Uncrossed Multiflows and Applications to Disjoint Paths

Chandra Chekuri[1], Guyslain Naves[2], Joseph Poremba[3], and F. Bruce Shepherd[3]

[1]Department of Computer Science, University of Illinois, Urbana-Champaign
[2]Laboratoire d'Informatique et des Systèmes, Aix-Marseille Université
[3]Department of Computer Science, University of British Columbia

Abstract

A multiflow in a planar graph is *uncrossed* if the curves identified by its support paths do not cross in the plane. Such flows have played a role in previous routing algorithms, including Schrijver's Homotopy Method and unsplittable flows in directed planar single-source instances. Recently uncrossed flows have played a key role in approximation algorithms for maximum disjoint paths in fully-planar instances, where the combined supply plus demand graph is planar. In the fully-planar case, any fractional multiflow can be converted into one that is uncrossed, which is then exploited to find a good rounding of the fractional solution. We investigate finding an uncrossed multiflow as a standalone algorithmic problem in general planar instances (not necessarily fully-planar). We consider both a congestion model where the given demands must all be routed, and a maximization model where the goal is to pack as much flow in the supply graph as possible (not necessarily equitably).

For the congestion model, we show that determining if an instance has an uncrossed (fractional) multiflow is NP-hard, but the problem of finding an integral uncrossed flow is polytime solvable if the demands span a bounded number of faces. For the maximization model, we present a strong (almost polynomial) inapproximability result. Regarding integrality gaps, for maximization we show that an uncrossed multiflow in a planar instance can always be rounded to an integral multiflow with a constant fraction of the original value. This holds in both the edge-capacitated and node-capacitated settings, and generalizes earlier bounds for fully-planar instances. In the congestion model, given an uncrossed fractional multiflow, we give a rounding procedure that produces an integral multiflow with edge congestion 2, which can be made unsplittable with an additional additive error of the maximum demand.

1 Introduction

In multicommodity flow problems, we are given a capacitated supply graph G and a demand graph H on the same node set V (whose edges are called *demands* or *commodities*, and the ends of which are called *terminals*). We seek to route flow for the commodities simultaneously, while respecting the capacities of the supply graph. The objective may be to simply maximize the total amount of flow within the capacities (the *maximization* setting), or to route for each edge $h \in E(H)$ a specific target amount $d(h)$ of flow (the *congestion* setting). Both the fractional and integer versions of these problems are fundamental in combinatorial optimization. Much recent work has been undertaken to understand these problems in specific settings, such as when the supply graph is planar.

A (fractional) multiflow[1] in a planar graph is said to be *uncrossed* if its support paths do not cross in the plane. Such multiflows have played an important role in designing algorithms for routing in planar graphs, and graphs embedded on surfaces. The most far-reaching is Schrijver's Homotopy Method [21] for node-disjoint paths. This approach fixes a homotopy and then algorithmically shifts paths in the plane until they 'obey' the homotopy, or it is shown that this is not possible using node-disjoint paths. The shifting algorithm itself runs in polytime for a fixed homotopy. If demands lie on a bounded number of faces, only a polynomially bounded number of homotopies need to be considered; hence the overall algorithm is polytime.

[1]We adopt the convention that multiflows are fractional and explicitly say when we want integrality.

This method can be applied to other problems such as finding induced cycles between a fixed node pair s, t in a planar graph [15] and finding node-disjoint directed s, t and t, s paths in planar digraphs [23] (the edge version of this problem remains open cf. `https://assert-false.science/callcc/guyslain/works/round-trip-problem`).

Another setting where uncrossed flows appear is for *fully-planar* instances, where the joint supply-demand graph $G + H$ is planar, as in Figure 1. Note that when we mean G is planar but H can be arbitrary, we say the instance is *planar* rather than fully-planar. In [14], it is shown how to construct uncrossed fractional routings for fully-planar instances in polynomial time. This has recently been used to give constant-factor approximation algorithms for *maximum edge-disjoint paths problem* (MEDP) [10] and *maximum node-disjoint paths problems* (MNDP) [20, 19], which are NP-hard even for fully-planar instances. These approximation techniques using uncrossed flows have been generalized to when $G+H$ is embedded on a higher-genus surface [13, 20]. Uncrossed flows can also be found when H is single-source, which has been used in recent work on planar single-source unsplittable flow in directed graphs [25].

Figure 1: A fully-planar instance with two different multiflows routing its demands. Dashed edges represent demands, and coloured paths are flow paths. The flows paths on the left cross, while the multiflow on the right is uncrossed.

The aforementioned results use uncrossed flows as an intermediate step in a "relax-and-round" approach for approximating MEDP/MNDP: relax the integer problems to their fractional versions, compute uncrossed solutions to the fractional problems, and then round those uncrossed solutions to integer solutions. In fully-planar and single-source instances, any multiflow can be converted into one that is uncrossed, which is not always the case when G is planar. However, the family of "uncrossable" instances is larger than just fully-planar and single-source instances. In Section 2.2 we introduce the class of *pairwise-planar* instances, and an interesting subcategory we call *series-compliant*, for which we can also uncross any given multiflow.

In this paper, we consider uncrossed multiflow as an algorithmic model in its own right for general planar instances (not necessarily fully-planar). Motivated by the success of uncrossed flows for relax-and-round applications in specific instance classes, we have two primary questions. First, given a general planar instance, can we determine if there exists an uncrossed (fractional) multiflow that routes all of the demands (for the congestion setting), or compute the uncrossed solution whose value is largest (for the maximization setting)? Second, given a fractional uncrossed multiflow in a general planar instance, can we round it to an integer solution that is "close" to the fractional, either in terms of capacity utilization (for congestion) or total value (for maximization)? We emphasize that the rounding arguments used for fully-planar instances (e.g., in [10, 20]) do not directly apply for general planar instances, since they use additional structure from the fact that demands are non-crossing and lie within faces, which we do not necessarily have. For general uncrossed flows the endpoints of paths could be far apart and the demands could cross.

In the process of answering these questions, we also expand on the theory of uncrossed flows by examining different categories of uncrossed flows and their properties, as well as identifying several standard operations and assumptions in network flows that do not work for the uncrossed multiflow model. As a preview example, we argue that a robust model of uncrossed multiflow must allow for flow to be routed on walks that can repeat vertices and edges rather than just simple paths as is normally done for flows.

1.1 Problem Settings and Main Results

In what follows, when we discuss a planar graph, we generally consider it coming with a particular embedding on the surface (however, at the end of this subsection we remark on the flexibility of re-embedding the graph). All graphs are undirected.

Problem settings. We consider two settings for multicommodity flow: *maximization* and *minimum congestion*. In the maximization setting, each edge $e \in E(G)$ has an integer capacity $u(e) \geq 0$, or each node $v \in V(G)$ has an integer capacity $u(v) \geq 0$. A *maximization instance* thus is a triple (G, H, u). The objective is to pack the maximum total amount of flow onto paths whose endpoints are edges in H. Letting $\mathcal{P}_{st}$ denote the set of st-paths in G, and $\mathcal{P} = \bigcup_{st \in E(H)} \mathcal{P}_{st}$, the associated linear programs are:

$$\max_f \sum_{P \in \mathcal{P}} f(P) \quad \text{s.t.} \quad \begin{cases} \sum_{P \in \mathcal{P}: e \in P} f(P) \leq u(e) & \forall e \in E(G) \\ f \geq 0 \end{cases} \tag{1}$$

$$\max_f \sum_{P \in \mathcal{P}} f(P) \quad \text{s.t.} \quad \begin{cases} \sum_{P \in \mathcal{P}: v \in P} f(P) \leq u(v) & \forall v \in V(G) \\ f \geq 0 \end{cases} \tag{2}$$

Note that we do not require $\sum_{P \in \mathcal{P}_{st}} f(P) \leq 1$ for each $st \in E(H)$. The integer program versions, (1) and (2) are called *maximum edge-disjoint paths* (MEDP) and *maximum node-disjoint paths* (MNDP) respectively when the capacities are unit-valued.

In the minimum congestion setting, each $e \in E(G)$ has integer capacity $u(e) \geq 0$ and each $h \in E(H)$ has an integer value $d(h) \geq 0$ attached, and the flow must satisfy the demand values. A *congestion instance* thus is a tuple (G, H, u, d). The goal is to determine if the following program has a solution, or to increase u by as little as possible (e.g. multiplying by a small factor) so that a solution can be found that routes all demands.

$$\text{Find } f \text{ s.t.} \quad \begin{cases} \sum_{P \in \mathcal{P}_{st}} f(P) \geq d(st) & \forall st \in E(H) \\ \sum_{P \in \mathcal{P}: e \in P} f(P) \leq u(e) & \forall e \in E(G) \\ f \geq 0 \end{cases} \tag{3}$$

Main results. We are interesting in solving modified versions of (1), (2), and (3) where we add an additional constraint that the multiflow is uncrossed. We call these the *U-constrained* versions of these problems. We emphasize that fractional solutions are allowed. Recall that for fully-planar instances, adding this constraint does not affect the optimal value of (1), (2) or the feasibility of (3), but in general planar instances this constraint can affect the optimal value/feasibility. Additionally, we seek to understand whether we can use (fractional) uncrossed solutions to recover good integer solutions for (1), (2), and (3).

We begin with our integral rounding results, starting with maximization.

Theorem 1.1. *If G is planar, given a feasible uncrossed solution f to LP (1) or LP (2), there exists a feasible integral solution to the corresponding LP that achieves an $\Omega(1)$ fraction of the value of f. For unit capacities, this solution can be computed in polynomial time.*

We prove this result by establishing an "approximate integer decomposition property" for polyhedron of uncrossed multiflows. This is proved by exploiting colouring results on a restricted type of string graphs (path intersection graph). In fact, integer decomposition also implies $\Omega(1)$ rounding holds when weights are attached to the problem; that is, each $st \in E(H)$ has an associated per-unit profit $w(st)$, and the objective to the LPs is instead $\max_f \sum_{st \in E(H)} \sum_{P \in P_{st}} w(st) f(P)$.

For the minimum congestion setting, we give a factor 2 rounding for converting a fractional uncrossed multiflow into an integral flow. Moreover, if we accept an extra additive penalty, we can round to a flow that is *unsplittable*, which means each commodity is routed only on a single path.

Theorem 1.2. *If G is planar, given a feasible strongly[2] uncrossed solution to (3), there is a polynomial time algorithm to compute both:*

[2]We defer the more technical definition of strongly uncrossed to Section 2.2.

1. *an integral multiflow routing all demands where the flow on each $e \in E(G)$ is at most $2u(e)$, and*
2. *an unsplittable multiflow routing all demands where the flow on each $e \in E(G)$ is strictly less than $2u(e) + d_{\max}$, where $d_{\max} = \max_{h \in E(H)} d(h)$.*

For the algorithmic question of solving the U-constrained programs, even though they may feel like linear optimization problems, the requirement of uncrossed flow paths makes these computationally much more challenging. Interestingly, the positive result of Theorem 1.1 can be combined with results of [6, 7] to establish strong almost-polynomial inapproximability for maximum uncrossed fractional multiflow. This is in strong contrast to the case of fully-planar instances, where the U-constrained programs can be solved exactly by "uncrossing" any fractional solution, or by a combinatorial algorithm [14] for the edge-capacitated problems.

Theorem 1.3. *For planar G, the U-constrained versions of (1) and (2) are not approximable to within $O(2^{\Omega(\log^{1-\epsilon} n)})$ for any $\epsilon > 0$ assuming that $NP \not\subseteq DTIME(n^{polylog\ n})$. Moreover, there is no $O(n^{O\left(\frac{1}{(\log \log n)^2}\right)})$-approximation polynomial-time algorithm assuming that for some constant $\delta > 0$, $NP \not\subseteq DTIME(2^{n^\delta})$. These results hold even when G is maximum degree 3 (or even if G is a wall graph) and the capacities are unit.*

For the congestion setting, we prove that the U-constrained version of (3) is NP-complete.

Theorem 1.4. *For planar G, it is NP-complete to determine whether the U-constrained version of (3) has a feasible solution, even with unit-valued capacities and demands.*

We remark that since the U-constrained problem is not a linear program anymore, it is not even obvious that the problem is in NP, however we establish this (even if the capacities and demands are not unit-valued) in Section 2.3.

On the positive side, we show that finding an integral uncrossed solution to (3) can be reduced to a node-disjoint paths instance. Hence we may find uncrossed solutions if H is incident with a bounded number of faces using a result of Schrijver [22].

Theorem 1.5. *For instances where G is planar, the edges of H are incident with a bounded number of faces, and the capacities and demands are unit-valued, there is a polynomial time algorithm to find an integral uncrossed solution to (3) or declare that none exists.*

Remark on re-embedding the supply graph. Whether a multiflow is uncrossed depends on the particular embedding of G, so one may wonder if the hardness results of Theorems 1.3 and 1.4 rely on a maliciously chosen embedding. This is not the case. We show that the problem of determining whether there exists an embedding of G such that U-constrained (3) has a feasible solution is NP-hard. The flexible-embedding version of U-constrained (1) and (2) would be to maximize the objective over all possible embeddings of G. This only potentially increases the optimal value closer to that of MEDP and MNDP, and ultimately makes the problem computationally at least as hard as the fixed-embedding version.

