the given labeling  $\phi$  such that the label of each arc is either the neutral element or one of the generators. Thus, each generator  $g_i$  gives rise to the arc subset  $\psi^{-1}(g_i)$  such that those subsets are pairwise disjoint. By appropriately choosing  $\phi$  we will be able ensure that  $\psi^{-1}(g_i)$  contains a path from  $s_i$  to  $t_i$  and these paths are non-crossing as curves in the plane, however they may touch at vertices. To ensure real vertex-disjointness, we need to augment the dual graph slightly.

Take the dual  $D^*$  of  $D$ . For each vertex  $u \in V(D)$  and each pair of faces  $f_1, f_2$  that are incident to  $u$ , but are not consecutive in the cyclic ordering of faces around  $u$ , we add arcs  $(f_1, f_2)$  and  $(f_2, f_1)$ . These new arcs will be called **contact arcs**, and the digraph obtained from the dual by adding all contact arcs is called the **extended dual**, denoted  $D^+$ . Note that the extended dual is not necessarily planar, but this will not be a problem, since the algorithm for COHOMOLOGY FEASIBILITY works on any digraph.

### 5.2.3 Disjoint Paths in Directed Planar Graphs

With all the tools prepared, we are ready to encode the search for a solution within one homology type as an instance of COHOMOLOGY FEASIBILITY. We first need to describe a homology type via a representative  $\Lambda$ -labeling.

Let us fix an instance  $(D, ((s_i, t_i))_{i=1, \dots, k})$  of  $k$ -DISJOINT PATHS. Without loss of generality we may assume that each source  $s_i$  has exactly one outgoing arc and no incoming arcs, whereas each sink  $t_i$  has exactly one incoming arc and no outgoing arcs. Indeed, we may add new sources and sinks as degree-

one vertices adjacent only to the corresponding old sources and sinks. The following definition describes initial labelings we are interested in.

**Definition 5.2.5** *A  $\Lambda$ -labeling  $\phi: V(D) \rightarrow \Lambda$  is consistent if the following conditions are satisfied:*

- For each source  $s_i$  and  $t_i$ , both the only arc outgoing from  $s_i$  and the only arc incoming to  $t_i$  are labeled by  $g_i$  in  $\phi$ .
- For each non-terminal node  $u$ , if  $a_1, \dots, a_\ell$  are arcs incident to  $u$  in the clockwise order around  $u$ , and  $b_1, \dots, b_\ell \in \{-1, +1\}$  are such that  $a_i$  has  $u$  as the head if and only if  $b_i = +1$ , then

$$\phi(a_1)^{b_1} \cdot \phi(a_2)^{b_2} \cdot \dots \cdot \phi(a_\ell)^{b_\ell} = \varepsilon.$$

Note that in the second condition it does not matter from which arc we start the enumeration of arcs incident to  $u$ : if the product is  $\varepsilon$  for one possible starting arc, it is  $\varepsilon$  for all of them.

Observe that the conditions in the above definition somewhat resemble flow conservation equations. The first condition says that each  $s_i$  is a “source” of one unit of the flow of type  $g_i$ , and each  $t_i$  is a “sink” of  $g_i$ . The second condition says that every nonterminal vertex satisfies a conservation property much stronger than the usual flow conservation: not only the incoming flow needs to be equal to the outgoing one, but also in some sense the flow paths cannot “cross” at a vertex.

On the other hand, the definition of a consistent labeling allows for multiple paths to be routed via the same arc, and even in the wrong direction; this corresponds to the possibility of having the label being not just a single generator. The idea is to express the requirement that this is forbidden in the language of the COHOMOLOGY FEASIBILITY problem.

Let  $\phi$  be a consistent  $\Lambda$ -labeling of  $D$ . Consider now the following COHOMOLOGY FEASIBILITY instance  $I(\phi)$  on the extended dual  $D^+$ . As the given  $\Lambda$ -labeling of  $D^+$  we take  $\phi^+$  defined as follows:

- For each arc  $a$  of  $D$ , put  $\phi^+(a^*) = \phi(a)$ .
- For each contact arc  $(f_1, f_2)$ , say added for a vertex  $u$ , let  $a_1, \dots, a_p$  be the consecutive arcs incident to  $u$  that we encounter when scanning the arcs around  $u$  in the clockwise order, starting from  $f_1$  and ending in  $f_2$ . Further, let  $b_1, \dots, b_p \in \{-1, +1\}$  be such that  $a_i$  has  $u$  as the head if and only if  $b_i = +1$ . Then  $\phi^+((f_1, f_2)) = \prod_{i=1}^p \phi(a_i)^{b_i}$ .

Next, we put  $H(a^*) = \{\varepsilon, g_1, \dots, g_k\}$  for each  $a \in A(D)$ , while for each contact arc  $(f_1, f_2)$ , we put  $H((f_1, f_2)) = \{\varepsilon, g_1, \dots, g_k, g_1^{-1}, \dots, g_k^{-1}\}$ . Finally, the set  $S$  of forbidden vertices of  $D^+$  consists of all faces of  $D$  that are incident to some terminal. The following proposition, whose proof we leave as an easy exercise, explains that solving the instance  $(D^+, \phi^+, H, S)$  of COHOMOLOGY FEASIBILITY immediately yields the solution to the whole problem.

**Proposition 5.2.6** *Suppose  $\psi$  is a solution to the instance  $(D^+, \phi^+, H, S)$ . For  $i = 1, 2, \dots, k$ , let  $X_i$  be the set of those arcs  $a$  of  $D$  for which  $\psi(a^*) = g_i$ . Then the subgraphs induced by  $X_1, \dots, X_k$  in  $D$  are pairwise vertex-disjoint and the subgraph induced by  $X_i$  contains a directed path leading from  $s_i$  to  $t_i$ .*

If now  $\mathcal{P} = (P_1, \dots, P_k)$  is a solution to the original instance, then we can define a consistent labeling  $\phi_{\mathcal{P}}$  of  $D$  as follows: for each arc  $a \in A(D)$ , put  $\psi_{\mathcal{P}}(a) = g_i$  if  $a$  lies on  $P_i$ , and put  $\psi_{\mathcal{P}}(a) = \varepsilon$  if  $a$  does not lie on any of the paths  $P_i$ . Then it is easy to see that  $\psi_{\mathcal{P}}^+$  is a feasible solution to  $(D^+, \phi^+, H, S)$  for any consistent labeling  $\phi$  with the following property:  $\phi$  is homologous to  $\psi_{\mathcal{P}}$  via some  $\rho$  which maps all faces of  $S$  to  $\varepsilon$ .

Thus, we will apply the following strategy: we enumerate a small set  $\Phi$  of consistent labelings of  $D$  such that if there is a solution  $\mathcal{P}$  to the problem, then  $\Phi$  contains a labeling  $\phi$  that is **well-homologous** to  $\psi_{\mathcal{P}}$ , that is, homologous via some  $\rho$  as above. Such a set  $\Phi$  will be called **exhaustive**. Then the algorithm for  $k$ -DISJOINT PATHS boils down to iterating through an exhaustive set  $\Phi$ , and for each  $\phi \in \Phi$  solving the COHOMOLOGY FEASIBILITY instance  $(D^+, \phi^+, H, S)$ . If we obtain a solution for any of these instances, Proposition 5.2.6 gives us a way to extract a solution to the original problem. Otherwise, if none of the instances has a solution, then we can conclude that the original problem has no solution, because  $\Phi$  is exhaustive.

Thus, to conclude the proof of Theorem 5.2.1, it remains to prove the following lemma. Since a complete verification requires some technical details, we give a short sketch.

**Lemma 5.2.7** *There exists an exhaustive set  $\Phi$  of size  $n^{\mathcal{O}(k^2)}$  which can be constructed in time  $n^{\mathcal{O}(k^2)}$ .*

**Proof: (Sketch)** First, we generalize the problem slightly. We will be interested in families of walks  $\mathcal{P} = (P_1, \dots, P_k)$  such that:

- Each  $P_i$  is a walk connecting  $s_i$  with  $t_i$  in the undirected graph underlying  $D$ . That is, we do not require that the arcs on each  $P_i$  are oriented in the direction from  $s_i$  to  $t_i$ , and a vertex can be visited by  $P_i$  multiple times.
- Walks  $P_i$  are pairwise arc-disjoint and non-crossing. That is, whenever we look at two visits of a vertex  $u$  by some  $P_i$  and  $P_j$  (possibly  $i = j$ ), then the four arcs incident to  $u$  in these two visits are not interlacing in the cyclic order of arcs around  $u$ .