1.2 Organization of the Paper

Section 2 lays the groundwork for the theory of uncrossed flows, gives a more detailed definition of uncrossed flows, discusses challenges with standard flow operations in the uncrossed flow setting, and presents different useful subsets of uncrossed flows and "uncrossable" instances. Section 3 covers our results for maximization, both rounding (Theorem 1.1) and inapproximability (Theorem 1.3). The rounding result for the congestion setting (Theorem 1.2) appears in Section 4. For length reasons, the NP-hardness result (Theorem 1.4) is deferred to the full version/the appendix (Section B). Section 5 contains the algorithm to solve integral congestion instances when demands are incident with a bounded number of faces (Theorem 1.5).

1.3 Related Work

While we have shown certain intractability results for uncrossed flows, we have discussed subclasses where this relax-and-round method works well. For example, recent results for unsplittable flow [25] rely on the fact that planar single-source instances are uncrossable. Most related to this work, uncrossed multiflows have been used recently to give a polytime constant factor approximation algorithm for maximum disjoint path problems in fully-planar instances (i.e. $G + H$ is planar), first with edge-disjoint paths in [13, 10], and then extended to the the node-disjoint setting in [20, 19]. In fully-planar instances, one can take the uncrossed flow paths and join them with their respective demand edges to obtain an uncrossed (laminar) family of cycles in a planar graph. This contrasts with our uncrossed flows, where the endpoints of demands need not share a common face or may cross inside faces. In fact some instances that admit uncrossed flows in the plane may have $G+H$ with arbitrarily high genus. Results in [20] use the context of uncrossed cycle families, as defined by Goemans and Williamson [12]. As with our notion of uncrossability, this is a more general but less tractable model. The definition is combinatorial rather than topological. The approximation algorithm of [20] also extends to uncrossable families of cycles in higher-genus surfaces, with the approximation ratio scaling with the genus.

Regarding the actual rounding techniques, in [10] the initial flow is first rounded to a half-integral flow using tools from linear programming theory (namely network matrices and total unimodularity), then this half-integral flow is rounded to an integral flow by using a 4-colouring of the (planar) intersection graph of the cycles. To extend to the node-disjoint setting, [20] uses a different argument, iteratively constructing a high-value solution by showing one can always find an "efficient" cycle that only intersects the other cycles on a small set of its nodes. Our rounding technique for Theorem 1.1 instead uses an approach based on establishing an "approximate integer decomposition property" [3, 4]. A key ingredient comes from looking at a subset of *string graphs*, which are the intersection graphs from curves. Specifically, we are interested in those that arise from curves that do not cross, studied for example in [8], and for which an important colouring result was proven by Fox and Pach [9][3].

2 Uncrossed Paths and Multiflows

All graphs are undirected in this paper. We always assume we have a fixed embedding of a planar graph $G = (V, E)$. We emphasize that $G + H$ is not necessarily planar. By *path* we always mean simple path. A *trail* may repeat nodes (but not edges), and a *walk* may repeat edges and nodes.

2.1 Uncrossed Paths and Multiflows

First, we define what it means for paths to cross. Edge-disjoint paths can cross at a common node, but in general we need to consider common subpaths (see Figure 2).

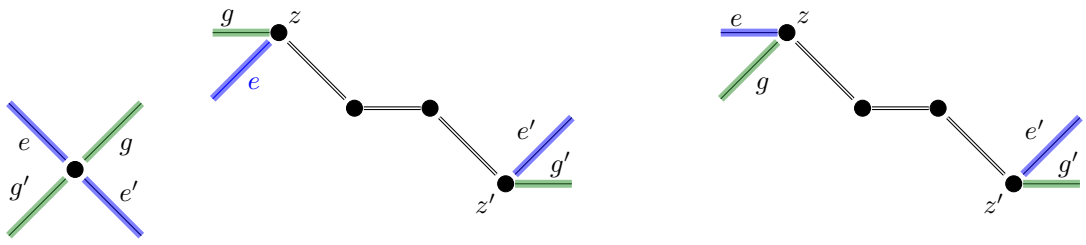

Figure 2: (Left) Two edge-disjoint paths that cross at a common node. (Center) Two paths that cross at a common subpath. (Right) Two paths that share a common subpath, but are uncrossed. Uncoloured edges are shared between the two paths.

[3]This manuscript is unpublished, but the result is referenced in [26] and their writeup was shared with us [18].

Definition 2.1 (Crossing)**.** Let Z be a maximal shared subpath of paths P and Q that does not include any of their endpoints. Say the ends of Z are z and z'. The edges of $(P \cup Q) \setminus Z$ incident with z or z' have a clockwise ordering about Z in the plane. The paths *cross* at Z if the edges of P and Q alternate clockwise about Z. That is, if $P = P_1, e, Z, e', P_2$ and $Q = Q_1, g, Z, g', P_2$, the edges occur e, g, e', g' or e, g', e', g clockwise about Z. Two paths are *uncrossed* if they do not cross on any such subpath.

The *support paths* of a multiflow f are the paths with $f(P) > 0$. The following is the natural extension of "uncrossed" to the multiflow setting (though, we will discuss a stronger uncrossing condition later).

Definition 2.2 (Uncrossed Multiflow)**.** A multiflow is *uncrossed* if its family of support paths is pairwise uncrossed.

There are more intuitive ways to think about uncrossed paths that we frequently use.

Widened Graph: Suppose we "widen" G, making its nodes into small disks and its edges into small channels. A family of pairwise uncrossed paths can have its paths drawn as curves simultaneously, each curve following the same nodes and edges as the corresponding path, so that the curves are disjoint.

Parallelizations: To avoid dealing with the formal topology of widened graphs, we use an equivalent notion based on "separating" the paths at each edge. Consider a family of paths $\mathcal{P}$ and for each $e \in E(G)$ let $\mathcal{P}_e = \{P \in \mathcal{P} : e \in E(P)\}$. Let G' be obtained by replacing each e with $|\mathcal{P}_e|$ parallel copies of itself. Intuitively, the parallel edges represent the curves within the channel of e.

Figure 3: (Left) An uncrossed parallelization of the uncrossed paths from Figure 2. (Right) Disk-expansion of the node z after uncrossed parallelization.

Now, for each $e \in E(G)$ we decide on some one-to-one mapping ϕ_e from $\mathcal{P}_e$ to the parallel class of e in G'. For each $P \in \mathcal{P}$, we obtain a path P_ϕ in G' whose edges are $\{\phi_e(P) : e \in P\}$. This induces a family of edge-disjoint paths $\mathcal{P}_\phi$ in G'. We call $\mathcal{P}_\phi$ a *parallelization* of $\mathcal{P}$. If we (arbitrarily) orient each $e \in E(G)$, it gives a linear ordering $\leq_e$ of $\mathcal{P}_e$. Different choices for $(\phi_e : e \in E(G))$ yield different parallelizations. The family $\mathcal{P}$ is pairwise uncrossed if and only if there is some parallelization that is pairwise uncrossed at every node of G' (rather than needing to consider longer shared subpaths). Using parallelizations helps us simplify subsequent definitions by assuming paths are edge-disjoint. Given a collection of paths, one can compute in polynomial time an uncrossed parallelization or find a crossing pair of paths.

Disk-Expansion: After parallelization, we sometimes go a step closer to the widened graph idea by "expanding" a node v. We draw a small disk with v at its centre (small enough that its boundary intersects only $\delta(v)$, and hits each edge once). Where an edge copy intersects the boundary, we place a new node. We terminate the edges at these new nodes. For an uncrossed parallelization, it is then possible to (simultaneously) draw edges in the open interior of the disk between nodes that correspond to consecutive edges of the same path, such that the drawing is planar. An example is shown in Figure 3.

We abuse notation and use the same symbol for a path P of G and the path corresponding to it after parallelization and disk-expansion. We also frequently identify a path with its associated curve in the plane.

2.2 Uncrossable Classes

Here we review some classes of multiflow instances that are "uncrossable". We also discuss some standard disjoint path "tricks" which do not always work in the context of uncrossed flows.

We say a multiflow instance is *uncrossable* if whenever there is a feasible flow routing particular demands, there exists an uncrossed multiflow routing the same demands (hence, it applies for both the congestion and maximization settings). Planar multiflow instances are not always uncrossable (see Figure 5), but several classes of instances are uncrossable. As mentioned before, fully-planar instances ($G + H$ is planar) are uncrossable. Other well-known uncrossable families are single-commodity and single-source instances (i.e. H is a star) for arbitrary planar G. Another class is when G is a cycle but H is arbitrary, which we call *ring* instances - in this case, two paths within G never cross.

We introduce a new class we call *pairwise-planar* instances that generalizes fully-planar. In such an instance, we have an embedding of G such that each $h \in E(H)$ has both endpoints in a face of G (not necessarily the same face for different demands), and moreover for any pair $h_1, h_2 \in E(H)$, h_1 and h_2 can be embedded inside (possibly different) faces of G so that they do not cross each other. This is more general than fully-planar instances, because we do not require all demands to be simultaneously embedded so they do not cross. Figure 4 shows a simple example. We prove that these are uncrossable in Section A, and also describe an interesting subcategory of these that we call *series-compliant* instances, which generalizes rings. Informally, in such instances the graph G is a series-parallel graph, and whenever in its series-parallel decomposition we have a series composition of k graphs $G_1, \ldots, G_k$, we allow H to contain arbitrary demand edges between the $k+1$ sources and sinks of $G_1, \ldots, G_k$, even if those demand edges would cross (thus, $G+H$ can have arbitrarily high genus). These capture most of the complexity of the *fully-compliant* instances from [5]. Combining results in [5] with our uncrossing argument, one can show that if a series-compliant instance (with integral demands) has a fractional flow and satisfies the Euler condition[4], then it has an integral *uncrossed* flow routing the same demands; this was not previously known. Other interesting members of the pairwise-planar family include the *odd spindles* [5, 1], which are special in that they are not cut-sufficient. That is, the max-flow min-cut property does not hold. Contrast this with fully-planar instances, which are cut-sufficient by a result of Seymour [24].

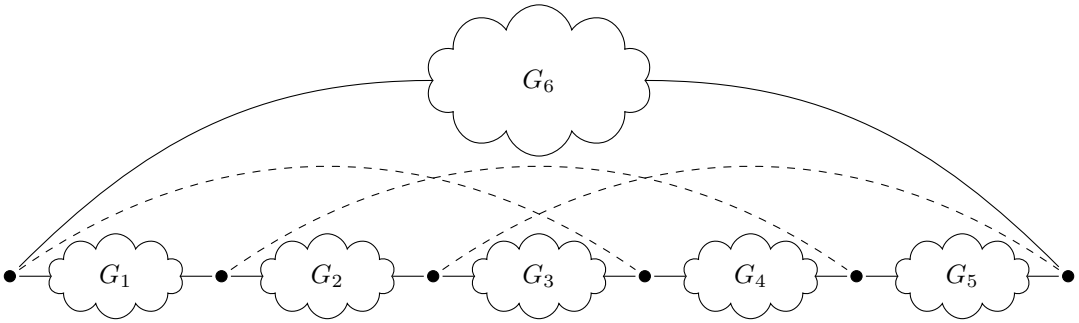

Figure 4: A very simple pairwise-planar instance. Each G_i is some planar graph. The demand edges cannot be simultaneously embedded so that $G + H$ is planar ($G + H$ contains a $K_{3,3}$-minor assuming each G_i is connected). However, any pair of demands can be drawn without crossing by putting one in the inside face, and one on the outside face. Of course, one could make the instance more complicated by putting demands inside the G_i's following a similar pattern.

2.2.1 Leaf Uncrossed Flows

A standard pre-processing operation for multiflows is to move each terminal to its own new leaf (i.e. degree 1) node. However, this may turn an uncrossable instance into one which is not - see the modified ring instance in Figure 5. For this instance, there is a different choice for embedding the leaves that works, but this is not always the case, even for rings (e.g. if sufficiently many demands cross inside the ring). We refer to an

[4]The Euler condition is that for each node v, the sum of the capacities and demands of its indicent edges in G and H is even.

Figure 5: Problematic instances for uncrossability. (Left) A congestion instance with a feasible multiflow but no uncrossed multiflow routing both demands. (Right) The *Keyhole Instance*, a congestion instance with a feasible uncrossed trail-multiflow but no uncrossed path-multiflow.

instance as *leafable* if all the terminals can be moved into leaves such that there is an uncrossed multiflow. Both fully-planar and single-source instances are leafable.

2.2.2 Strongly Uncrossed Multiflows

While Theorem 2.2 is intuitive, our current proof of Theorem 1.2 requires a slightly stronger condition. Converting a fractional flow to an integral flow entails "rounding" of each commodity's (fractional) flow paths to some unit flow path. A natural approach is to focus on the topological region where each commodity routes its flow. The support paths for a commodity $(s,t) \in E(H)$ have a clockwise ordering about s (and t) given by the parallelization. We refer to the regions bounded between consecutive paths in this order (defined more formally later) as *sectors*. Our arguments assume that sectors of different commodities do not cross. One might think this property follows from pairwise uncrossed flow paths, however Figure 6 shows this is not always the case.

The desired non-crossing sector property does hold for leafably uncrossed flows - however as discussed previously adding leaves can eliminate valid uncrossed solutions. We rely on a condition we call *strongly uncrossed* which implies non-crossing sectors but is less restrictive than leafably uncrossed. Informally, we disallow flow paths to "cut through" a terminal and split the paths originating there. Specifically, we prohibit the following configurations (again, we use parallelization to simplify the definition by assuming edge-disjoint paths).