We will call such families of walks **pre-solutions**. As before, each pre-solution  $\mathcal{P}$  naturally defines a consistent labeling  $\psi_{\mathcal{P}}$ . We are interested in finding a small set  $\Phi$  of consistent labelings of  $D$  that is exhaustive for all pre-solutions: for each pre-solution  $\mathcal{P}$ , there exists a labeling  $\phi$  in  $\Phi$  that is well-homologous to  $\psi_{\mathcal{P}}$  as in the definition of being exhaustive. As every solution is also a pre-solution, this suffices to prove the lemma.

The next step is to simplify the graph at hand to the case where there is exactly one vertex other than sources and sinks. However we will introduce

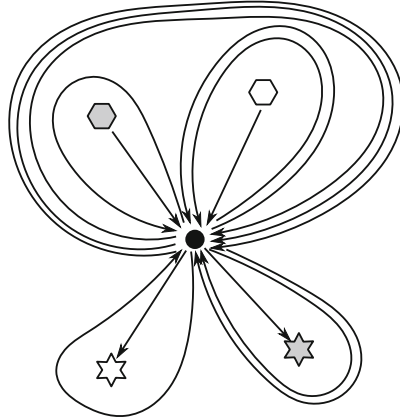
loops (arcs with the head equal to the tail). Consider any non-loop arc  $a$  such that neither the head nor the tail of  $a$  is a terminal. Construct the digraph  $D'$  by contracting  $a$ : identify the head and the tail of  $a$  and remove  $a$  from the graph. Every arc that is parallel to  $a$ , that is, has the same head and tail as  $a$ , or its head is the tail of  $a$  and vice versa, becomes a loop at the vertex obtained by identifying the endpoints of  $a$ . It is easy to see that every pre-solution in  $D$  can be naturally projected to a pre-solution in  $D'$ , and every consistent labeling  $\phi'$  of  $D'$  can be naturally lifted to a consistent labeling  $\phi$  of  $D$  so that the following holds: if  $\psi_{\mathcal{P}'}$  is well-homologous to  $\phi'$  in  $D'$ , where  $\mathcal{P}'$  is the projection of  $\mathcal{P}$ , then  $\psi_{\mathcal{P}}$  is well-homologous to  $\phi$  in  $D$ . Thus, it suffices to find a small exhaustive set in  $D'$ .

Supposing the original digraph is weakly connected, we can apply this reduction exhaustively until the vertex set of  $D$  consists of sources  $s_i$ , each with one outgoing arc, sinks  $t_i$ , each with one incoming arc, and one vertex  $u$  that has multiple loops attached to it. The number of these loops is bounded by  $m$ , the number of arcs of the initial graph, which is bounded linearly in  $n$ .

Let  $T$  be the set of all terminals. Each loop  $a$  at the vertex  $u$  can be associated with a partition  $\{X_a, Y_a\}$  of  $T$  as follows: the drawing of  $a$  on the sphere divides it into two regions, and  $X_a$  and  $Y_a$  are the subsets of terminals contained in these regions, respectively. Two loops  $a, a'$  at  $u$  will be called **parallel** if the partitions  $\{X_a, Y_a\}$  and  $\{X_{a'}, Y_{a'}\}$  are equal; of course, being parallel is an equivalence relation. Since the drawing of the loops is non-crossing, it is not hard to convince oneself that parallel loops are homotopic in the topological space formed by the sphere on which the whole drawing is embedded, with the terminals pierced out. Therefore, the equivalence classes of the relation of being parallel really look like sets of parallel arcs: they can be ordered so that there are faces of length 2 between every two consecutive ones, as in Fig. 5.11. Each such equivalence class will be called a **bundle**. Since we do not care about the orientation of arcs in pre-solutions, we may assume that all arcs in each bundle are oriented in the same manner, as in Fig. 5.11. Formally, each 2-face between consecutive arcs of the bundle is not an oriented 2-cycle.

We may assume that there is no bundle for which the corresponding partition is  $\{\emptyset, T\}$ , as arcs from such a bundle can be always removed from walks of any pre-solution without any harm. Then it is not hard to prove that since the bundles are non-crossing, their number is bounded by  $2|T| - 3 \leq 4k$ . By somehow abusing the notation, we treat the arcs outgoing from sources and incoming to sinks also as one-arc bundles, which increases the total number of bundles to at most  $6k$ .

We now explain the crux of the argument. Consider any pre-solution  $\mathcal{P} = (P_1, \dots, P_k)$ . Take any walk  $P_i$  and let  $a_1, a_2, \dots, a_p$  be the consecutive arcs traversed by  $P_i$ . Further, let  $B_1, B_2, \dots, B_p$  be bundles such that  $a_j \in B_j$  for each  $j \in \{1, 2, \dots, p\}$ . For each  $j = 1, 2, \dots, p - 1$ , let us *charge* the pair  $(B_j^\alpha, B_{j+1}^\beta)$ , where  $\alpha$  is equal to  $\pm 1$  depending on whether  $a_j$  is oriented in the



**Figure 5.11** The situation after applying the contractions. Sources are depicted as hexagons, sinks as stars, and the middle vertex is  $u$ . The loops at  $u$  are partitioned into 5 bundles.

direction from  $s_i$  to  $t_i$  on  $P_i$ , or from  $t_i$  to  $s_i$ ;  $\beta$  is defined in the same manner for  $a_{j+1}$ . For a pair  $(A^\alpha, B^\beta)$ , where  $A, B$  are bundles and  $\alpha, \beta \in \{-1, +1\}$ , let  $c(A^\alpha, B^\beta)$  be the number of times the pair  $(A^\alpha, B^\beta)$  is charged; obviously  $c(A^\alpha, B^\beta) \leq m$ .

The following claim is now crucial: the system of numbers  $c(A^\alpha, B^\beta)$  uniquely defines a pre-solution, up to being well-homologous. The proof of this fact is not hard and boils down to a careful reconstruction of a pre-solution from the numbers  $c(A^\alpha, B^\beta)$ , using the fact that walks in a pre-solution are pairwise non-crossing. There are at most  $4 \cdot (6k)^2$  numbers  $c(A^\alpha, B^\beta)$ , and each of them attains a value between 0 and  $m$ , hence the number of pre-solutions reconstructed in this manner is bounded by  $n^{\mathcal{O}(k^2)}$ .  $\square$

#### 5.2.4 Fixed-Parameter Algorithm: Highlights

The algorithm of Schrijver that we sketched above was later revisited by Cygan, Marx, Pilipczuk, and Pilipczuk [7], who improved the running time from the form  $n^{f(k)}$  to fixed-parameter tractable. More precisely, they proved the following.

**Theorem 5.2.8** ([7]) *The  $k$ -DISJOINT PATHS problem can be solved in time  $2^{2^{\mathcal{O}(k^2)}} \cdot n^c$  when the input digraph is planar, where  $c$  is a universal constant.*

To prove Theorem 5.2.8 it is sufficient to give an exhaustive set of size  $2^{2^{\mathcal{O}(k^2)}} \cdot n^c$ , as the size of an exhaustive set was the only bottleneck in the algorithm of Schrijver. Unfortunately, the number of different homology classes

of solutions can be as large as  $n^{\Omega(k)}$ , hence we cannot hope for such a small exhaustive set in general. Therefore, Cygan *et al.* resorted to using the **irrelevant vertex technique** as follows.

Let  $u$  be a non-terminal vertex of the input digraph  $D$ . A sequence  $C_1, C_2, \dots, C_\ell$  of vertex-disjoint cycles in  $D$  is called a **concentric sequence of alternating orientation around  $u$**  if the following conditions are satisfied.

- Each cycle  $C_i$  separates cycles  $C_j$  for  $j < i$  from cycles  $C_j$  for  $j > i$  in the plane.
- None of the cycles passes through  $u$ . Moreover, for each  $i = 1, 2, \dots, k$ , the region of the plane with  $C_i$  cut out to which  $u$  belongs does not contain any terminals.
- For even  $i$ , the cycle  $C_i$  goes around  $u$  in the clockwise direction, and for odd  $i$  in the counterclockwise.

Intuitively, if a vertex  $u$  can be encircled by such a concentric sequence of alternating orientation of large size, then it is “far” from terminals and not likely to be used in the solution. Cygan *et al.* formalized this intuition by proving that given the sequence is large enough, any solution can be rerouted to a solution that does not traverse  $u$ , and hence  $u$  can be safely removed from the instance.

**Lemma 5.2.9** ([7]) *There is a function  $d(k) \in 2^{\mathcal{O}(k^2)}$  such that the following holds. Suppose  $u$  is a non-terminal vertex around which there exists a concentric sequence of cycles of alternating orientation of size  $d(k)$ . Then if there exists a solution, there is also a solution in which  $u$  is not traversed by any path.*

Therefore, we can remove such vertices exhaustively from the instance. Cygan *et al.* then show that in the absence of such vertices, there is a small exhaustive set.

**Lemma 5.2.10** ([7]) *Suppose there is no vertex  $u$  that satisfies the prerequisite of Lemma 5.2.9. Then there exists an exhaustive set  $\Phi$  of size at most  $2^{2^{\mathcal{O}(k^2)}}$  that can be constructed in time  $2^{2^{\mathcal{O}(k^2)}} \cdot n^{\mathcal{O}(1)}$ .*

The algorithm claimed in Theorem 5.2.8 now boils down to solving an instance of COHOMOLOGY FEASIBILITY for each labeling in  $\Phi$ , exactly as in the previous section. The improved bound on the size of the exhaustive set gives us the fixed-parameter tractable upper bound on the running time.

The proof of Lemma 5.2.9 in [7] is based on a complicated analysis of the interaction of a solution to the  $k$ -DISJOINT PATHS with a sequence of concentric cycles of alternating orientation. It is proved that if the sequence is large enough, its cycles can be used as shortcuts for the paths in the solution, so that the paths can be rerouted simultaneously in order not to traverse

vertex  $u$ . This argument is based on a similar analysis for the undirected case performed by Adler, Kolliopoulos, Krause, Lokshtanov, Saurabh, and Thilikos [1].

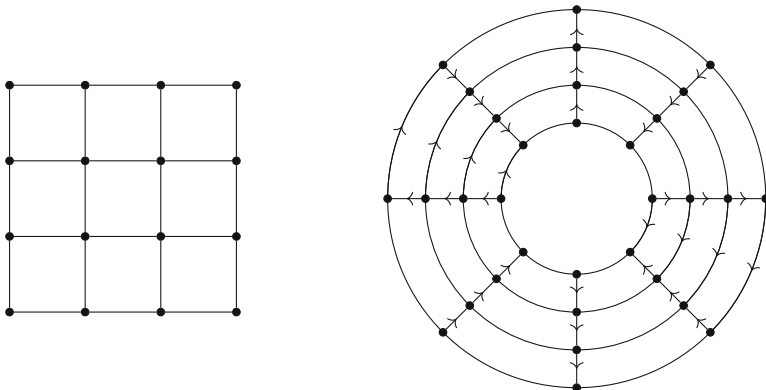
The most technically involved part of the reasoning is the proof of Lemma 5.2.10. Cygan *et al.* proved that in absence of vertices that are irrelevant in the sense of Lemma 5.2.9, the graph can be decomposed into a small number of components, each of them embedded into a disc or into a ring in the plane. The boundary of each component is well-behaved: if one travels along the boundary of, say, a disc component, then the number of times one sees an arc incoming to the component after an outgoing one, or vice versa, is bounded by a function of  $k$ . Having computed such a decomposition, one enumerates an exhaustive set of  $\Lambda$ -labelings by means of a branching procedure that “guesses” consecutive parts of a homology type. Both the depth and the degree of the search tree of this branching procedure are bounded in terms of  $k$ , hence the number of labelings produced by the procedure is bounded by a function of  $k$ .

### 5.2.5 Perspective

The fixed-parameter algorithm of [7] has double-exponential dependency on the parameter, namely  $2^{2^{\mathcal{O}(k^2)}}$ , which is very close to the  $2^{2^{\mathcal{O}(k)}}$  dependency in the fastest known algorithm for undirected planar graphs, due to Adler, Kolliopoulos, Krause, Lokshtanov, Saurabh, and Thilikos [1]. In the undirected case, the algorithm of [1] follows a typical irrelevant vertex approach: if the treewidth of the graph is larger than  $\Delta := 2^{\theta(k)}$ , an irrelevant vertex inside a  $\mathcal{O}(\Delta) \times \mathcal{O}(\Delta)$  grid minor is identified and deleted, whereas in the other case a standard dynamic programming routine on graphs of bounded treewidth runs in time  $2^{\mathcal{O}((\Delta+k) \log \Delta)} n = 2^{2^{\mathcal{O}(k)}} n$ . In [1], the authors show that this is the limit of this approach: the dependency  $\Delta = 2^{\Omega(k)}$  is necessary for the irrelevant vertex rule, while multiple lower bounds for dynamic programming algorithms on graphs of bounded treewidth (see the survey of Lokshtanov, Marx, and Saurabh [20]) strongly suggest that an exponential dependency on the treewidth bound  $\Delta$  is necessary for the second step of the algorithm. Hence, while there are no known lower bounds refuting the existence of an algorithm for  $k$ -DISJOINT PATHS in undirected planar graphs with only single-exponential dependency on the parameter, such an algorithm would need to depart significantly from the current framework and the question of its existence remains widely open.

## 5.3 Directed Grids

In this section we discuss the Directed Grid Theorem (Theorem 9.3.14) in the context of planar digraphs. The Directed Grid Theorem is a directed



**Figure 5.12** An undirected grid and a directed cylindrical grid.

analog of the Excluded Grid Theorem for undirected graphs, asserting that any graph of sufficiently large treewidth contains a large grid as a minor.

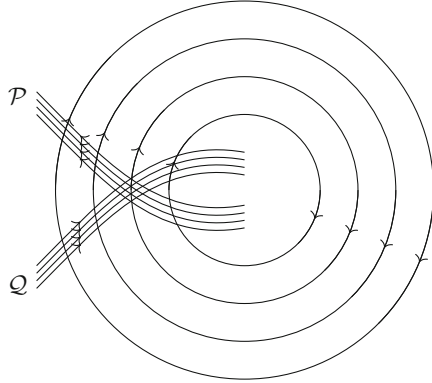
For digraphs, we need first to replace the notion of (undirected) treewidth with **directed treewidth**, introduced by Johnson, Robertson, Seymour, and Thomas [15]. Treewidth is a graph width measure that focuses on the structure of cuts in undirected graphs; directed treewidth is a graph width measure that aims at understanding the structure of cuts in a graph — but, this time, *directed* cuts. Directed treewidth and other digraph measures will be discussed in depth in Chapter 9 and hence we refrain here from providing the (quite complex) formal definition of this measure. Instead, we will work with a dual notion of well-linked sets, introduced later in this section.

Let us move to the directed counterpart of the second ingredient of the Excluded Grid Theorem: instead of a grid, we have here the **directed cylindrical grid**. A cylindrical grid is depicted in Figure 5.12. It consists of  $k$  vertex-disjoint directed cycles  $C_1, C_2, \dots, C_n$ , linked by  $2k$  vertex-disjoint paths  $P_1, Q_1, P_2, Q_2, \dots, P_k, Q_k$ . The paths  $P_i$  connect the cycles in the increasing order of indices, while the paths  $Q_i$  connect the cycles in the decreasing order of indices. Along every cycle, the order of paths seen on that cycle is  $P_1, Q_1, P_2, Q_2, \dots, P_k, Q_k$ . In 2001, Johnson, Robertson, Seymour, and Thomas conjectured that the cylindrical grid plays the role of the undirected grid as a canonical obstacle to small directed treewidth. This conjecture has only been recently proven by Kawarabayashi and Kreutzer [17]:

**Theorem 5.3.1** ([17]) *For every positive integer  $k$  there exists an integer  $f(k)$  such that every digraph of directed treewidth at least  $f(k)$  contains a cylindrical grid of order  $k$  as a (butterfly) minor.*

A digraph  $D'$  is a **butterfly minor** of  $D$  if  $D'$  can be obtained from  $D$  by means of arc and vertex deletion, as well as contraction of arcs  $e = (u, v)$  for which  $e$  is the only outgoing arc of  $u$  or the only ingoing arc of  $v$ .





**Figure 5.13** A schematic view of a relaxed cylindrical grid of order 4. Formally, the linkages  $\mathcal{P}$  and  $\mathcal{Q}$  may start and end on the extreme cycles, but we will construct them as leading between the outside and inside of the concentric cycles.