Figure 6: (Left) An uncrossed multiflow where the sectors of the two commodities cross. There is a quasi-crossing at s_1. (Right) After disk-expansion of s_1 (solid edges), there is no way to add a single terminating node connecting the ends of the blue paths within the interior of the disk without crossing the green edge (a failed attempt is shown with the hollow node and wavy edges).

Definition 2.3 (Quasicrossing)**.** Let f, f' be single-commodity flows, and fix a parallelization of their support paths. The flows *quasicross* at v if in the parallelization, there are four distinct edges incident with v that

alternate clockwise between the two flows, and one of the following two conditions holds:

- all four edges terminate their respective support paths, or
- the two edges for one flow terminate their support paths, and the other two are consecutive edges of the same support path for the other flow.

The flows in Figure 6 quasicross at s_1, and also at t_1.

Definition 2.4 (Strongly Uncrossed)**.** A multiflow is *strongly uncrossed* if there exists a parallelization of its support paths such that no pair of paths cross and no pairs of its single-commodity flows quasicross.

Consider the disk expansion in Figure 6 (ignore the wavy edges and hollow node for now). The curves for different paths are disjoint in the disk, even for those of the same commodity. Suppose we wish to extend the curves into the disk interior so that curves of a common commodity ending at that node have the same termination point; in graph terms, for each commodity ending there, we seek to add a node into the open interior of the disk and connect it to the endpoints for that commodity's paths on the disk boundary (so each such commodity has a star in the disk). In the figure, it is impossible to place a star for the blue commodity without crossing the green path (the wavy edges are a failed attempt). However, it would be possible if there was no quasicrossing; then paths of the same commodity would be internally disjoint, while paths for distinct commodities remain totally disjoint.

Many instance classes admit strongly uncrossed flows, including fully-planar, single-source, and ring instances. Strongly uncrossable is less restrictive than leafable, since as mentioned some ring instances are not leafable.

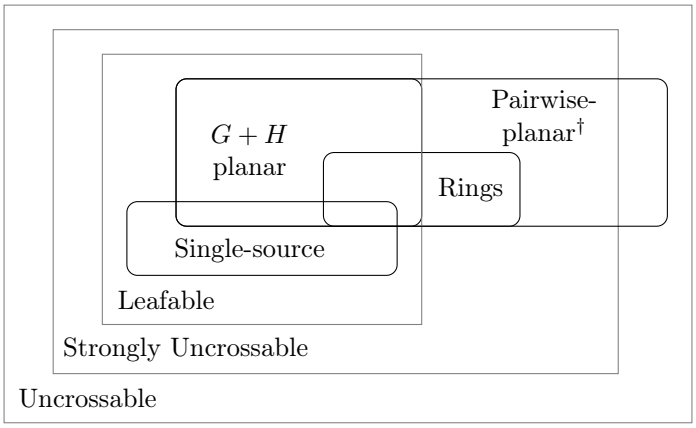

Figure 7: Venn diagram of the various uncrossable classes. †We do not know if there exist pairwise-planar instances that are uncrossable but not strongly uncrossable.

2.2.3 Paths, Trails, and Walks

In network flow theory it is standard to consider only routing on (simple) paths, since a flow walk could just be short-circuited into a flow path. Somewhat disturbingly, this operation is also not benign in the uncrossed flow setting. The *Keyhole Instance* of Figure 5 is an example that requires the use of a non-simple flow trail if we seek an uncrossed solution. Since we would like to understand which instances have uncrossed solutions, we must consider *trail-multiflows* and *walk-multiflows* in addition to the usual *path-multiflows*. Recall a trail can repeat nodes but not edges, while a walk can repeat both. Ultimately our rounding and hardness results presented earlier hold for all three models.

Of course, we should define what it means for walks to be uncrossed. The intuition remains that uncrossed walks correspond to disjoint curves within a "widened" version of the graph; for ease we essentially take this

to be the definition. We obtain a generalized notion of a parallelization (from Section 2.1) by splitting an edge into a parallel copy for each *occurrence* of the edge in a family of walks, and then assigning each walk a number of those copies equal to the edge's multiplicity within the walk. The parallelization is *uncrossed* if there is no node v such that, in clockwise order about v, there are four distinct edges e, g, e', g' where e, e' are consecutive edges of a walk and g, g' are consecutive edges of a walk. Note that this also prohibits a walk from crossing itself. We define a family of walks to be *uncrossed* if it has some uncrossed parallelization. The definition of *quasicrossing* also extends to walks (note that where the definition specifies *consecutive* edges is very relevant for walks, since they may have many edges incident with a node). For the definition of *strongly uncrossed*, we also disallow a single-commodity flow from quasicrossing with itself.

For walk-multiflows, a question arises of whether we should count a walk's repeat nodes or edges multiple times towards the capacities. That is, if we let $m(P, e)$ denote the number of times e appears in walk P, should the edge capacity constraint be $\sum_{P \in \mathcal{P}: e \in P} f(P) \leq u(e)$ or $\sum_{P \in \mathcal{P}: e \in P} m(P, e) \cdot f(P) \leq u(e)$ (and similar for node capacities)? We say flows satisfying the former are *feasible* and the latter *m-feasible* (short for "multiplicative"). The m-feasible condition is arguably more intuitive for capacities, however as we will discuss our rounding arguments still apply for feasible flows (giving a feasible integral solution). This is useful because if we are using uncrossed flows only as a means to obtain integer flows, without caring whether the final integer solutions are uncrossed, it is fine to short-circuit the final integer walk-multiflow to a path-multiflow. For this we do not need the integer walk-multiflow to be m-feasible, only feasible. Hence we broaden the applicability of our methods by only requiring feasibility rather than m-feasibility.

We remark that there are special cases where we can safely perform short-cutting operations without introducing crossing, or assume we have simple flow-paths. For example, if a walk has two occurrences of an edge e that are adjacent in the linear order of e (including for example when only that walk uses that edge), then it is safe to short-cut. We cover other conditions as they become relevant to our proofs.

2.2.4 Uncrossing with Congestion

Finally, a natural question is whether one may always convert a fractional multiflow solution to an uncrossed one with bounded edge congestion[5]. Unfortunately this is not the case; one may modify the canonical grid instance [11] so it has a half-integral routing of demands, but any uncrossed multiflow incurs polynomial congestion.

2.3 Polynomial-Sized Solutions

When discussing algorithms that take uncrossed multiflows as input, we assume we are given the (uncrossed) support paths and their values $f(P)$. However, since the U-constrained programs of (1), (2), and (3) are no longer linear programs, it is not necessarily obvious that we should expect uncrossed solutions that are polynomial in the size of the input (i.e. in terms of $|V|, |E(G)|, |E(H)|$, and the bit-size of u, d).

We give a brief argument for why polynomial-sized solutions exist. Let f be a feasible uncrossed (fractional) multiflow for a maximization or congestion instance. Consider a single commodity $h = (s, t) \in E(H)$, and consider the single-commodity uncrossed multiflow f^h, obtained by restricting f to h. Let $P_1, \ldots, P_k$ be the clockwise ordering of the support paths about s. Each sector (i.e. the region bounded between consecutive paths in this ordering) contains some face F of G that is not contained in any other sector. Hence there are at most $O(|V|)$ sectors, and thus $O(|V|)$ paths. Thus over all commodities the set of support paths $\mathcal{P}^*$ is polynomial in cardinality. To obtain values of $f(P)$ that have polynomial bit-size, we can set up restricted versions of the linear programs (1), (2), and (3) that only have variables for $P \in \mathcal{P}^*$. By standard linear programming theory there exists a solution for these problems that has polynomial bit-size, and any solution for these LPs is uncrossed as they draw support paths only from $\mathcal{P}^*$.

A consequence of this argument is the U-constrained version of (3) is in NP, as promised by Theorem 1.4. Given a polynomial-sized candidate solution, it is straightforward to check that each pair of support paths is uncrossed and that the solution is feasible in polynomial time.

[5]We thank Bertrand Guenin for posing this question to us.

3 Rounding and Hardness for Maximization Setting

In this section, we prove Theorem 1.1 and Theorem 1.3, which cover our rounding and inapproximability resulting for maximum uncrossed edge-capacitated (1) and node-capacitated multiflow (2).

3.1 Uncrossed String Graphs and Approximate Integer Decomposition

First we prove Theorem 1.1. We focus on the node-capacitated problem (2), since the edge-capacitated version (1) follows from the same arguments (and is slightly easier to argue anyway).

Our approach is to establish a so-called approximate integer decomposition property [3, 4] for uncrossed multiflows. The outline of the strategy is as follows. Suppose we are given an uncrossed multiflow f, which achieves a value of β. Imagine scaling f by a positive integer k such that kf is integral. Of course, kf is unlikely to be feasible. However suppose we are able to show that kf can be decomposed as the sum of a small number b of feasible integral flows. That is $kf = \sum_{i=1}^{b} f^i$ where each f^i is integral and feasible. At least one of the f^i flows achieves a $1/b$ fraction of the value of kf, and hence a k/b fraction of the value of f.

How can we achieve such a decomposition? Suppose (for simplicity) that kf sends 1 unit of flow along (simple) paths $\mathcal{P} = \{P_1, \ldots, P_\ell\}$. Furthermore suppose that each node of G has unit capacity and hence each appears on at most k paths. Our goal then is to partition the paths into classes such that every node appears at most once in each class; the classes then define the flows f^i. Equivalently, we seek to partition the paths so that paths in the same class do not share a node. Phrased this way, we can recognize this as a graph colouring problem on the node-intersection graph of $\mathcal{P}$.

The node-intersection graph U of $(G, \mathcal{P})$ has node set $\mathcal{P}$, and a node between distinct $P_1, P_2 \in \mathcal{P}$ if there exists $v \in V(P_1) \cap V(P_2)$. The *load* of $v \in V(G)$ is the number of paths in which v occurs, and the *load* of $(G, \mathcal{P})$ is the maximum load of a node $v \in V(G)$. Graphs U constructed in this way are a special type of the well-studied family of *string graphs*, which are the intersection graphs of curves in the plane.

Definition 3.1 (Uncrossed String Graph, Realization)**.** An *uncrossed string graph* U is the node-intersection graph of a family of uncrossed paths $\mathcal{P}$ in a planar graph G. We say $(G, \mathcal{P})$ is a *path-realization* of U.

Uncrossed string graphs from path-realizations have been studied before for their own sake [8, 26]. Given that U has a load k path-realization, we want that its chromatic number $\chi(U)$ is some small function of k, ideally $O(k)$. This question is strongly related to the concept of a χ-*bounded* graph family. A family of graphs $\mathcal{G}$ is χ-bounded if there exists a function f such that for all $U \in \mathcal{G}$, $\chi(U) \leq f(\omega(U))$, where $\omega(U)$ is the clique number of U. It is worth noting that general string graphs are not χ-bounded, and it is not obvious that uncrossed string graphs should be. An $O(k)$ bound was conjectured originally in [8], and eventually proven by Fox and Pach [9].

Theorem 3.2 (Fox and Pach)**.** *Let U be an uncrossed string graph with realization $(G, \mathcal{P})$ such that each node $v \in V(G)$ has load at most k. Then $\chi(U) \leq 6ek + 1$.*

The result of Fox and Pach has not been published, however the proof is elegant and short. It establishes via a probabilistic proof (similar to a standard proof giving a bound on the crossing number of a graph) that U has at most $3ek|\mathcal{P}|$ edges. Hence a greedy colouring procedure that removes the minimum degree vertex (which has degree at most $6ek$) and colours the rest of the graph inductively obtains a $6ek + 1$ colouring. We were not initially aware of Fox and Pach's result and independently proved a weaker bound that $\chi(U) \leq O(k^2 \log |\mathcal{P}|)$. For the sake of completeness we present this proof in Section C.

Now, thus far we have discussed (simple) paths, but as we mentioned previously, understanding walk-multiflows is important to the problem model of uncrossed flows. It is straightforward to generalize the definition of uncrossed string graph and realization to walks (giving us *walk-realizations).* We note that we do not include loops (from self-intersection) in U. From here on when we say uncrossed string graph, we mean it can arise via walks.

Technically, Fox and Pach only state their colouring result when $\mathcal{P}$ is a set of paths, but the generalization to walks is not hard, though there is one technicality to consider. Similar to feasible vs. m-feasible, because

of the possibility that a node appears multiple times in a single walk, there are two natural notions of load: for $v \in V(G)$, we define its *load* to be the number of distinct walks in which v appears, and its *m-load* to be the number of occurrences of v in the walk family (counting multiple times if v appears more than once in a given walk). The Fox and Pach proof (and our colouring result in the appendix) actually only requires the load to be at most k to get the colouring.

We now use the colouring result to establish an approximate integer decomposition property and prove the following theorem, which is a generalization of Theorem 1.1 for the node-capacitated case.

Theorem 3.3. *There exists a universal constant $\gamma \geq 1$ such that the following holds. Let (G, H, u) be a planar maximization instance of LP (2), equipped with per-unit profits $w(h) \geq 0$ for $h \in E(H)$. Let f be a feasible uncrossed walk-multiflow. There exists a feasible integral walk-multiflow f^* that is uncrossed[6] and whose value with respect to w is at least $\frac{1}{\gamma}$ times that of f. This solution can be computed in polynomial time if the capacities are unit-valued.*

Proof. We first handle where $u = \vec{1}$. Let β be the value of f under weights w. Let $\mathcal{P}$ be the family of support walks for f. Let k be a positive integer such that kf is integral. Since kf is integral, for each P we have that $f(P) = \alpha_P/k$ where α_P is a positive integer. By treating each P as α_P distinct walks, we may assume that each walk routes exactly $\frac{1}{k}$ units of flow. Then, since $\sum_{P:v\in P} f(P) \leq 1$ by feasibility, at most k walks use each node of G (not counting multiplicity).

Let U be the node-intersection graph of $(G, \mathcal{P})$, which has load (not necessarily m-load) at most k. By the result of Fox and Pach, U is b-colourable where $b = O(k)$. Now for each colour class i, we let f_i denote the flow obtained by sending one unit of flow on each walk in that colour class, and zero flow on all other walks. Then f_i is integral and satisfies that $\sum_{P:v\in P} f_i(P) \leq 1$ for each $v \in V(G)$, and $kf = \sum_{i=1}^{b} f_i$ since each walk's flow was blown up by a factor of k. Since the value of kf is $k\beta$, at least one of the f_i flows has a value of $k\beta/b$. Since $b = O(k)$, then this flow has value $\Omega(1)$ times that of f, as desired.

Of course, since k may be large, this does not immediately give a polynomial time algorithm. To fix this, we pre-process f. By performing randomized rounding and re-scaling (which can actually be de-randomized to give a deterministic algorithm) we can obtain a feasible f' whose fractionality is polynomial in $|V|$ (i.e. kf' is integral where $k \leq \text{poly}(|V|)$, and whose value is $\Omega(1)$ times that of f. We refer the reader to [2] (Lemma 5.1) for more details. Note that we may always assume $\alpha_P \leq k$, since otherwise we can remove $\lfloor f(P) \rfloor$ flow from each P, perform our decomposition argument on the remaining flow to round it, and re-add the removed flow to get our final integral solution.

For general capacities, we reduce to the unit capacity case by splitting each node v into $u(v)$ copies[7]. First, using the uncrossed parallelization, we perform disk expansion at v (see Section 2). Then, proceeding clockwise, we combine the nodes together until we have 1 unit of flow. We can create multiple copies of paths and split flow between the copies to ensure that we do not ever exceed 1 unit of flow when merging nodes together.

□

3.2 Inapproximability of Maximum Uncrossed Fractional Multiflow

Next we prove Theorem 1.3 on the inapproximability of maximum uncrossed (fractional) multiflow. We make use of a strong inapproximability result given by Chuzhoy, Kim and Nimavat for maximum node-disjoint paths (MNDP) and maximum edge-disjoint paths (MEDP) in planar graphs; i.e. the integer program versions of (1) and (2) with unit capacities. One minor technicality is that in their formulation, a feasible solution is only allowed to route one unit of flow for each demand pair. We thus call these problems *capped MNDP* and *capped MEDP* respectively.

Theorem 3.4 (Theorems 1.1 and 1.2, [7])**.** *For every constant $\epsilon > 0$, there is no $2^{\Omega(\log^{1-\epsilon} n)}$-approximation polynomial-time algorithm for capped MEDP or capped MNDP, assuming that $NP \not\subseteq DTIME(n^{polylog\ n})$.*

[6] Recall, we can short-circuit this flow to a path-multiflow, though we cannot guarantee it will remain uncrossed.

[7] This reduction is not polynomial time. We believe the general capacity case can be solved in polytime but defer this to the full version.

Moreover, there is no $n^{O\left(\frac{1}{(\log\log n)^2}\right)}$*-approximation polynomial-time algorithm for capped MEDP or capped MNDP, assuming that for some constant* $\delta > 0$*,* $NP \not\subseteq DTIME(2^{n^\delta})$*. These results hold even when the input graph is a maximum degree* 3 *planar graph, or even if* G *is a wall graph.*

We now prove Theorem 1.3 which relies on this result.

Proof of Theorem 1.3. We prove that result by giving a polytime reduction from capped MEDP/MNDP in wall graphs to maximum uncrossed walk-multiflow in wall graphs, that preserves the approximation ratio up to a constant factor. We only present the proof for the edge-capacitated, walk-multiflow case, since the other reductions with node capacities and/or path-multiflows are essentially the same (the arguments are actually slightly easier). For an instance (G, H) of the capped MEDP problem where G is a wall graph (which is planar and has maximum degree 3), define the following quantities.

- Let $D_E(G, H)$ denote the maximum number of demands in H which can be routed on edge-disjoint paths in G (i.e. the optimal value for capped MEDP).
- Let $U_E(G, H)$ denote the maximum value of a (fractional) uncrossed walk-multiflow for $(G, H, u = \vec{1})$.

Observe that any capped MEDP solution for (G, H) is a feasible (integral) uncrossed path-multiflow for $(G, H, u = \vec{1})$. To see this, consider a collection of edge-disjoint paths that form a capped MEDP solution. Recall from Section 2.1 that two edge-disjoint paths cross only if they cross at a node (rather than a longer common subpath). Since G has maximum degree 3, a node crossing is not possible. Thus we have $D_E(G, H) \leq U_E(G, H)$.

Suppose we have a β-approximation algorithm for maximum uncrossed walk-multiflow in wall graphs. We obtain an approximation algorithm for capped MEDP as follows. We run our β-approximation algorithm for maximum uncrossed (fractional) walk-multiflow on $(G, H, u = \vec{1})$ to obtain flow f whose value is at least $U_E(G, H)/\beta$. We apply the polytime algorithm from Theorem 3.3 to compute a feasible uncrossed integral walk-multiflow f' whose value is $1/\gamma$ that of f. We then short-cut f' into a feasible integral path-multiflow[8]. It may still not be a valid capped MEDP solution, since it may route more than one flow path for the same commodity. However, since G has maximum degree 3, it routes on at most 3 paths for each commodity, so by selecting exactly one path for each commodity we obtain a capped MEDP solution whose value is at least $\frac{1}{3}$ that of f'. Thus this algorithm outputs a capped MEDP solution whose value is at least $\frac{U_E(G,H)}{3\gamma\beta} \geq \frac{D_E(G,H)}{3\gamma\beta}$, and is thus a $3\gamma\beta$-approximation algorithm.

□

4 Rounding for Congestion Minimization

In this section, we focus on planar congestion instances (3), where we are given demands d that must all be routed. We show Theorem 1.2, which states that we can convert (fractional) strongly uncrossed multiflows into integral flows that violate the capacities by a factor of at most 2, and if we accept an additive error of less than $d_{\max} = \max_{h \in E(H)} d(h)$ the flow can be made unsplittable. Section 4.1 presents the relevant properties of strongly uncrossed flows, and Section 4.2 gives the proof of Theorem 1.2.

4.1 Clockwise Orderings and Sectors

Let f be a strongly uncrossed walk-multiflow for planar congestion instance (G, H, u, d). Fix a strongly uncrossed parallelization ϕ of its support walks. We parallelize the edges and perform disk-expansion at each node (see Section 2.1) so that the walks become node-disjoint paths. For each commodity terminal, we add a node into the disk to join the endpoints of paths for the same commodity, which can be done in a planar fashion since the flow is strongly uncrossed (see Section 2.2.2). In the new graph, which we denote G_ϕ,

[8]Given that G is maximum degree 3 and $u = \vec{1}$ we can actually show this short-cutting can be done so the resulting flow is still uncrossed, but it is not necessary for this proof.

walks for the same commodity become internally node-disjoint paths, while walks for different commodities (even if the commodities have parallel edges in H) are (completely) node-disjoint paths.

Consider a commodity $h = (s, t) \in E(H)$ and the uncrossed family of st-walks making up the support for f^h. We use the notation $G_\phi[f^h]$ to mean the support graph of f^h in G_ϕ. In G_ϕ, the paths corresponding to the support walks have a clockwise cyclic ordering about s, say $P_1, \ldots, P_k$.

Informally, a *sector* is the region between consecutive paths in the clockwise ordering. Consider two paths P_i and P_j in G_ϕ. Orient P_i from s to t, and orient P_j from t to s. The directed curve $P_i \bullet P_j$, obtained by concatenating these oriented curves, is a simple closed (Jordan) curve, and partitions the plane into two regions. We define $\mathrm{CW}^\circ(P_i, P_j)$ to be the set of points x in $\mathbb{R}^2 \setminus (P_i \cup P_j)$ such that x is "to the right" of $P_i \bullet P_j$; that is, if we draw a directed curve from x to $P_i \bullet P_j$ (avoiding $P_i \bullet P_j$ except at the extremity), it hits $P_i \bullet P_j$ from the right. We let $\mathrm{CW}(P_i, P_j)$ denote the closed region $\mathrm{CW}^\circ(P_i, P_j) \cup P_i \cup P_j$. We analogously define $\mathrm{ACW}^\circ(P_i, P_j)$ and $\mathrm{ACW}(P_i, P_j)$ by taking the opposite orientation. Any path that is uncrossed and internally disjoint from P_i and P_j either stays entirely $\mathrm{CW}(P_i, P_j)$ or stays entirely within $\mathrm{ACW}(P_i, P_j)$.

Now, for our clockwise-ordered family of st-paths $P_1, \ldots, P_k$, we refer to $\mathrm{CW}^\circ(P_i, P_{i+1})$ (resp. CW) for $i = 1, \ldots, k$ (addition modulo k) as the *open* (resp. *closed*) *sectors* of f^h. The open sectors partition $\mathbb{R}^2 \setminus (P_1 \cup \cdots \cup P_k)$. One sector contains the infinite face of $G_\phi[f^h]$; we call this the *outer sector*, and the others the *inner sectors*. We call the two paths of G_ϕ that define the outer sector the *outer paths*, and their corresponding walks in G the *outer walks*. If an edge e of the original graph G lies in an outer walk, it is an *outer edge*, otherwise it is an *inner edge*. For our congestion result, we exploit the following key properties.

Lemma 4.1. *Let f be a strongly uncrossed walk-multiflow, with a fixed strongly uncrossed parallelization ϕ. For distinct commodities $h, h' \in E(H)$, the sets of inner edges of h and h' are disjoint.*

Proof. To prove the result, we show that if an edge e is shared between the two single-commodity flows in G, then it is an outer edge for one of the commodities.

Claim 4.2. *$G_\phi[f^{h'}]$ is entirely contained within a single sector of h, and $G_\phi[f^h]$ is entirely contained within a single sector of h.*

Proof. By symmetry, it suffices to prove the first part of the claim. Recall that in G_ϕ, support paths for distinct commodities are completely disjoint curves, so each individual support path for h' lies in at most one sector of h. Additionally, all support paths for the same commodity are connected at the terminals, so that sector must be the same for all support paths of h'. ◇

Claim 4.3. *Either $G_\phi[f^{h'}]$ is contained in the outer sector of h, or $G_\phi[f^h]$ is contained in the outer sector of h.*

Proof. By the previous claim and by symmetry, it suffices to show that if $G_\phi[f^{h'}]$ is contained within an inner sector of h, then $G_\phi[h]$ is contained within the outer sector of h'. Suppose $G_\phi[f^{h'}]$ is contained within the inner sector of h defined by paths P_i and P_{i+1}. Since $\mathrm{CW}(P_i, P_{i+1})$ is an inner sector, it is bounded, which means $G_\phi[f^{h'}]$ is contained in the interior of the curve $P_i \bullet P_j$ rather than the exterior. This implies that P_i and P_{i+1} are contained within the infinite face of $G_\phi[f^{h'}]$, and so therefore are in the outer sector of h'. Since the entire support of h is within the same sector of h', it is within the outer sector of h'. ◇

Now, without loss of generality, suppose that $G_\phi[f^{h'}]$ is contained within the outer sector of h. Then $G_\phi[f^h]$ and $G_\phi[f^{h'}]$ both contain one of copies of the shared edge e. The copy belonging to h' is contained in the infinite face of $G_\phi[f^h]$, and (by how G_ϕ is constructed) so too is the copy belonging to h. So then e is an outer edge of h, not an inner edge. □

Lemma 4.4. *Suppose that in G, P_i and P_j share an edge e. One of the following holds:*

- *every walk between P_i and P_j in the clockwise ordering also uses e, or*
- *every walk between P_j and P_i in the clockwise ordering also uses e.*

Proof. In G_ϕ, consider subdividing the copies of e belonging to P_i and P_j by adding one node into each edge. Then add an edge e' between the two new nodes. Add e' such that it only crosses copies of e. The new edge either separates $\mathrm{CW}(P_i, P_j)$ or it separates $\mathrm{ACW}(P_i, P_j)$, where by separates we mean that any curve that is within the sector must cross e'. The paths between P_i and P_j all lie in $\mathrm{CW}(P_i, P_j)$, and the paths between P_j and P_i all lie in $\mathrm{ACW}(P_j, P_i)$. If a path P_r crosses e', then it means that it uses one of the copies of e, and hence its corresponding walk in G also uses e. The result follows. □

4.2 Main Congestion Argument

We prove the following theorem, which generalizes Theorem 1.2.