In an unpublished manuscript dating back to 2001 [16], Johnson, Robertson, Seymour, and Thomas proved the theorem for planar digraphs. Our goal in this section is to sketch the proof of this theorem, following recent work of Chekuri, Ene, and Pilipczuk [5] that applied the ideas of [16] to design an approximation algorithm for the  $k$ -DISJOINT PATHS. We will not obtain such a rigid structure as the cylindrical grid, but a relaxed one (see Figure 5.13):

**Definition 5.3.2** A relaxed cylindrical grid of order  $k$  in a digraph  $G$  embedded on a sphere consists of

- a sequence  $C_1, C_2, \dots, C_k$  of vertex-disjoint cycles arranged concentrically, that is, for every  $1 \leq i < j \leq k$ , the cycle  $C_i$  is to the left of  $C_j$ ;
- a linkage  $\mathcal{P}$  of order  $k$ , in which every path starts at a vertex on or to the left of  $C_1$ , and ends at a vertex on or to the right of  $C_k$ ;<sup>1</sup>
- a linkage  $\mathcal{Q}$  of order  $k$ , in which every path starts at a vertex on or to the right of  $C_k$  and ends at a vertex on or to the left of  $C_1$ .

In other words, in a relaxed cylindrical grid we relax the requirement that the paths  $P_i$  cannot intersect the paths  $Q_j$  and we relax the required order in which these paths intersect every cycle  $C_i$ . Note that due to the spherical embedding of the graph, every path in the linkages  $\mathcal{P}$  and  $\mathcal{Q}$  intersects every cycle  $C_i$ .

Having sacrificed the rigid structure of a cylindrical grid, we will aim at a near-linear relation between the grid size and the directed treewidth. That is, our goal is to sketch the proof of the following theorem:

**Theorem 5.3.3** ([5]) *There exists a polynomial  $p$  such that every planar digraph  $G$  of directed treewidth  $k$  contains a relaxed cylindrical grid of order at least  $k/p(\log k)$ .*

<sup>1</sup> Recall that a **linkage** is a family of pairwise vertex-disjoint paths.

In other words, the size of the obtained relaxed cylindrical grid is the same as the directed treewidth, up to polylogarithmic factors.

### 5.3.1 Well-Linked Sets

As announced at the beginning of this section, instead of directed treewidth we will work with a dual notion of a **well-linked set**. To this end, let us first recall the notion of a **separation** in a digraph  $D$ : a pair of vertex subsets  $(A, B)$  is a separation in  $D$  if  $A \cup B = V(D)$  and there is no arc with tail in  $A \setminus B$  and head in  $B \setminus A$ . The **order** of the separation  $(A, B)$  is  $|A \cap B|$ .

A set  $X \subseteq V(D)$  is **node-well-linked** in  $D$  if for any two disjoint subsets  $A, B$  of  $X$  of equal size, there exists  $|A| = |B|$  vertex-disjoint paths such that every vertex of  $A$  is a starting vertex of exactly one path, and every vertex of  $B$  is an ending vertex of exactly one path. By relaxing vertex-disjointness to arc-disjointness we obtain the notion of an **edge-well-linked set**. By Menger's theorem, a set  $X \subseteq V(D)$  is edge-well-linked if and only if for any partition  $V(D) = A \uplus B$  the number of edges in  $\delta^+(A)$  is at least  $\min\{|X \cap A|, |X \cap B|\}$ . Similarly, a set  $X \subseteq V(D)$  is node-well-linked if and only if any separation  $(A, B)$  of  $D$  has order at least  $\min\{|X \cap A|, |X \cap B|\}$ . The second equivalent notion allows us to define fractional well-linkedness: for a real  $\alpha \in [0, 1]$ , a set  $X \subseteq V(D)$  is  **$\alpha$ -edge-well-linked** if for every partition  $V(D) = A \uplus B$  we have  $|\delta^+(A)| \geq \alpha \min\{|X \cap A|, |X \cap B|\}$ , while it is  **$\alpha$ -node-well-linked** if every separation  $(A, B)$  has order at least  $\alpha \min\{|X \cap A|, |X \cap B|\}$ .

Observe that node-well-linkedness is stronger than edge-well-linkedness: any  $\alpha$ -node-well-linked set is also  $\alpha$ -edge-well-linked, while in the other direction we lose a factor proportional to the maximum degree: an  $\alpha$ -edge-well-linked set in a digraph of maximum degree  $\Delta$  is  $\alpha/\Delta$ -node-well-linked.

Johnson, Robertson, Seymour, and Thomas [15, 16] showed that the size of the largest node-well-linked set is tightly related to directed treewidth.

**Theorem 5.3.4** ([15, 16]) *Every digraph of directed treewidth  $k$  contains a node-well-linked set of size  $\Omega(k)$ , and, conversely, every digraph containing a node-well-linked set of size  $k$  has directed treewidth  $\Omega(k)$ .*

A standard tool in studying well-linked sets is the following lemma that shows that one can extract an  $\Omega(1)$ -node-well-linked set from an  $\alpha$ -node-well-linked set without losing much more than necessary. This particular statement for directed graphs is due to Chekuri and Ene [4].

**Lemma 5.3.5** ([4]) *If  $X$  is an  $\alpha$ -node-well-linked set in a digraph  $D$ , then there exists a set  $X' \subseteq X$  of size  $\Omega(\alpha|X|)$  that is  $\frac{1}{32}$ -node-well-linked in  $D$ .*

### 5.3.2 Eulerian Digraphs

A digraph is **Eulerian** if it is weakly connected and for every vertex  $v$ , the in-degree and the out-degree of  $v$  are equal. Note that in an Eulerian digraph, the maximum in-degree is equal to the maximum out-degree. We will use the following simple “balancedness” argument in Eulerian digraphs.

**Lemma 5.3.6** *Suppose  $D$  is an Eulerian digraph and  $V(D) = A \uplus B$  is a partition of the vertex set of  $D$ . Then the number of arcs of  $D$  that have tail in  $A$  and head in  $B$  is equal to the number of arcs of  $D$  that have tail in  $B$  and head in  $A$ .*

**Proof:** Since  $D$  is Eulerian, by summing the in-degrees and the out-degrees of vertices in  $A$  we infer that the number of arcs with heads in  $A$  is equal to the number of arcs with tails in  $A$ . By subtracting the number of arcs with both heads and tails in  $A$  we obtain the asserted equality.  $\square$

The critical insight of the work of Johnson, Robertson, Seymour, and Thomas [16] is that Eulerian digraphs of small maximum degree behave in some ways similarly as undirected graphs. This can be seen in the following simple lemma, used, e.g., in [5].

**Lemma 5.3.7** *Let  $A, B$  be two vertex subsets in an Eulerian digraph  $D$  of maximum in-degree  $\Delta$ , and let  $k$  be a nonnegative integer. Then, if in the underlying undirected graph there exist  $(\Delta+1)k+1$  vertex-disjoint undirected paths from  $A$  to  $B$ , then in  $D$  there exist  $k+1$  vertex-disjoint directed paths from  $A$  to  $B$ .*

**Proof:** If the conclusion is not true, then by Menger’s theorem there exists a separation  $(A', B')$  of order at most  $k$  separating  $A$  from  $B$ . That is, we have  $A' \cup B' = V(D)$ ,  $A \subseteq A'$ ,  $B \subseteq B'$ ,  $|A' \cap B'| \leq k$ , and no arc of  $D$  has its tail in  $A' \setminus B'$  and its head in  $B' \setminus A'$ . Since there are  $(\Delta+1)k+1$  undirected paths from  $A$  to  $B$ , and only  $k$  of them can pass through  $A' \cap B'$ , the remaining  $\Delta k+1$  paths need to go via arcs connecting  $A' \setminus B'$  and  $B' \setminus A'$ . Since there are no arcs with tail in  $A' \setminus B'$  and head in  $B' \setminus A'$ , we infer that there are at least  $\Delta k+1$  arcs with tail in  $B' \setminus A'$  and head in  $A' \setminus B'$ . However,  $D$  contains at most  $\Delta|A' \cap B'| \leq \Delta k$  arcs with tail in  $A' \setminus B'$  and head in  $B'$ , as every such arc needs to have its head in  $A' \cap B'$ . This is a contradiction, as by Lemma 5.3.6, the number of arcs with tail in  $A' \setminus B'$  and head in  $B'$  should be equal to the number of arcs with tail in  $B'$  and head in  $A' \setminus B'$ .  $\square$

Lemma 5.3.7 shows the surprising power of the “balancedness” argument of Lemma 5.3.6. In planar digraphs, we can exploit this argument even further, focusing on cuts represented by curves.