Theorem 4.5. *Given a strongly uncrossed walk-multiflow that is feasible for (3), there is a polynomial time algorithm to compute both:*

1. *an integral path-multiflow routing all demands where the flow on each $e \in E(G)$ is at most $2u(e)$, and*
2. *an unsplittable path-multiflow routing all demands where the flow on each $e \in E(G)$ is strictly less than $2u(e) + d_{\max}$, where $d_{\max} = \max_{h \in E(H)} d(h)$.*

Proof. First, we show how to prove Item 2, since Item 1 ultimately follows from it. In what follows, let $f(e) = \sum_{P : e \in P} f(P)$.

Consider a commodity h. Let $P_1, \ldots, P_k$ be the family of uncrossed walks forming the support of f^h, ordered according to the cyclic clockwise ordering of their paths in G_ϕ. Re-index so that P_1 and P_k are the outer walks. Let i^* be the smallest index i such that $\sum_{j \le i} f^h(P_j) \ge \frac{d(h)}{2}$. We define $P^h = P_{i^*}$, and call it the *central walk* for h. Note that it is straightforward to find P^h in polynomial time.

Claim 4.6. *If $e \in P^h$ and e is an outer edge for h (i.e. it lies in P_1 or P_k) , then $f^h(e) \ge \frac{d(h)}{2}$.*

Proof. Since e is an outer edge, it is on either P_1 or P_k, in addition to $P^h = P_{i^*}$. From Theorem 4.4 we have that $e \in P_r$ either for all $1 \le r \le i^*$ or for all $i^* \le r \le k$. In the former case, $f^h(e) \ge \sum_{1 \le r \le i^*} f^h(P_r) \ge \frac{d(h)}{2}$. For the latter case, $f^h(e) \ge \sum_{i^* \le r \le k} f^h(P_r) = d(h) - \sum_{1 \le r < i^*} f^h(P_r) > d(h) - \frac{d(h)}{2} = \frac{d(h)}{2}$, where the last inequality follows since $\sum_{1 \le r < i^*} f^h(P_r) < \frac{d(h)}{2}$ by definition of i^*. ◇

To obtain an unsplittable flow f', for each $h \in E(H)$ we short-circuit P^h into a path Q^h, and route $d(h)$ units of flow along the path. The short-circuiting may cause the paths to cross, but that does not matter.

For the congestion bound, consider an arbitrary edge $e \in E(G)$. We will bound the amount of flow through e. Let $H_e = \{h \in E(H) : e \in Q^h\}$, so $f'(e) = \sum_{h \in H_e} d(h)$. Define $O_e = \{h \in H_e : e \text{ is an outer edge of } h\}$ and also $I_e = \{h \in H_e : e \text{ is an inner edge of } h\}$. These sets partition H_e, so $f'(e) = \sum_{h \in O(e)} d(h) + \sum_{h \in I_e} d(h)$. We know that $|I_e| \le 1$ from Theorem 4.1. Additionally, by Theorem 4.6, for each $h \in O_e$ we have that $f^h(e) \ge \frac{d(h)}{2}$. We now split into two cases.

Case 1: $\sum_{h \in O_e} \frac{d(h)}{2} < u(e)$. Then,

$$f'(e) < 2u(e) + \sum_{h \in I_e} d(h) \le 2u(e) + d_{\max}.$$

Case 2: $\sum_{h \in O_e} \frac{d(h)}{2} \ge u(e)$. Then $\sum_{h \in O_e} f^h(e) \ge u(e)$, so the commodities of O_e saturate e in f. It follows that $I_e = \emptyset$. Hence,

$$f'(e) = \sum_{h \in O_e} d(h) \le 2u(e) < 2u(e) + d_{\max}.$$

This concludes the proof of Item 2. The above is easily turned into a polynomial time algorithm. We now prove Item 1 from Item 2.

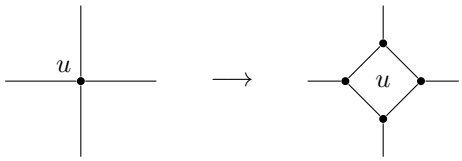

Figure 8: A gadget to replace a degree-4 vertex by 4 degree-3 vertices.

First, we split each $e \in E(G)$ into $u(e)$ parallel copies with capacity 1, and each $h \in E(H)$ into $d(h)$ parallel copies with demand 1, so that we have a new planar multiflow instance $(G', H', u' = \vec{1}, d' = \vec{1})$ with unit capacities and demands. The splitting step is technically not polynomial time, but at the end we discuss how to convert this to an efficient algorithm. An integral congestion 2 multiflow for the new instance clearly implies a congestion 2 multiflow for the original instance. Given the feasible strongly uncrossed walk-multiflow for the original instance, it is straightforward to construct a feasible strongly uncrossed walk-multiflow for the new instance. To do this, for commodity $h = (s, t) \in E(H)$, order the support walks clockwise about s. The first unit-demand st-flow is obtained by iterating over the walks, assigning up to $f^h(P)$ amount of flow on each walk P, until 1 unit of flow has been routed. We then continue iterating from this walk and assign flow for the second unit-demand st-flow, ensuring we do not assign more than $f^h(P)$ flow in total to each walk P, and so on.

Then, Item 2 implies an there exists an unsplittable (and hence, integral) multiflow f' for the unit weight instance, where the congestion on each edge is strictly less than 3. Since the flow is integral, the congestion is at most 2.

We now briefly explain how to convert this into a polynomial time algorithm. Consider a demand $h = (s, t) \in E(H)$. The following is an equivalent description of how the pseudo-polynomial time algorithm above behaves. It picks a starting support walk and scans clockwise about s counting the total amount of flow it sees, finding the first walk at which 0.5 units of total flow has been routed, then 1.5 units in total, then 2.5 units in total, and so on. These are what would be the central walks if we split h up into $d(h)$ separate demands. The algorithm iterates over this list, and routes one unit of flow on each walk (if a walk appears multiple times in the list because it's flow is large, we route one unit of flow on it each time it appears). However, just by scanning clockwise about s and performing a constant number of simple arithmetic calculations at each walk, we can determine how many times each walk P would appear in this list of central walks, which tells us how much flow to route on P. □

5 Reducing Integral Uncrossed Flow to Node Disjoint Instances

In this section we prove Theorem 1.5. We reduce to the standard (not U-constrained) planar edge-disjoint paths problem (i.e. the integer version of (3) with unit demands and capacities), and then use a result of Schrijver [21] which says that edge-disjoint paths can be solved in polynomial time in planar graphs if the number of faces the demand edges are incident upon is bounded.

Our proof uses contractions. We let G/e to denote the graph obtained by contracting edge e in a graph G; we use similar notation for a multiflow f.

Lemma 5.1. *Let f be an uncrossed flow in G and e some supply edge. Then f/e is uncrossed for $G/e, H/e$.*

We first present a gadget that reduces the degree of an arbitrary vertex of degree at least 5, by one, adding vertices of maximum degree 4. Using this gadget repeatedly, we get an equivalent instance with maximum degree 4. Then every degree-4 vertex can be replaced by a C_4, as illustrated in Figure 8, to obtain a graph with maximum degree 3.

Consider a vertex v with $\deg(v) \geq 5$. In order to obtain maximum degree 4, the natural approach replaces v by a grid large enough to make any routing incident to v feasible in that grid. This is more straightforward in instances where the terminals are leaves. As we have seen however, creating leaves may destroy uncrossability. We need to be more careful then because v can be a terminal for some of the

demand pairs. Therefore we must pre-specify some vertices in the grid to substitute v in those demand pairs. Because of the uncrossable constraint, the choice of these new terminals is more delicate. We give an explicit construction, presented in a recursive fashion: by showing how to decrease the degree of v by one in G and H simultaneously.

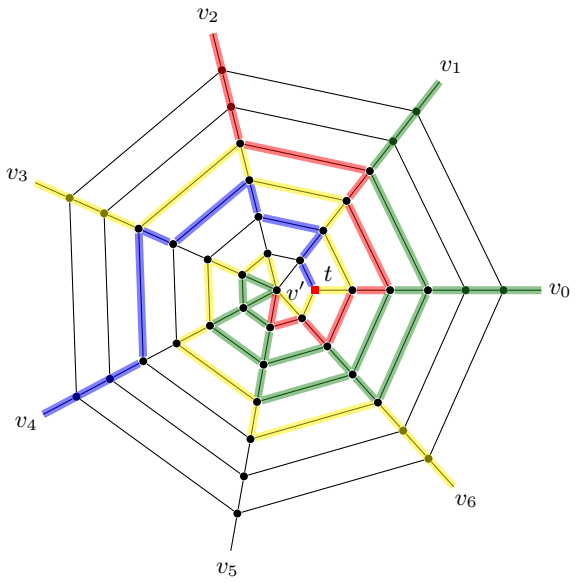

Figure 9: The spider web gadget: how to decrease the degree of a vertex by one in G and H, adding only vertices of degree at most 4. Notice the red terminal that can be used as terminal for one of the demand pair ending at this vertex. In highlight with colors, an example of reconfiguration of four paths as given by the proof, with two paths which had been ending at v (red and blue), and two paths passing through v (yellow and green).

Proposition 5.2. *Let G, H be an instance of the uncrossable edge-disjoint paths problem, and v be a vertex of G with $\deg_G(v) > 4$, or $\deg_G(v) \geq 2$ and $\deg_H(v) \geq 1$. Define G', H' from G, H by replacing v in G by the spider web gadget from Figure 9. If $d_H(v) \geq 1$, then replace arbitrary demand (s, v) in H by a demand (s, t) where t is the red terminal in the gadget (which then has degree 1 in H'). Any other terminals at v are located at v'. Then G, H admits an uncrossable routing if and only if G', H' does.*

Proof. Let $d = \deg(v)$. We label each vertex except v in the spider web as $u_{i,j}$ such that $u_{i,j}$ is at the polar coordinate $(\frac{2i\pi}{d}, j)$ (for instance $t = u_{0,1}$), with $i \in \{0, \dots, d-1\}$ and $j \in \{1, \dots, d-1\}$. Let $v_0, \dots, v_{d-1}$ be the neighbours of v, with v_i adjacent to $u_{i,d-1}$. To avoid confusion, we will call v' the central vertex in the spider web, and v the original vertex in G. Also we will consider all indices modulo d. Given a routing in G', H' by contracting all new vertices in the web (and deleting loops) they become the original v, and we obtain a routing in G, H, proving the *if* implication.

Consider an uncrossed routing $\mathcal{P}$ in G, H. We compute a corresponding uncrossed routing in G', H'. The first case is when H does not contain a demand ending at v. We claim then there is a routing that does not use v'. Note that our figure does not depict this case and in particular v' will not be used and can be safely removed, so that all of the new nodes have degree at most 4. Indeed to see this routing, consider a support flow path P containing $v_i v$, $v v_j$ (with $i < j$). Let P' be the flow path obtained from P by replacing $v_i v, v v_j$ by a subpath v_i, $u_{i,d-1}$, $u_{i,d-2}$, $\dots u_{i,d-k}$, $u_{i+1,d-k}$, $u_{i+2,d-k}, \dots$, $u_{j,d-k}$, $u_{j,d-k+1}, \dots$, $u_{j,d-1}$, v_j, with $k = j - i + 1$. This subpath uses the rays indexed by i and j and the ring indexed by $d - k$. We claim that no two such subpaths share an edge. Indeed all the rays are distinct, and if two paths used the same ring, one from v_i to v_j and the other from $v_{i'}$ to $v_{j'}$, as $j - i = j' - i'$, we must have v_i, $v_{i'}$, v_j, $v_{j'}$ appearing in that order cyclically around v. Hence the two corresponding paths cross, contradiction.

In the second case H has some demands terminating at v; this case is depicted in Figure 9. Let (s, v) be the arbitrary demand in H terminating at v that is replaced in H' by a demand (s, t), where $t = u_{0,1}$. Let $v_i v$ be the last edge in the s, v-path P in $\mathcal{P}$. Replace in P the edge $v_i v$ by the subpath v_i, $v_{i,d-1}$, $v_{i,d-2}$,..., $v_{i,i}$, $v_{i-1,i}$, $v_{i-1,i-1}$,...,$v_{2,2}$, $v_{1,2}$, $v_{1,1}$, $v_{0,1} = t$. For any other edge vv_j in H that appears in some path $Q \in \mathcal{P}$ (either transiting through or terminating at v), similarly replace vv_j in H by a subpath v, $v_{j-k,1}$, $v_{j-k+1,1}$, $v_{j-k+1,2}$,...,$v_{j-k+i,i} = v_{j,i}$, $v_{j,i+1}$,..., $v_{j,d-1}$, v_j. Each such subpath is thus spiraling counterclockwise from v (or t for P) into the i innermost rings, then goes straight on its own ray, and thus all such subpaths, which are rotations of each other, are disjoint. Moreover, the relative cyclic order of the paths incident to v' is the same as in v (except for the path P that is not incident to v'). Therefore, this routing in G', H' is uncrossed. □

We may repeatedly apply the preceding lemma, in polytime until we have a new "equivalent" instance G', H' with the following properties. If v has no terminals, then it has degree at most four and any terminal node has degree at most 3. The last step of the reduction now applies a degree 4 reduction (Figure 8) to each remaining vertex of degree 4. For each terminal node of degree at most 2 in G, we leave it. If a terminal node v has $d_G(v) = 3$, then we subdivide each edge incident to v and add a triangle onto these new degree 2 nodes. We show that this last reduction step is fine as follows.