Let  $D$  be a digraph embedded in the plane. A curve  $\gamma$  on a sphere is in *general position* with respect to  $D$  if  $\gamma$  has a finite number of intersections

with (the embedding of)  $D$ , and whenever  $\gamma$  intersects an arc  $e$  of  $D$ , it intersects  $e$  transversally, that is, in a small neighborhood of the intersection the arc  $e$  splits  $\gamma$  into two parts lying on the opposite sides of  $e$ . Furthermore, if  $\gamma$  in general position with respect to  $D$  does not visit any vertex of  $D$ , it is called a **face-edge** curve. An *imbalance* of a curve  $\gamma$  is the difference between the number of arcs of  $D$  traversing  $\gamma$  from left to right and the number of arcs of  $D$  traversing  $\gamma$  from right to left. By Lemma 5.3.6, we have the following:

**Lemma 5.3.8** *Every closed face-edge curve  $\gamma$  with respect to an Eulerian digraph  $D$  has zero imbalance.*

### 5.3.3 Cut-Matching Game

In Theorem 5.3.3 the given digraph  $D$  may be far from being Eulerian. Quite surprisingly, we can turn  $D$  into an Eulerian digraph with small maximum degree without losing much on the directed treewidth assumption. In [16], the authors obtained constant maximum degree by elaborate structural arguments, yielding a significant toll on the final relation between directed treewidth and the size of the obtained grid. The approach of [5], originating in the techniques developed in the area of routing, is conceptually cleaner, but leads only to a polylogarithmic bound on the maximum degree.

The key idea of [5] is to use the so-called **cut-matching game** to construct an embedding. To define this game, we first need to recall the notion of an *edge expansion*:

**Definition 5.3.9** *Let  $G$  be an undirected multigraph. The **edge expansion** of a set  $S \subseteq V(G)$  is defined as the ratio*

$$\frac{|\delta(S)|}{\min\{|S|, |V(G) \setminus S|\}},$$

where  $\delta(S)$  is the set of edges with exactly one endpoint in  $S$ . The *edge expansion* of a graph is the minimum edge expansion among all sets  $S \subseteq V(G)$ .

In directed (multi)graphs, the **directed edge expansion** is defined by replacing  $\delta(S)$  with  $\delta^+(S)$ : the set of arcs with tails in  $S$  and heads outside of  $S$ .

The crucial property of digraphs with large directed edge expansion is that they contain large well-linked sets; in particular, note that the definition of edge expansion immediately implies that if  $D$  has edge expansion  $\alpha$ , then  $V(D)$  is  $\alpha$ -edge-well-linked.

The cut-matching game of Khandekar, Rao, and Vazirani [18] is played on an  $n$ -vertex multigraph  $G$  for even  $n$ , which is initially empty. In every round, the first player, called the **Cut Player**, chooses a partition  $V(G) = A \uplus B$  of the vertex set into two equal-sized sets  $A$  and  $B$ . Then, the second player, called the **Matching Player**, chooses a perfect matching between  $A$  and

$B$ , which is then added to  $G$  (which may lead to  $G$  being a multigraph). The game ends when the graph  $G$  has edge expansion at least  $\alpha$ , where  $\alpha$  is a parameter of the game. The Cut Player wants to conclude the game as quickly as possible, while the Matching Player tries to stall the game. The main result of Khandekar, Rao, and Vazirani [18] is the following:

**Theorem 5.3.10** ([18]) *For every constant  $\alpha$  there exists a randomized strategy for the Cut Player in **undirected** graphs that finishes the game in expected  $\mathcal{O}(\log^2 n)$  rounds. A single move of the strategy is computable in polynomial time.*

In the directed version of the game, the matching is oriented from  $A$  to  $B$  (i.e., every added arc has its tail in  $A$  and head in  $B$ ), and the game ends when the directed edge expansion reaches a required threshold. This variant has been analyzed by Louis [21], who proved an analogous statement:

**Theorem 5.3.11** ([21]) *For every constant  $\alpha$  there exists a randomized strategy for the Cut Player in **directed** graphs that finishes the game in expected  $\mathcal{O}(\log^2 n)$  rounds. A single move of the strategy is computable in polynomial time.*

Both Theorems 5.3.10 and 5.3.11 provide a randomized strategy, with a bound on the expected number of rounds. In this description we will henceforth ignore the randomization aspect, as it is irrelevant for the purely graph theoretical existential claims.

The strength of the cut-matching game lies in the small, only polylogarithmic, number of rounds needed for the Cut Player. Consider a digraph  $D$  with a node-well-linked set  $X$ . Without loss of generality assume that  $k := |X|$  is even (we can always drop one vertex of  $X$ ). We will play the directed version of the cut-matching game, constructing a new digraph  $D_X$  with vertex set  $X$ . For the Matching Player, let us implement the following strategy. Given a partition  $X = X_1 \uplus X_2$  into two equal-sized sets, we invoke the definition of node-well-linkedness to obtain a linkage  $\mathcal{P}(X_1, X_2)$  in  $D$  from  $X_1$  to  $X_2$ . This linkage induces a directed matching between  $X_1$  and  $X_2$ : we pair up vertices that were linked by a path in the linkage  $\mathcal{P}(X_1, X_2)$ . This matching is the response of the Matching Player for the partition  $X = X_1 \uplus X_2$ .

The result of Louis [21] shows that the Cut Player can obtain a digraph with constant directed edge expansion in  $L := \mathcal{O}(\log^2 k)$  rounds. Furthermore, we can assume that whenever the Cut Player plays a partition  $(X_1, X_2)$ , she also immediately after plays the partition  $(X_2, X_1)$ . With the above behavior of the Matching Player, we obtain a final digraph  $D_X$  of constant directed edge expansion and every vertex of  $D_X$  has in- and out-degree  $L$ . This digraph  $D_X$  naturally projects down to  $D$ , that is, we can construct a digraph  $H_X$ , starting from  $V(H_X) = V(D)$ , and for every round of the game with partition  $X = X_1 \uplus X_2$  we add the linkage  $\mathcal{P}(X_1, X_2)$  to  $H_X$ . More precisely, we add all arcs of all paths in  $\mathcal{P}(X_1, X_2)$  to  $H_X$ , duplicating

some arcs of  $D$  if necessary. In this manner, every vertex of  $H_X$  has equal in- and out-degree and these degrees are bounded by  $2L$ . Furthermore, since  $D_X$  has edge expansion  $\Omega(1)$ , we have that  $X = V(D_X)$  is  $\Omega(1)$ -edge-well-linked in  $D_X$ ; by the construction of  $H_X$ , we have that  $X$  is also  $\Omega(1)$ -edge-well-linked in  $H_X$ . By the degree bound,  $X$  is  $\Omega(1/L)$ -node-well-linked in  $H_X$ . By Lemma 5.3.5, we can find a set  $X' \subseteq X$  of size  $\Omega(|X|/L) = \Omega(k/\log^2 k)$  that is  $\frac{1}{32}$ -node-well-linked in  $H_X$ .

The following lemma summarizes the above reasoning.

**Lemma 5.3.12** *Let  $D$  be a digraph with a node-well-linked set  $X$  of size  $k$ . Then there exists an integer  $L = \mathcal{O}(\log^2 k)$  and a subgraph  $H_X$  of the graph  $D$  with every edge duplicated at most  $L$  times, such that every vertex of  $H_X$  has equal in- and out-degree, these degrees are bounded by  $L$ , and  $X$  is  $\Omega(1)$ -edge-well-linked in  $H_X$ . Furthermore, there exists a set  $X' \subseteq X$  of size  $\Omega(k/\log^2 k)$  that is  $1/32$ -node-well-linked in  $H_X$ .*

Observe that if  $D$  is planar, then so is the graph  $H_X$  given by Lemma 5.3.12.

The final observation is that in our case it is sufficient to find a relaxed cylindrical grid in  $H_X$  instead of  $D$ : a relaxed cylindrical grid in  $H_X$  projects naturally onto  $D$ , and the duplicated edges do not break the structure, as we required vertex-disjointness of both the cycles  $C_i$  and the linkages  $\mathcal{P}$  and  $\mathcal{Q}$ . Thus, by losing an  $\mathcal{O}(\log^2 k)$  factor in the size of the well-linked set  $X$ , and relaxing node-well-linkedness to  $1/32$ -node-well-linkedness, we can henceforth assume that the given graph  $D$  is Eulerian with maximum degree  $\Delta = \mathcal{O}(\log^2 k)$ .

### 5.3.4 Finding a Grid in an Eulerian Digraph

In this section we show the following:

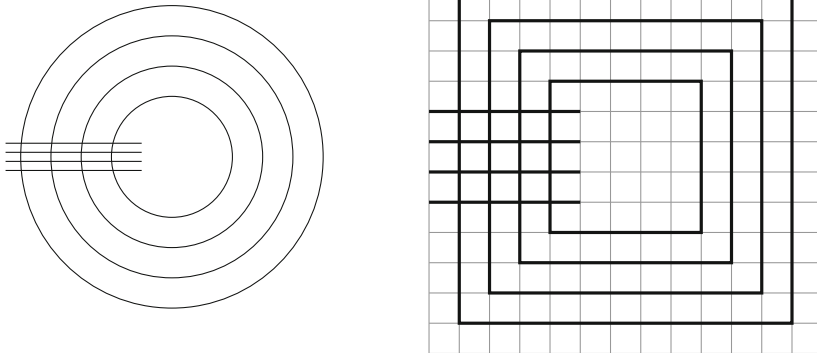
**Theorem 5.3.13** *If a planar Eulerian digraph  $D$  of maximum degree  $\Delta$  contains an  $\alpha$ -node-well-linked set  $X$  of size  $k$ , then it also contains a relaxed cylindrical grid of order  $\Omega(\alpha k/\Delta^2)$ .*

As the previous section reduced us to this case with  $\alpha = 1/32$  and  $\Delta = \mathcal{O}(\log^2 k)$ , for the proof of Theorem 5.3.3 it suffices to prove Theorem 5.3.13. We follow the exposition of [5], which builds upon the arguments of [16].

The proof of Theorem 5.3.13 heavily relies on the assumption that  $D$  is Eulerian via tools introduced in Section 5.3.2. On a very high level, we start with a large *undirected* grid in  $D$  and then argue about *directed* structures inside this grid using arguments relying on the assumption that  $D$  is Eulerian. Let  $G$  be the undirected (multi)graph underlying of  $D$ .

#### Obtaining an undirected grid

The first step is to obtain an undirected grid in  $D$ . To this end, we recall that in undirected planar graphs, a linear relation between treewidth and the largest grid minor is known [12, 25]:



**Figure 5.14** Structure obtained by Lemma 5.3.15 and how it can be found inside a sufficiently large undirected grid minor.

**Theorem 5.3.14** ([12, 25]) *A planar undirected graph of treewidth  $k$  contains a grid of sidelength  $9k/2$  as a minor.*

Note that if  $X$  is  $\alpha$ -node-well-linked in  $D$ , it is also  $\alpha$ -node-well-linked in  $G$ . Furthermore, a graph containing an  $\alpha$ -node-well-linked set of size  $k$  has treewidth  $\Omega(\alpha k)$ . As a result we obtain the following claim; see Fig. 5.14 for a pictorial proof. Recall that in the context of undirected graphs embedded on a plane, a sequence  $C_1, C_2, \dots, C_r$  of vertex-disjoint cycles is **concentric** if each cycle  $C_i$  separates the cycles  $\{C_j : j < i\}$  from the cycles  $\{C_j : j > i\}$ .

**Lemma 5.3.15** *There exists an integer  $r = \Omega(\alpha k)$  such that  $G$  contains a sequence  $C_1, C_2, \dots, C_r$  of  $r$  concentric cycles and a set of  $r$  vertex-disjoint paths connecting  $C_1$  with  $C_r$ .*

**Isles.** We now need the following notion. Given a vertex  $v \in V(G)$ , a set  $Q \subseteq V(G)$  with  $v \notin Q$ , and an integer  $\ell$ , a  $(v, Q, \ell)$ -*isle* is a set  $S \subseteq V(G)$  such that  $v \in S$ ,  $S \cap Q = \emptyset$ ,  $G[S]$  is connected, and  $|N_G(S)| \leq \ell$ . In other words,  $S$  is a connected part of the graph around  $v$  with small boundary and separated from  $Q$ .

Fix  $\ell = \Theta(r/\Delta) = \Theta(\alpha k/\Delta)$ . The constants hidden in the  $\Theta(\cdot)$  notation will be chosen in the course of the argumentation, but the reader may think that  $\ell$  is a small (but constant) fraction of  $r/\Delta$ , in particular  $\ell$  is much smaller than  $r$ . Pick a vertex  $v_1$  on the cycle  $C_1$ . Since we can assume that  $2\Delta < \ell = \Theta(\alpha k/\Delta)$  (as otherwise the statement of Theorem 5.3.13 is immediate),  $\{v_1\}$  is a  $(v_1, V(C_r), \ell)$ -isle. Let  $S_1$  be an inclusion-wise maximal  $(v_1, V(C_r), \ell)$ -isle, and let us analyze its properties.

First, since  $\ell < r$  and  $G$  contains  $r$  vertex-disjoint paths from  $C_1$  to  $C_r$ , the set  $S_1$  cannot contain the whole cycle  $C_i$  for any  $i$ . Since  $G[S_1]$  is connected and the cycles  $C_i$  are concentric,  $S_1$  is disjoint from every cycle  $C_i$  for  $i > \ell$ .

Note that  $\ell$  is much smaller than  $r$ ; the last statement shows that  $S_1$  lives locally in the graph  $G$ , and does not go deep into the set of concentric cycles  $\{C_i : 1 \leq i \leq r\}$ .

Symmetrically, we pick an arbitrary vertex  $v_r$  on  $C_r$  and define a maximal  $(v_r, V(C_1), \ell)$ -isle  $S_r$ ; we again have that  $S_r$  is disjoint from cycles  $C_i$  for  $i \leq r - \ell$ . Since we can assume that  $\ell$  is much smaller than  $r$ , the isles  $S_1$  and  $S_r$  are disjoint and separated by  $r - 2\ell$  cycles  $C_i$ .

By  $N_G^i[S]$  we denote the set of vertices within distance at most  $i$  from  $S$  in  $G$ . We have that  $N_G^2[S_1]$  does not intersect the cycle  $C_{\ell+3}$ ; by the maximality of  $S_1$ , there are  $\ell + 1$  vertex-disjoint paths connecting  $N_G^2[S_1]$  with  $C_r$ . Symmetrically, there are  $\ell + 1$  vertex-disjoint paths connecting  $N_G^2[S_r]$  and  $C_1$ . Since  $\ell$  is much smaller than  $r$ , there are many more than  $\ell$  cycles  $C_i$  for  $\ell + 3 \leq i \leq r - \ell - 2$ ; note that all these cycles are disjoint from  $N_G^2[S_1 \cup S_r]$  and separate  $S_1$  from  $S_r$ . By combining the aforementioned linkages of  $\ell + 1$  paths and these cycles, we obtain that there exists a flow of size at least  $\ell/3$  from  $N_G^2[S_1]$  to  $N_G^2[S_r]$ : just treat the linkages and cycles as flow paths each carrying a flow of  $1/3$  to avoid congestion, and combine the flow paths naively, following first the flow paths from  $N_G^2[S_1]$  to  $C_r$ , then cycles  $C_i$  for  $\ell + 3 \leq i \leq r - \ell - 2$ , and finally the flow paths from  $C_1$  to  $N_G^2[S_r]$ . By the integrality of flows, there exists a linkage in  $G$  of size at least  $\ell/3$  leading from  $N_G^2[S_1]$  to  $N_G^2[S_r]$ . A symmetric reasoning yields a linkage in  $G$  of size at least  $\ell/3$  leading from  $N_G^2[S_r]$  to  $N_G^2[S_1]$ . These linkages are undirected (in  $G$ ), but the digraph  $D$  is Eulerian: by Lemma 5.3.7, in  $D$ , there exists a (directed) linkage  $\mathcal{P}$  from  $N_G^2[S_1]$  to  $N_G^2[S_r]$  and a (directed) linkage  $\mathcal{Q}$  from  $N_G^2[S_r]$  to  $N_G^2[S_1]$ , both of size at least  $\ell/(3(\Delta + 1)) = \Theta(\alpha k/\Delta^2)$ . Note that every path in  $\mathcal{P}$  and  $\mathcal{Q}$  intersects every cycle  $C_i$  for  $\ell + 3 \leq i \leq r - \ell - 2$ . Figure 5.15 illustrates the structure obtained so far.