Lemma 5.3. *Let G be a maximum degree 4 planar graph and H an associated demand graph which forms a matching and if $\deg_H(v) = 1$, then $\deg_G(v) \leq 3$. Let G'' be the graph obtained by performing the last phase of reductions. H has an integral uncrossed flow in G if and only if it has a totally node-disjoint flow in G''.*

Proof. We use the familiar idea that feasibility of edge- versus node-disjoint paths are equivalent in maximum degree three instances. If H has a node-disjoint solution in G'', then by planarity, there is an uncrossed edge-disjoint solution in G. We obtain an edge-disjoint solution in G by first contracting the new 4 cycles. This cannot create a crossing at the contracted node due to the fact we had a node-disjoint solution. Second, consider one of the nodes v where we applied triangle operation. Regardless of whether a demand transits through the triangle gadget, we obtain an uncrossed edge-disjoint flow by contracting v together with the new degree 2 nodes.

Conversely, suppose that we have an uncrossed edge-disjoint path solution in G. At any non-terminal node v, if there are two paths using it, since they are non-crossing they can be realized as node-disjoint on the 4-cycle introduced by the reduction. Now consider a terminal node v. If it has degree at most 2, then no other path could use edges incident to v, so assume it has degree 3. We can convert this to a node-disjoint solution by first letting the terminal v route via the node which subdivided the edge it used in G. This allows us to use the edge between the two other subdivided nodes for the transiting demand, thus keeping node disjointness at the gadget. □

This completes the proof of the overall reduction.

Theorem 5.4. *Suppose that G, H is a planar instance of EDP (or integer multiflow). We can compute a planar instance $\hat{G}, \hat{H}$ of node-disjoint paths such that G, H has an integral uncrossed trail flow if and only if $\hat{G}, \hat{H}$ has (totally) node-disjoint paths for the demands in $\hat{H}$. Moreover, $\hat{G}, \hat{H}$ can be computed in polytime, and $|V(\hat{G})| \leq c\Delta^3|V(G)|$ where Δ is the maximum degree in G and c is a constant.*

References

[1] Amit Chakrabarti, Lisa Fleischer, and Christophe Weibel. When the cut condition is enough: A complete characterization for multiflow problems in series-parallel networks. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 19–26. ACM, 2012.

[2] Chandra Chekuri and Sanjeev Khanna. Edge-disjoint paths revisited. *ACM Transactions on Algorithms*, 3(4):46, November 2007.

[3] Chandra Chekuri, Marcelo Mydlarz, and F Bruce Shepherd. Multicommodity demand flow in a tree and packing integer programs. *ACM Transactions on Algorithms (TALG)*, 3(3):27, 2007.

[4] Chandra Chekuri and F Bruce Shepherd. Approximate integer decompositions for undirected network design problems. *SIAM Journal on Discrete Mathematics*, 23(1):163–177, 2009.

[5] Chandra Chekuri, F Bruce Shepherd, and Christophe Weibel. Flow-cut gaps for integer and fractional multiflows. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1198–1208. Society for Industrial and Applied Mathematics, 2010.

[6] Julia Chuzhoy, David HK Kim, and Rachit Nimavat. Almost polynomial hardness of node-disjoint paths in grids. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1220–1233, 2018.

[7] Julia Chuzhoy, David Hong Kyun Kim, and Rachit Nimavat. Almost polynomial hardness of node-disjoint paths in grids. *Theory of Computing*, 17(6):1–57, 2021.

[8] Louis Esperet, Daniel Gonçalves, and Arnaud Labourel. Coloring a set of touching strings. *Electronic Notes in Discrete Mathematics*, 34:213–217, August 2009.

[9] Jacob Fox and János Pach. Touching Strings. 2012.

[10] Naveen Garg, Nikhil Kumar, and András Sebő. Integer plane multiflow maximisation: one-quarter-approximation and gaps. *Mathematical Programming*, 195(1):403–419, 2022.

[11] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, 18(1):3–20, 1997.

[12] Michel X Goemans and David P Williamson. Primal-dual approximation algorithms for feedback problems in planar graphs. *Combinatorica*, 1(18):37–59, 1998.

[13] Chien-Chung Huang, Mathieu Mari, Claire Mathieu, Kevin Schewior, and Jens Vygen. An approximation algorithm for fully planar edge-disjoint paths. *SIAM Journal on Discrete Mathematics*, 35(2):752–769, 2021.

[14] Kazuhiko Matsumoto, Takao Nishizeki, and Nobuji Saito. Planar multicommodity fows, maximum matchings and negative cycles. *SIAM Journal on Computing*, 15(2):495–510, 1986.

[15] Colin McDiarmid, Bruce Reed, Alexander Schrijver, and Bruce Shepherd. Induced circuits in planar graphs. *Journal of Combinatorial Theory, Series B*, 60(2):169–176, 1994.

[16] Matthias Middendorf and Frank Pfeiffer. On the complexity of the disjoint paths problem. *Combinatorica*, 13(1):97–107, March 1993.

[17] Guyslain Naves. The hardness of routing two pairs on one face. *Mathematical Programming*, 131(1-2):49–69, February 2012.

[18] Janos Pách. personal communication.

[19] Niklas Schlomberg. An improved integrality gap for disjoint cycles in planar graphs. *arXiv preprint arXiv:2404.17813*, 2024.

[20] Niklas Schlomberg, Hanjo Thiele, and Jens Vygen. Packing cycles in planar and bounded-genus graphs. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2069–2086. SIAM, 2023.

[21] Alexander Schrijver. Homotopic routing methods. *Paths, Flows, and VLSI-layout*, pages 329–371, 1990.

[22] Alexander Schrijver. Disjoint homotopic paths and trees in a planar graph. *Discrete & Computational Geometry*, 6:527–574, 1991.

[23] Alexander Schrijver. Finding k disjoint paths in a directed planar graph. *SIAM Journal on Computing*, 23(4):780–788, 1994.

[24] Paul D Seymour. Matroids and multicommodity flows. *European Journal of Combinatorics*, 2(3):257–290, 1981.

[25] Vera Traub, Laura Vargas Koch, and Rico Zenklusen. Single-source unsplittable flows in planar graphs. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 639–668. SIAM, 2024.

[26] Wouter Cames Van Batenburg, Louis Esperet, and Tobias Müller. Coloring Jordan Regions and Curves. *SIAM Journal on Discrete Mathematics*, 31(3):1670–1684, January 2017.

A Uncrossed Flows in Pairwise-Planar Instances

Here we introduce a new class of multiflow instances and prove they are uncrossable.

Definition A.1. We say (G, H) is *pairwise-planar* if there exists a planar embedding of G such that each $h \in E(H)$ lies in some face of G, and moreover for any pair $h_1, h_2 \in E(H)$, h_1 and h_2 can each be embedded inside (possibly different) faces of G so that they do not cross each other.

This is more general than fully-planar instances, because we do not require all demands to be simultaneously uncrossed.

A.1 Pairwise-Planar Instances are Uncrossable

Theorem A.2. *For a pairwise-planar instance (G, H, u, d), if there exists a feasible multiflow, there exists a feasible uncrossed multiflow.*

Proof. For a flow f of and a given parallelization (not necessarily uncrossed), we define the following. For $v \in V$ and distinct paths P, Q, let $\kappa(f, v, P, Q)$ be $f(P) \cdot f(Q)$ if P and Q cross at v, and 0 otherwise. For $v \in V$ and $h_1, h_2 \in E(H)$ (not necessarily distinct), let $\kappa(f, v, h_1, h_2)$ be the sum over all distinct paths P, Q such that P is an h_1-path and Q is an h_2-path of $\kappa(f, v, P, Q)$. We define the *crossing weight* of f to be $\sum_v \sum_{h_1, h_2 \in E(H)} \kappa(f, v, h_1, h_2)$[9].

We rely on the following lemma.

Lemma A.3 (No Double Crossing Lemma)**.** *Let f be a feasible flow that is minimum cost under unit edge costs. Suppose there exist support paths P, Q that cross twice. Let z be any node at which they cross. Then there exists a feasible flow f' that routes the same demands and such that $\kappa(f', v, h_1, h_2) \leq \kappa(f, v, h_1, h_2)$ for all $v \in V$ and $h_1, h_2 \in E(H)$, and such that $\kappa(f', z, h_P, h_Q) < \kappa(f, z, h_P, h_Q)$ where h_P and h_Q are the demand edges that P and Q route respectively.*

Proving this lemma is somewhat tedious, but such arguments have appeared in the literature before - the essence of the argument is in Lemma 1 of [17].

A consequence of this lemma is that by taking a minimum cost flow, which subject to that also minimizes the crossing weight, we may assume flow paths cross at most once.

We use contraction of edges of G in this proof. While we may write $h = (s, t)$ for a demand edge h with ends (s, t), we consider demand edges to have their own identity beyond their incident nodes. So for example, we could contract a supply edge incident with s and h automatically becomes incident with the contracted node (and also still t). The following lemma is easy to verify; informally it says that contracting supply edges never increases crossing weight.

[9] You could order the paths P, Q or the demands h_1, h_2 to avoid double-counting them in the sums, but it ultimately does not matter for this proof.

Lemma A.4. *Let f be a flow. Consider contracting a supply edge e, and let f' be the flow that arises performing the same contraction to support paths in f. Then $\kappa(f', v, h_1, h_2) \leq \kappa(f, v, h_1, h_2)$ for all $v \in V$ and $h_1, h_2 \in E(H)$.*

Now, let f a minimum cost flow, which subject to that also minimizes the crossing weight. For the sake of contradiction suppose two support paths P, Q cross at a node z. Let $h_P = (s_P, t_P)$ and $h_Q = (s_Q, t_Q)$ be their respective demand edges. Note that $z \notin \{s_P, t_P, s_Q, t_Q\}$. We assume $h_P \neq h_Q$. This assumption is without loss of generality, since if they are equal we can split them into parallel demand edges and attribute the path P to routing demand for h_P and Q to routing demand for h_Q.

Consider the embedding of h_P and h_Q so that each lies inside a face of G and they do not cross each other. Create a new multiflow instance as follows. In the face containing h_P, add leaf nodes s_P^*, t_P^* with supply edges $(s_P, s_P^*), (t_P, t_P^*)$. Move the endpoints of h_P to s_P^*, t_P^*. Similarly, in the face containing h_Q, add leaf nodes s_Q^*, t_Q^* with supply edges $(s_Q, s_Q^*), (t_Q, t_Q^*)$, and move the endpoints of h_Q to s_Q^*, t_Q^*. We can do this so the resulting supply graph plus h_P and h_Q is planar.

Extend the support paths of f for h_P, h_Q to the leaves to obtain a feasible flow f' for the new instance. Note that f' is also minimum cost for the new instance, under unit edge costs. This extension may create crossings, however only at s_P, s_Q, t_P, t_Q. More specifically, one can verify that for all v, h_1, h_2, $\kappa(f', v, h_1, h_2) = \kappa(f, v, h_1, h_2)$, except when 1) $v \in \{s_P, t_P\}$ and $h_P \in \{h_1, h_2\}$ or 2) $v \in \{s_Q, t_Q\}$ and $h_Q \in \{h_1, h_2\}$. Call such (v, h_1, h_2) *exceptional*. For exceptional triples only, we may indeed have that $\kappa(f', v, h_1, h_2) > \kappa(f', v, h_1, h_2)$.

Let P^* and Q^* denote the extensions of P, Q respectively in the new instance. We claim that P^* and Q^* cross at least twice - still at z and at least once more. To see this, consider the cycles $C(P^*)$ and $C(Q^*)$ respectively obtained by joining P^* with h_P and joining Q^* with h_Q. These are cycles in a planar graph that cross at z, and must cross at least one more time. This crossing cannot be at any of $s_P^*, t_P^*, s_Q^*, t_Q^*$, because P^* does not use s_Q^*, t_Q^* and Q^* does not use s_P^*, t_P^*. So it must be at some other node (in fact, it must be one of s_P, t_P, s_Q, t_Q), in which case P^* and Q^* also cross at this node. We can then apply Theorem A.3 to obtain a flow f'' such that $\kappa(f'', v, h_1, h_2) \leq \kappa(f', v, h_1, h_2)$ for all v, h_1, h_2, and such that $\kappa(f'', z, h_P, h_Q) < \kappa(f', z, h_P, h_Q)$.

Now, we apply Theorem A.4, contracting the leaf edges $(s_P, s_P^*), (t_P, t_P^*), (s_Q, s_Q^*), (t_P, t_P^*)$ to obtain a flow f''' such that $\kappa(f''', v, h_1, h_2) \leq \kappa(f''', v, h_1, h_2)$ for all v, h_1, h_2. Then f''' is a feasible flow for the original instance. Moreover, for any non-exceptional triple v, h_1, h_2, we have that

$$\kappa(f''', v, h_1, h_2) \leq \kappa(f'', v, h_1, h_2) \leq \kappa(f', v, h_1, h_2) = \kappa(f, v, h_1, h_2).$$

However, for the triple z, h_P, h_Q, we have that

$$\kappa(f''', v, h_1, h_2) \leq \kappa(f'', v, h_1, h_2) < \kappa(f', v, h_1, h_2) = \kappa(f, v, h_1, h_2).$$

For exceptional triples v, h_1, h_2, we unfortunately may have that $\kappa(f', v, h_1, h_2) > \kappa(f', v, h_1, h_2)$. However, for such triples we automatically get that $\kappa(f''', v, h_1, h_2) = 0$ because v is always an endpoint of either h_1 or h_2. Therefore, f''' has strictly less crossing weight than f. This is a contradiction, so no such crossing P, Q exist.

□

Note that the proof can be modified to produce an integral uncrossed flow if given an integral flow.

A.2 Pairwise-Planar and Series-Compliant Instances

One might ask what is in the class of pairwise-planar instances that is not fully-planar. We already mentioned the odd spindles [5, 1], which are interesting in that they are not cut-sufficient (i.e. the max-flow min-cut property does not hold), whereas fully-planar instances are cut-sufficient. Another interesting way to construct pairwise-planar instances is using series-parallel supply graphs. There are multiple ways to define series-parallel graphs, we take a bottom-up approach. Each series-parallel graph G has an ordered pair of

nodes $(s(G), t(G))$ associated with it (often these are called *terminals*, but we avoid this terminology to not add confusion with terminals of demands in multiflows). The class of series-parallel graphs is defined inductively, according to the following rules.

- *Single edges.* If G is a single edge (s, t) with distinct endpoints, then G is series-parallel with $s(G) = s, t(G) = t$.
- *Series composition.* Suppose $G_1 = (V_1, E_1), \ldots, G_k = (V_k, E_k)$ are series-parallel such that $t(G_i) = s(G_{i+1})$ for $i \in \{1, \ldots, k-1\}$ and $s(G_1), \ldots, s(G_k), t(G_k)$ are $k+1$ distinct nodes. Then $(V_1 \cup \cdots \cup V_k, E_1 \cup \cdots E_k)$ is series-parallel with $s(G) = s(G_1)$ and $t(G) = t(G_k)$.
- *Parallel composition.* Suppose $G_1 = (V_1, E_1), \ldots, G_k = (V_k, E_k)$ are series-parallel such that $s(G_i) = S(G_j)$ and $t(G_i) = t(G_j)$ for all i, j, and $s(G_1) \neq t(G_1)$. Then $G = (V_1 \cup \cdots \cup V_k, E_1 \cup \cdots E_k)$ is series-parallel with $s(G) = s(G_1)$ and $t(G) = t(G_1)$.

Note that more than one sequence of operations may lead to the same graph. We always assume we are working with a fixed sequence of series-parallel operations, we refer to this as the decomposition. A fixed decomposition gives a natural way to construct a planar embedding of G with $s(G)$ and $t(G)$ on the outside face: for a series-composition, assuming each G_i has $s(G_i)$ as its leftmost point and $t(G_i)$ as its rightmost, glue the graphs together left-to-right, and for a parallel-composition, glue the graphs together top-to-bottom. That is G_1 is embedded 'above' G_2 and so on.

When we perform a series composition, we declare all of the nodes in the set $\{s(G_i) : i \in \{1, \ldots, k\}\} \cup \{t(G_k)\}$ to be *series-siblings* of each other. A node may be involved in multiple series operations and will have siblings associated with each.

Definition A.5. We say (G, H) is *series-compliant* if G is series-parallel, and with respect to the decomposition used to build G, every demand (s, t) of H has s, t are series-siblings of G.

Series-compliant instances are a special case of *fully-compliant* instances defined in [5], and they capture most of the complexity of this family. If (G, H) is series-compliant, then one can show it is pairwise-planar when using the "natural" embedding of G. However, since the number of graphs k in a series operation can be large, we may have an arbitrarily large clique in H, and hence need not be fully-planar. It is also straightforward to embed a $K_{3,3}$-minor in $G + H$, as in Figure 4.

B NP-Hardness of Fractional Uncrossed Multiflow

In this section we prove the NP-hardness of deciding the existence of a fractional uncrossed flow satisfying all demands. The proof is reminiscent of a reduction by Middendorf and Pfeiffer [16], for the node-disjoint paths problem when $G + H$ is planar. We remark that this shows NP-completeness even for leaf instances of multiflows.

We refer to a fractional uncrossed flow which satisfies all demands as a *solution*. If all flow walks are (simple) paths, then we call it a *path-solution*; if they are all trails we call it a *trail-solution*. We rely on the following.

Theorem B.1. *Consider a planar congestion instance (G, H, u, d) where $d(h) \geq u(e)$ for all $e \in E(H)$ (including for example unit capacity and demands). If f is a feasible strongly uncrossed walk-multiflow, then through some short-circuiting operations on the support walks of f, we can obtain a feasible strongly uncrossed trail-multiflow f'.*

In the instance produced in the reduction, all capacities are 1, all terminals are leaves, and all other nodes have degree at most 3. Under these assumptions, any uncrossed walk-solution can be short-circuited to a trail-solution, and furthermore any trail-solution is a path-solution. So, the existence of a walk-solution, trail-solution, and path-solution are equivalent. Hence, we only consider path-solutions in our argument, and the hardness result automatically applies to all three variants.

In the next subsection, we introduce the key gadgets which are used in the proof, and in the subsection after we present the main proof.

B.1 Gadgets

Many of the ideas build on the structure of solutions for the instance in Figure 10 which we call Double Diamond.

Lemma B.2 (Double Diamond Instance)**.** *Let G, H be the instance of fractional uncrossed flow from Figure 10. This instance has only two possible path-solutions, and both are integral.*

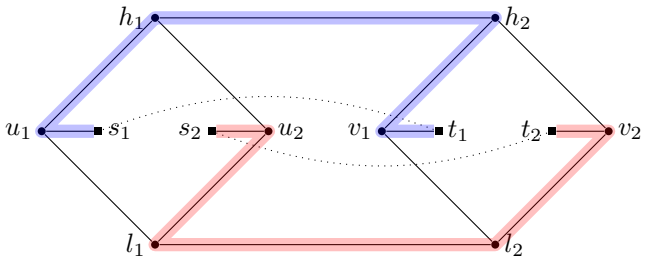

Figure 10: *Double Diamond Instance.* An instance for the fractional uncrossed flow problem that has only integral solutions (one of these solutions is given by the two colored paths with value 1).

Proof. Suppose there is a support path P in a fractional solution starting with $s_1u_1, u_1h_1, h_1u_2, u_2l_1$. Then, because the support path are uncrossed, there is no support path starting by s_2u_2, u_2h_1, as such a flow path Q would cross P at either the subpath $Z = u_2, h_1$ or $Z = u_2, h_1, u_1$. Then all support paths for the demand s_2t_2 uses u_2l_1, which, considering also P, implies that the total value of flow paths on that edge is more than 1, contradiction.

By similar arguments, one can show that the support paths use only one of the two highlighted paths from Figure 10 P_1 (in blue) and P_2 (in red) and their symmetric paths by a flip along an horizontal axis Q_1 and Q_2. To conclude, notice that P_1 crosses Q_2 and Q_1 crosses P_2, implying that the only two solutions are $P_1 + P_2$ and $Q_1 + Q_2$. □

We next discuss the Terminal Gadget, which is a modification of the Double Diamond Instance.

Lemma B.3 (Terminal Gadget, Figure 11)**.** *Let G, H be an instance of fractional uncrossed flow, with at least two demands s_1t_1, s_2t_2 as described in Figure 11 (modified from Figure 10). Let G_1 be the subgraph induced by the nodes outside the circle. Then, for any fractional solution z to G, H,*

(i) s_1t_1 is routed integrally inside G_1;

(ii) either all the support paths for s_2t_2 use c_1x or they all use d_1y;

(iii) no support path for any other demand uses an edge of G_1 (aside from s_1t_1, s_2t_2).

Proof. By considering the cut $\{c_1x, d_1y\}$, which is crossed by the demand s_2t_2, at most half a unit of s_1t_1-flow can be routed on these edges. This implies that there exists a support s_1t_1- path contained in G_1. There are 4 possible such paths, two of length 5 and two of length 7. Let $P = s_1u_1a_1u_2b_1d_1, v_1t_1$ and suppose that $z(P) > 0$. Since less than 1 unit of capacity is now available on u_2b_1 for s_2t_2, there must be an support s_2t_2-path Q starting with $s_2u_2a_1$. Hence Q must continue to u_1 and b_1, but this crosses P. Therefore $z(P) = 0$, and there must be a support s_1t_1-path P_1 of length 5. We may assume $P_1 = s_1u_1a_1c_1v_1t_1$.

Suppose that there is a support s_2t_2-flow path P_2, containing c_1x. Then as P_1 and P_2 do not cross, P_2 must contain a subpath $b_1u_1a_1c_1x$. Because $\{a_1c_1, b_1d_1\}$ is a tight cut and $a_1c_1 \in P_2$, P_2 does not contain b_1d_1. Hence P_2 starts with $s_2u_2b_1u_1 \ldots$ Because $z(P_2) > 0$ and $u_1a_1 \in P_2$, there is a support s_1t_1-path Q_1 starting with $s_1u_1b_1$. As P_2 and Q_1 do not cross, Q_1 continues with b_1u_2, but then it must cross P_2

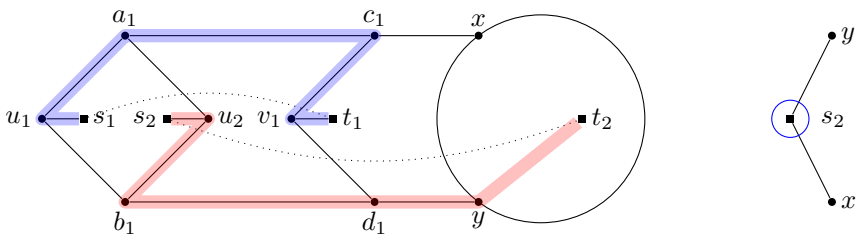

Figure 11: Terminal Gadget. A gadget that forces the demand s_2t_2 to be routed either entirely through x or through y. The nodes outside the circle form an induced subgraph, and deleting x and y disconnect these nodes from the rest of the graph. Furthermore, no demand other than s_1t_1 and s_2t_2 can have a support path intersecting the edges of the gadget. The colored paths represent a possible routing. On the right, a schematic to represent the gadget on a terminal, using a blue circle to denote that all the flow must use one of the two incident arcs.

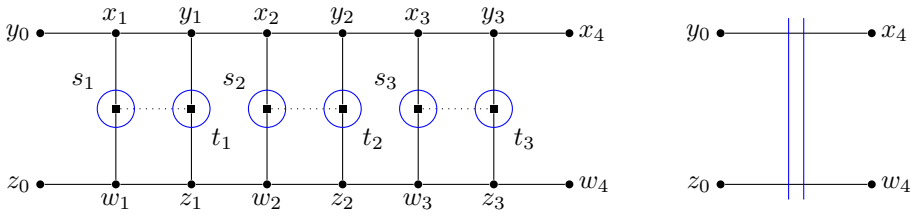

Figure 12: Coupled Arcs Gadget. A gadget that forbids the use of one arc among two, for any flow path from terminal outside the gadget. Recall that the circle refers to the schematic of the Terminal Gadget from Figure 11. On the right, a schematic representation of the Coupled Arcs Gadget.

at u_2. Hence no support s_2t_2-path contains c_1x, they must all contain d_1y. Furthermore, this shows that no support s_1t_1-path intersects the cut $\{c_1x, d_1y\}$, since s_2t_2 uses all the capacity of d_1y, and hence the support path would have to use c_1x twice. So the only possible support s_1t_1-paths are P_1 and its symmetric $P_1' = s_1u_1b_1d_1v_1t_1$.

Finally suppose that $z(P_1) > 0$ and $z(P_1') > 0$. As $P_1 \cup P_1'$ forms a circuit separating s_2 from t_2, any active flow s_2t_2-path would cross either P_1 or P_1'. Thus, (i) holds. Up to symmetry, we may assume $z(P_1) = 1$, $z(P_1') = 0$. From there, any active s_2t_2-flow path must start with $s_2u_2b_1d_1y$, that is (ii) holds, and (iii) follows. □

Lemma B.4 (Coupled Arcs Gadget, Fig 12)**.** *Let G, H be an uncrossable flow instance, containing the gadget of Figure 12 as an induced subgraph. Let P_{xy} and P_{wz} be the top and bottom paths $y_0x_1y_1x_2y_2x_3y_3x_4$ and $z_0w_1z_1w_2z_2w_3z_3w_4$ respectively. Then in any solution x, there is $Q \in \{P_{xy}, P_{wz}\}$ with the following property: for every external demand $st \in E(H) \setminus \{s_1t_1, s_2t_2, s_3t_3\}$ and for any support st-path P, the intersection of $E(P)$ with the gadget is either empty or exactly Q. Thus, this gadget behaves as though we have only two edges y_0x_4, z_0w_4, such that at most one of them can be used by external demands.*

Proof. We denote by P_i and Q_i the paths $s_ix_iy_it_i$ and $s_iw_iz_it_i$ respectively. We denote by $f_i(e)$ the flow value on edge e induced by the flow paths for s_it_i.