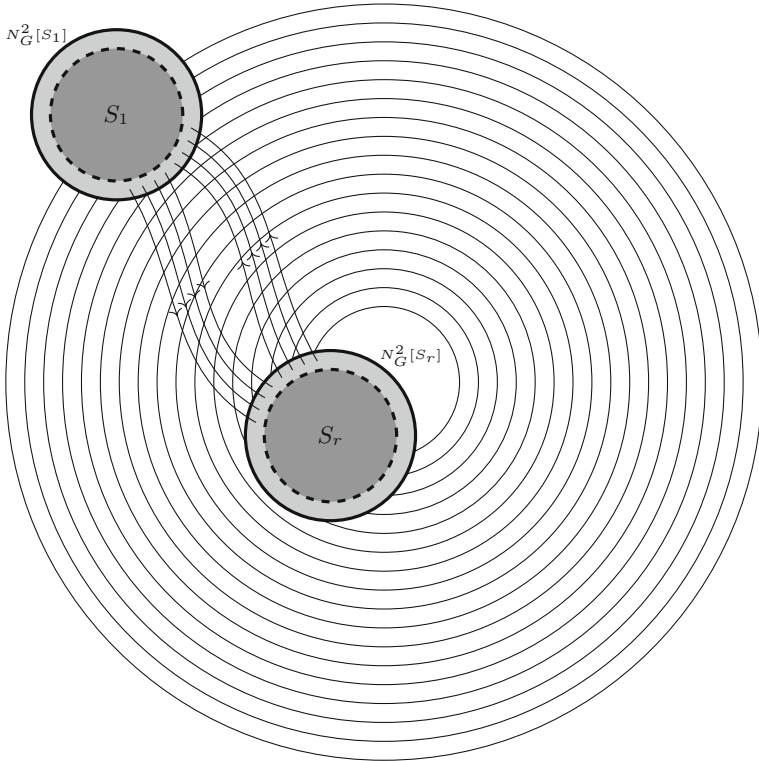
The linkages  $\mathcal{P}$  and  $\mathcal{Q}$  will form the desired linkages between the extreme cycles in the desired relaxed cylindrical grid. To conclude the construction, we need to show that there are  $\Theta(\alpha k/\Delta^2)$  concentric directed cycles with  $N_G^1[S_1]$  on one side and  $N_G^1[S_r]$  on the other side, so that they intersect every path in  $\mathcal{P} \cup \mathcal{Q}$ . To prove their existence, we use the (undirected) cycles  $C_i$ .

**Cycles.** Let  $D'$  be the digraph  $D$  with the vertices of  $N_G^1[S_1] \cup N_G^1[S_r]$  removed. Note that  $D'$  is no longer Eulerian, but it is close to being Eulerian: since  $S_1$  and  $S_r$  are isles, we have  $|N_G(S_1)|, |N_G(S_r)| \leq \ell$  and, consequently, at most  $2\ell\Delta$  arcs connect  $N_G^1[S_1] \cup N_G^1[S_r]$  with the vertices of  $D'$ .

Consider now the spherical embedding of  $D$  and the naturally induced embedding of  $D'$ . There are two distinguished faces of the embedding of  $D'$ :  $f_1$ , which contains  $S_1$  in the embedding of  $D$ , and  $f_r$ , which contains  $S_r$ . Let us try to find as many as possible vertex-disjoint directed cycles that have  $f_1$  to the left and  $f_r$  to the right.

The crucial observation is that there is a well-defined notion of a directed cycle that has  $f_1$  to the left, but is as close to  $f_1$  as possible, in the sense that it has as few faces of  $D'$  to the left as possible. To see this, consider the





**Figure 5.15** Structure obtained from isles  $S_1$  and  $S_r$ . To finish the construction, we lack sufficiently many concentric *directed* cycles separating  $S_1$  from  $S_r$ , but we have many *undirected* ones.

following procedure: mark  $f_1$  and every face of  $D'$  that is reachable from  $f_1$  via face-edge curves in  $D'$  that are crossed by the arcs of  $D'$  only from left to right. If such a curve  $\gamma$  reaches a face  $f$ , then  $\gamma$  certifies that  $f$  needs to be to the left of any cycle in  $D'$  that keeps  $f_1$  to the left; in particular, if  $f_r$  is marked, the corresponding curve shows that there is no cycle in  $D'$  that keeps  $f_1$  to the left and  $f_r$  to the right. In the other direction, it is easy to see that the boundary of the region of unmarked faces that contain  $f_r$  (if  $f_r$  is unmarked) forms the desired directed cycle.

By iterating the above argument, we can obtain the following claim:

**Lemma 5.3.16** ([5]) *For any integer  $t$ , in  $D'$  there exists either:*

1. *a family of vertex-disjoint cycles  $D_1, D_2, \dots, D_t$ , each having  $f_1$  to the left and  $f_r$  to the right;*
2. *a curve  $\gamma$  in general position with respect to  $D'$  that starts in  $f_1$ , ends in  $f_r$ , passes through at most  $t$  vertices of  $D'$ , and such that every arc of  $D'$  crossing  $\gamma$  crosses it from left to right.*

We pick  $t = \Theta(\alpha k/\Delta^2)$  and apply Lemma 5.3.16: once directly, and once with the roles of  $f_1$  and  $f_r$  swapped. If any of the application resulted in a family of  $t$  directed cycles, these cycles, together with linkages  $\mathcal{P}$  and  $\mathcal{Q}$ , form the desired relaxed grid. Thus, we are left with the case when both applications returned a curve; note that we may assume without loss of generality that each of these curves is without self-intersections. By joining these curves together inside  $f_1$  and  $f_r$ , we obtain a closed curve  $\gamma_0$  in general position with respect to  $D'$  that intersects at most  $2t$  vertices and every arc crossing  $\gamma_0$  crosses it from left to right. We modify  $\gamma_0$  slightly as follows: whenever  $\gamma_0$  visits a vertex  $v$  we move it a little so that it intersects a number of arcs incident with  $v$  instead. In this manner, the obtained curve  $\gamma$  is a closed face-edge curve in  $D'$  that visits both  $f_1$  and  $f_r$ , does not visit any vertex of  $D'$ , and at most  $2t\Delta$  arcs intersecting  $\gamma$  cross it from right to left.

However,  $\gamma$  needs to cross every cycle  $C_i$  for  $\ell + 3 \leq i \leq r - \ell - 2$ ; by taking  $\ell$  to be sufficiently small compared to  $r$ , there are at least  $r/2 = \Theta(\alpha k)$  such cycles. Since  $2t\Delta = \Theta(\alpha k/\Delta)$ , the absolute value of the imbalance of the curve  $\gamma$  can be assumed to be at least  $r/4$ .

Consider now a digraph  $D''$ , obtained similarly as  $D'$  from  $D$ , but instead of removing  $N_G^1[S_1]$ , we contract it onto a single vertex  $w_1$ , similarly we also contract  $N_G^1[S_r]$  onto a new vertex  $w_r$ . Any loops thus created at  $w_1$  or  $w_r$  are removed. Note that  $D''$  remains Eulerian and the degree of  $w_1$  and  $w_r$  is at most  $\ell\Delta$  in  $D''$ . Furthermore, by slight modifications of  $\gamma$  inside  $f_1$  and  $f_r$ , we may assume that  $\gamma$  is in general position with respect to  $D''$  as well, visits neither  $w_1$  nor  $w_r$ , and crosses every arc incident to these two vertices at most once (they are drawn inside  $f_1$  and  $f_r$ , where we can freely manipulate  $\gamma$ ). However, now  $\gamma$  is a closed curve in general position with respect to an Eulerian digraph  $D''$ , and thus has zero imbalance. Recall that  $D'$  and  $D''$  differ on at most  $2\ell\Delta$  edges, each crossed by  $\gamma$  at most once. By picking a sufficiently small constant in the definition of  $\ell = \Theta(r/\Delta)$  we obtain  $2\ell\Delta < r/4$ , yielding a contradiction.

Thus, at least one application of Lemma 5.3.16 resulted in a family of cycles, giving the final ingredient of the relaxed cylindrical grid, and concluding the proofs of Theorems 5.3.13 and 5.3.3.

### 5.3.5 Perspective

Theorem 5.3.3 shows that if one relaxes the structure of the cylindrical grid to allow intersections of the radial linkages, we can obtain good (up to poly-logarithmic factors) dependency between directed treewidth and the size of the grid. This resembles the situation from undirected graphs, where linear dependency between treewidth and the size of largest grid minor gave rise to multiple algorithmic applications through the theory of bidimensionality [10].

In the context of routing, the above theorem fits into a more general approach for designing approximation algorithms for the  $k$ -DISJOINT PATHS problem, pioneered by Chekuri, Khanna, and Shepherd [3]. This approach

turned out to be very successful in the context of undirected graphs, leading to a poly-logarithmic approximation with congestion 2 for the edge-disjoint version of  $k$ -DISJOINT PATHS by Chuzhoy and Li [6].

The first step is to decompose the input instance into a number of subinstances where in each subinstance the set of terminals is (fractionally) well-linked. This well-linkedness in turn allows us to reason about the existence of a good **crossbar**, a grid-like routing structure. The well-linkedness also implies the existence of a large flow between the terminals and the crossbar; an approximate solution is formed by these flow paths, joined together inside the crossbar in a way respecting the terminal pairs.

The crucial ingredient in this approach is to prove the existence of a crossbar in the presence of a large well-linked set; if the approximation factor is to be poly-logarithmic, the ratio between the size of the well-linked set and the size of the crossbar needs to be poly-logarithmic as well. The presented theorem serves as such an ingredient in the context of planar digraphs.

Apart from the context of routing [5], we do not know any other applications of Theorem 5.3.3. Furthermore, a number of questions regarding generalizations appear:

1. Can we reduce the upper bound on the maximum degree to constant, as opposed to poly-logarithmic, with only a poly-logarithmic loss on the directed treewidth? The cut-matching game approach has an inherent  $\mathcal{O}(\log^2 k)$  factor due to the number of rounds, while the arguments of [16] lead to a maximum degree of 6, but give a much worse parameter dependency.
2. Can we conduct the final part of the proof of [16], that is, obtain a regular cylindrical grid from a relaxed one, with only a poly-logarithmic loss on the size? Such an improvement may be needed if one wants to lower the allowed congestion in the approximation algorithm of [5].
3. Can we generalize these developments to other sparse graph classes? In undirected graphs, many results in the theory of bidimensionality hold in apex-minor-free or general proper minor-closed graph classes.

We remark here that the first part of the proof, which leads to an Eulerian digraph with a poly-logarithmic maximum degree and is based on the cut-matching game, works in general graphs; that is, this part does not require the planarity assumption. On the other hand, the second part of the reasoning seems to crucially depend on the topological structure of the digraph.

The existence of a large (relaxed) directed grid in the presence of a large well-linked set is also related to the Erdős–Pósa property of cycles. In undirected graphs, the classic result of Erdős and Pósa [8] asserts that if a graph does not contain  $k$  vertex-disjoint cycles, it admits a set of  $\mathcal{O}(k \log k)$  vertices that intersect every cycle. For directed graphs, a similar relation has been conjectured by Younger [30]; the conjecture was confirmed in 1996 by Reed, Robertson, Seymour, and Thomas [23]. However, the relation between the number of vertex-disjoint cycles and the size of the hitting set is not explicit

in [23] and at least exponential. Improving this relation to, say, polynomial remains widely open. More discussion on various aspects of the Erdős–Pósa property in directed graphs can be found in Section 9.5.3

Apart from the above questions, a number of very important questions remain regarding the Directed Grid Theorem in the general setting, where the proof of Kawarabayashi and Kreutzer [17] gives only a very weak parameter dependency. A discussion on these issues can be found in Chapter 9.

## References

1. I. Adler, S.G. Kolliopoulos, P.K. Krause, D. Lokshtanov, S. Saurabh, and D. M. Thilikos. Irrelevant vertices for the planar Disjoint Paths problem. *J. Combin. Theory Ser. B*, 122:815–843, 2017.
2. G. Borradaile and P.N. Klein. An  $O(n \log n)$  algorithm for maximum  $st$ -flow in a directed planar graph. *J. Assoc. Comput. Mach.*, 56(2), 2009.
3. C. Chekuri and A. Ene. The all-or-nothing flow problem in directed graphs with symmetric demand pairs. *Math. Program.*, 154(1-2):249–272, 2015.
4. C. Chekuri, A. Ene, and M. Pilipczuk. Constant congestion routing of symmetric demands in planar directed graphs. In I. Chatzigiannakis, M. Mitzenmacher, Y. Rabani, and D. Sangiorgi, editors, *ICALP 2016: The 43rd International Colloquium on Automata, Languages, and Programming*, volume 55 of *LIPICs*, pages 7:1–7:14, 2016.
5. C. Chekuri, S. Khanna, and F.B. Shepherd. Multicommodity flow, well-linked terminals, and routing problems. In H. N. Gabow and R. Fagin, editors, *STOC 2005: 37th Annual ACM Symposium on Theory of Computing*, pages 183–192. ACM, 2005.
6. J. Chuzhoy and S. Li. A polylogarithmic approximation algorithm for edge-disjoint paths with congestion 2. *J. Assoc. Comput. Mach.*, 63(5):45:1–45:51, 2016.
7. M. Cygan, D. Marx, M. Pilipczuk, and M. Pilipczuk. The planar directed  $k$ -Vertex-Disjoint Paths problem is fixed-parameter tractable. In *FOCS 2013: 54th Annual IEEE Symposium on Foundations of Computer Science*, pages 197–206. IEEE Computer Society, 2013.
8. P. Erdős and L. Pósa. On independent circuits contained in a graph. *Canadian J. Math.*, 17:347–352, 1965.
9. J. Erickson. Maximum flows and parametric shortest paths in planar graphs. In M. Charikar, editor, *SODA 2010: 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 794–804. SIAM, 2010.
10. F.V. Fomin, E.D. Demaine, M. Hajiaghayi, and D.M. Thilikos. Bidimensionality. In *Encyclopedia of Algorithms*, pages 203–207. Springer, 2016.
11. S. Fortune, J.E. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theor. Comput. Sci.*, 10:111–121, 1980.
12. Q.-P. Gu and H. Tamaki. Improved bounds on the planar branchwidth with respect to the largest grid minor size. *Algorithmica*, 64(3):416–453, 2012.
13. R. Hassin. Maximum flow in  $(s, t)$  planar networks. *Inform. Process. Lett.*, 13(3):107, 1981.
14. M.R. Henzinger, P.N. Klein, S. Rao, and S. Subramanian. Faster shortest-path algorithms for planar graphs. *J. Comput. Syst. Sci.*, 55(1):3–23, 1997.
15. T. Johnson, N. Robertson, P.D. Seymour, and R. Thomas. Directed Tree-Width. *J. Combin. Theory Ser. B*, 82(1):138–154, 2001.

16. T. Johnson, N. Robertson, P. D. Seymour, and R. Thomas. Excluding a grid minor in planar digraphs. *CoRR*, [arXiv:1510.00473](https://arxiv.org/abs/1510.00473), 2015.
17. 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.
18. R. Khandekar, S. Rao, and U.V. Vazirani. Graph partitioning using single commodity flows. *J. ACM*, 56(4), 2009.
19. P.N. Klein and S. Mozes. *Optimization Algorithms for Planar Graphs*. Available at <http://planarity.org>, 2016.
20. D. Lokshantov, D. Marx, and S. Saurabh. Lower bounds based on the exponential time hypothesis. *Bulletin of the EATCS*, 105:41–72, 2011.
21. A. Louis. Cut-matching games on directed graphs. *CoRR*, [arXiv:1010.1047](https://arxiv.org/abs/1010.1047), 2010.
22. B. Mohar and C. Thomassen. *Graphs on Surfaces*. Johns Hopkins University Press, 2001.
23. B. Reed, N. Robertson, P.D. Seymour, and R. Thomas. Packing directed circuits. *Combinatorica*, 16(4):535–554, 1996.
24. N. Robertson and P.D. Seymour. Graph minors. XIII: The disjoint paths problem. *J. Combin. Theory Ser. B*, 63:65–110, 1995.
25. N. Robertson, P.D. Seymour, and R. Thomas. Quickly excluding a planar graph. *J. Combin. Theory Ser. B*, 62(2):323–348, 1994.
26. A. Schrijver. Finding  $k$  disjoint paths in a directed planar graph. *SIAM J. Comput.*, 23(4):780–788, 1994.
27. A. Schrijver. Free partially commutative groups, cohomology, and paths and circuits in directed graphs on surfaces. Preprint, available at <http://homepages.cwi.nl/lex/files/dppg.pdf>.
28. D.D. Sleator and R.E. Tarjan. A data structure for dynamic trees. *J. Comput. Syst. Sci.*, 26(3):362–391, 1983.
29. 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.
30. D.H. Younger. Graphs with interlinked directed circuits,. In *Midwest symposium on circuit theory 2*, pages XVI 2.1–XVI 2.7, 1973.