From Theorem B.3, for each of the demands $\{s_i, t_i\}$ ($i \in \{1, 2, 3\}$), either $f_i(s_ix_i) = 1$ (s_i is routed up) or $f_i(s_iw_i) = 1$ (s_i is routed down). Similarly each t_i is routed down or up. We say that s_it_i is *coherent* if s_i and t_i are routed in the same direction. Observe that if a support path for an external demand enters the gadget, it cannot use any edges incident with s_i or t_i for $i = 1, 2, 3$. Hence our goal is to that at least one of P_{xy} or Q_{xy} is not part of any support path for any external demand.

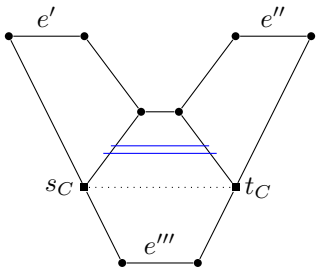

Figure 13: The Clause Gadget for a clause C. The two edges marked with a double blue line should be replaced by the gadget from Figure 12, allowing only one of these two edges to be used by the flow.

For any $r \in \{1, 2, 3\}$, if $s_r t_r$ is coherent (routing up, say), then any support $s_r t_r$-path is either exactly P_r or it starts with $s_r x_r y_{r-1} \dots y_0$ and ends with $x_4 y_3 \dots y_r t_r$. In the latter case we say the path *escapes*. If any coherent $s_r t_r$ does not have any escaping support paths (i.e., it routes entirely on P_r), then it is impossible for any external demand to route using P_{xy} if $s_r t_r$ routes up, and impossible to use P_{wz} if $s_r t_r$ routes down. In these cases, the lemma holds with $Q = P_{wz}$ in the former case and $Q = P_{xy}$ in the latter. So we may assume each coherent $s_r t_r$ has an escaping support path.

For any incoherent pair, all the support paths leave and re-enter the gadget, and therefore use up 2 units of capacity on the cut $\{y_0 x_1, z_0 w_1, y_3 x_4, z_3 w_4\}$. Hence if there are 2 incoherent pairs, then no external demand may cross through the gadget, in which case the lemma holds. We may thus assume that there are $i < j$ such that $s_i t_i$ and $s_j t_j$ are coherent. Let $k \in \{1, 2, 3\} \setminus \{i, j\}$.

Case 1: Suppose both $s_i t_i$ and $s_j t_j$ are routed in the same direction, say up. Since both $s_i t_i$ and $s_j t_j$ have a support path that escapes, these paths cross, a contradiction.

Case 2: The remaining case is when both $s_i t_i$ and $s_j t_j$ are routed in distinct directions, and $s_k t_k$ is not coherent (otherwise it would be in the same direction as $s_i t_i$ or $s_j t_j$ and the previous case applies). Without loss of generality, we may assume that $s_i t_i$ is routed up and $i < k$. To avoid crossing the escaping path for $s_i t_i$ at y_i, every support path for $s_k t_k$ must use $y_3 x_4$. However, this does not leave enough capacity for the escaping path for $s_i t_i$ that also uses this edge, a contradiction. □

By routing each demand up, or each demand down, one of the two paths P_{xy}, P_{wz} is free to be used by support paths from external demands.

Thus, the gadget from Figure 12 can be used to replace two edges e' and e'', with the effect of forbidding the use of both edges in the fractional uncrossed flow: it adds the constraint that either $x(e') = 0$ or $x(e'') = 0$.

Lemma B.5 (Clause Gadget, Figure 13)**.** *The clause gadget (Figure 13) has a solution even if we remove two of the three edges e', e'', e'''. It has no solution if we remove all three edges e', e'', e'''.*

Proof. It easily follows from Theorem B.4 by a simple case analysis, as only one of the two edges crossed by the double blue line can be used in routing the demand edge. □

B.2 Hardness of Finding Uncrossed Fractional Flows

Theorem B.6. *The problem Fractional Uncrossed Flow is NP-complete.*

Proof. We reduce from Planar 3-Satisfiability, in which the given 3-Sat instance has a planar variable-clause incidence graph. We may assume that each variable appears at most three times, and furthermore that it is not the same polarity in all of its incident clauses. From the variable-clause incidence graph G, we construct an uncrossed multiflow instance (G', H') as follows.

For each variable x, we replace the node for x in G with a 3-cycle, where the edges are labelled with the clauses incident with x as $e_{x,C}$. The clockwise order of the labels corresponds to the clockwise ordering of

the clauses about x in G. Now, we place a demand $s_x t_x$ between two nodes of the 3-cycle, so that the two $s_x t_x$ paths on the cycle correspond to the clauses where x appears positively and negatively. Call these two paths P_x and N_x respectively.

Now for each clause C, we replace the node for C in G with a clause gadget from Figure 13, where the edges e', e'' and e''' of the figure are labelled by the three variables occurring in C as $e_{C,x}$. The clockwise order of the labels corresponds to the clockwise ordering of the variables about C in G.

Now, consider each incident variable-clause pair (x, C). We replace the edges $e_{x,C}$ (from the 3-cycle of x) and $e_{C,x}$ (from the clause gadget of C) with a copy of the Coupled Arcs Gadget (Figure 12). By installing the Coupled Arcs Gadget in close proximity to the edges of the original graph G (recalling that the edge-labellings also mirror the original graph), we obtain a planar embedding of G'. We will abuse notation and still use $e_{x,C}$ and $e_{C,x}$ for the subdivided paths in the Coupled Arcs Gadget. By the properties of the Coupled Arcs Gadget, this has the effect that flow for demands outside of the Coupled Arcs Gadget can only use one of the edges $e_{x,C}$ or $e_{C,x}$. Additionally, support paths cannot "hop between" the two coupled edges, so the only two possible support paths for $s_x t_x$ are P_x and N_x. Similarly, any support path for the demand $s_C t_C$ of a clause gadget stays entirely inside the clause gadget.

If the satisfiability instance is satisfied by a truth assignment ϕ, then we can find an uncrossed (integral) routing. First, for each variable x, if $\phi(x) = \mathit{true}$, we set $f(N_x) = 1$, otherwise set $f(P_x) = 1$. This clearly satisfies all of the variable demands $s_x t_x$. We now extend this routing to satisfy the clause demands and the internal demands of the Coupled Arcs Gadgets.

Now, consider a clause C and its gadget. Some literal for a variable x in C is made true by the assignment. Without loss of generality, assume x occurs positively in C and $\phi(x) = \mathit{True}$. In our partial routing, $e_{x,C}$ is in P_x, and is not used by the routing for $s_x t_x$. Therefore we can route the demands from the Coupled Arcs Gadget using $e_{x,C}$. This leaves $e_{C,x}$ available to route the demand $s_C t_C$ and still satisfy the Coupled Arcs Gadget strictly inside the clause gadget (as in Lemma B.5). The other two Coupled Arcs Gadgets incident with the clause gadget for C can be routed inside the clause gadget using the other variable-labelled edges. This satisfies all of the demands that appear in clause gadgets and Coupled Arcs Gadgets. In this way, we find an integral uncrossed routing.

Conversely, suppose there is a fractional uncrossed flow f. For each variable x, either $f(P_x) > 0$ or $f(N_x) = 1$. Set $\phi(x) = \mathit{false}$ in the former case, and *true* otherwise. We claim that ϕ satisfies all the clauses. Consider a clause C with variables x_1, x_2, x_3 (with some polarity). One of the edges $e_{C,x_1}, e_{C,x_2}, e_{C,x_3}$ must be used by a support path for the demand $s_C t_C$. Say e_{C,x_i} is used. By the properties of the Coupled Arcs Gadget, this implies that the corresponding edge $e_{x_i,C}$ in the 3-cycle for x_i is saturated by the internal demands of the Coupled Arcs Gadget and is thus not available for routing $s_x t_x$. If x_i occurs positively in C, this means that $f(P_{x_i}) = 0$, and so $\phi(x_i) = \mathit{true}$, satisfying the clause. On the hand, if x_i occurs negatively in C, this means that $f(N_{x_i}) = 0$ and $f(P_{x_i}) = 1$, and so $\phi(x_i) = \mathit{false}$, satisfying the clause. As this holds for any clause, ϕ is a satisfying assignment, concluding the proof of reduction. □

B.3 Exists-Embedding Uncrossed Multiflow

Theorem B.7. *It is NP-hard to determine whether there is an embedding of G in which (G, H) has a fractional uncrossed multiflow.*

Proof. We reduce from the fixed-embedding version of fractional uncrossed multiflow. Start with a fixed embedding of G. Consider the following operation: add an edge $e = (u_1, u_2)$ between two nodes u_1, u_2 on the same face, then subdivide it into a path $P = u_1, v_1, v_2, u_2$, and add demand edges $u_1 v_1, u_2 v_2$. The new supply and demand edges have unit weight.

Claim B.8. *(G, H) has an uncrossed solution (in the original embedding) if and only if the new instance has an uncrossed solution.*

Proof. It is straightforward to verify that if (G, H) has an uncrossed solution, then so does the new instance. We show the converse, that if the new instance has an uncrossed solution, so too does (G, H). Consider an

arbitrary uncrossed solution f for the new instance. It suffices to show that there is an uncrossed solution where the demand u_iv_i is routed entirely on the supply edge u_iv_i for $i = 1, 2$.

Observe that, in f, all flow for u_1v_1 that is not routed through u_1v_1 must be routed through u_2v_2. So, $f^{u_1v_1}(u_1v_1) + f^{u_1v_1}(u_2v_2) = 1$. Similarly, $f^{u_2v_2}(u_2v_2) + f^{u_2v_2}(u_1v_1) = 1$. So, between the two new demands, all capacity on the new supply edges u_1v_1, u_2v_2 is used. Therefore, the demands of the original demand graph H are routed (uncrossed) inside the original supply graph G. So (G, H) has an uncrossed solution. ◇

We then repeat this operation until the supply graph is a (subdivision of) a 3-connected graph, which has a unique embedding. The claim implies that the original instance (G, H) has an uncrossed solution (in its given embedding) if and only if the new instance has an uncrossed solution in its embedding, which by the uniqueness of the embedding is equivalent to the new instance having an uncrossed solution in some embedding. □

C Colouring Uncrossed String Graphs

In this section, we prove the following result.

Theorem C.1. *Let U be an uncrossed string graph arising from a realization with at most k paths on any edge. Then $\chi(U) = O(k^2 \log |E(U)|)$.*

The following is the key lemma.

Lemma C.2. *Let U be an uncrossed string graph with a load k realization $(G, \mathcal{P})$. Then U is the union of $O(k^2 \log |E(U)|)$ planar graphs.*

Proof. We generate a random subgraph $H = (S, F)$ of U as follows. First, we select random subset $S \subseteq \mathcal{P}$ by drawing each $P \in \mathcal{P}$ independently and uniformly at random with probability $q = \frac{1}{k}$. Now, for each $e \in E(G)$ such that $|S \cap \mathcal{P}_e| = 2$ exactly, we add to F an edge between the two members of $S \cap \mathcal{P}_e$. Note that H is planar.

Let $PP' \in E(U)$, and consider the event that $PP' \in E(H)$. This event happens if there exists some e with $P, P' \in \mathcal{P}_e$ where $P \in S$, $P' \in S$, and no other path in $\mathcal{P}_e$ is in S. Clearly there is at least one e such that $P, P' \in \mathcal{P}_e$, and we lower bound the probability that PP' is included due to this particular edge. Then,

$$\Pr[PP' \in E(H)] \geq q \cdot q \cdot (1-q)^{k-2} \geq \frac{1}{k^2}\left(1 - \frac{1}{k}\right)^{k-2} \geq \frac{1}{ek^2},$$

where the last inequality follows from the fact that $(1 - 1/x)^{x-2} \geq 1/e$ for all $x > 1$.

We repeat this process $C = 2ek^2 \log |E(U)|$ times, independently, obtaining random subgraphs $H_1, \ldots, H_C$ of U. For $PP' \in E(U)$, let $A_{PP'}$ be the event that for all $i \in [C]$, $PP' \notin E(H_i)$. Then,

$$\Pr[A_{PP'}] \leq \left(1 - \frac{1}{ek^2}\right)^{2ek^2 \log |E(U)|} \leq \left(\frac{1}{e}\right)^{2 \log |E(U)|} = |E(U)|^{-2},$$

where the second inequality follows from the fact that $(1 - 1/x)^x \leq 1/e$ for $x > 1$.

Let A be the event that there exists $PP' \in E(U)$ such that for all $i \in [C]$, $PP' \notin E(H_i)$. That is, there exists $PP' \in E(U)$ such that $A_{PP'}$ holds. By a basic union bound, $\Pr[A] \leq |E(U)| \cdot |E(U)|^{-2} < 1$. So there is an outcome in which A does not hold. In this outcome, U is covered by the graphs $H_1, \ldots, H_C$. Hence U is the sum of C planar graphs. □

From this, Theorem C.1 follows easily.

Proof of Theorem C.1. We show that U is $O(k^2 \log |E(U)|)$-degenerate, from which the result follows. As the property in the assumption of the theorem is a hereditary graph property, it suffices to show that U contains a node with at most $O(k^2 \log |E(U)|)$ neighbours.

By Theorem C.2, U is the sum of $O(k^2 \log |E(U)|)$ planar graphs. Each planar graph contributes at most $3|V(U)|$ parallel edge classes, so it follows that $E(U)$ contains at most $O(k^2 |V(U)| \log |E(U)|)$ parallel edge classes. Hence it contains a node with at most $O(k^2 \log |E(U)|)$ neighbours. □