An Approximation Algorithm for the Disjoint Paths Problem in Even-Degree Planar Graphs

Jon Kleinberg *

Abstract

In joint work with Éva Tardos in 1995, we asked whether it was possible to obtain a polynomial-time, polylogarithmic approximation algorithm for the disjoint paths problem in the class of all even-degree planar graphs [19]. This paper answers the question in the affirmative, by providing such an algorithm. The algorithm builds on recent work of Chekuri, Khanna, and Shepherd [7, 8], who considered routing problems in planar graphs where each edge can carry up to two paths.

1 Introduction

The problem of connecting nodes in a network via disjoint paths is a basic algorithmic question for graphs. In addition to its applications in such areas as network routing [4, 28] and VLSI layout [16, 23, 29], it is also a fundamental issue in graph theory, with developments on disjoint paths problems often proceeding in close connection with general structural results concerning graphs [12, 34].

In this paper we consider the following well-studied optimization version of the disjoint paths problem: given an undirected graph G, and a collection $\mathcal{T}$ of pairs of nodes in G – the *terminal pairs* – we wish to connect as many terminal pairs as possible using paths that are mutually edge-disjoint. (If all terminal pairs can be connected by edge-disjoint paths, we say the instance is *realizable*.) This is one of Karp's original NP-complete problems [15]; due to the problem's intractability, and its role in applications, attention has turned to the design of approximation algorithms: a *c-approximation algorithm* for this problem is a polynomial-time algorithm that routes at least $\frac{1}{c} \cdot \text{OPT}(G, \mathcal{T})$ terminal pairs using edge-disjoint paths, where $\text{OPT}(G, \mathcal{T})$ is the maximum possible.

*Department of Computer Science, Cornell University, Ithaca NY 14853. Email: kleinber@cs.cornell.edu. Supported by a David and Lucile Packard Foundation Fellowship and NSF grants CCF-0325453, IIS-0329064, and CCR-0122581; work done in part while on sabbatical leave at Carnegie Mellon University.

Despite a significant amount of research, there are wide gaps in our understanding of the approximability of the disjoint paths problem. It is known that no polynomial-time algorithm can achieve an approximation guarantee of $O(m^{1/2-\varepsilon})$ on *directed graphs* with m edges (unless $P = NP$) [14], but this result is intrinsically based on intractabilities for the directed case (specifically, hardness with just two terminal pairs) that do not have analogues in the undirected case. For undirected graphs, the strongest hardness-of-approximation bounds are very recent results of Andrews and Zhang and Chuzhoy and Khanna, leading to a lower bound of $\Omega((\log m)^{1/2-\varepsilon})$ [2, 9]; and it is entirely plausible — though quite out of reach using current techniques — that there may be a corresponding polylogarithmic upper bound for all undirected graphs as well. Currently, however, the only classes of undirected graphs for which constant or polylogarithmic approximation algorithms are known are the classes of trees with parallel edges [13], expanders [18, 21], and grids and grid-like graphs [3, 4, 19, 20].

The approximation results for trees are based on the highly simplified structure of the underlying graph, and the results for expanders are based on existence results asserting that every instance of the disjoint paths problem in an expander has a large optimum [5, 22]. The results for grids and grid-like graphs, on the other hand, make use of a technique that can potentially be applied much more broadly: relating the optimum to its *fractional relaxation*. In the fractional variant of the problem (a version of the *multicommodity flow problem*), each terminal pair can route a real-valued amount of flow between 0 and 1, and this flow can be split fractionally across a set of distinct paths. If we denote the maximum value of a solution to the fractional relaxation by $\text{OPT}^*(G, \mathcal{T})$, then this provides a tractable upper bound on the true optimum $\text{OPT}(G, \mathcal{T})$ — a fact that provides significant analytical leverage when $\text{OPT}(G, \mathcal{T})$ and $\text{OPT}^*(G, \mathcal{T})$ are close in value.

The Class of Even-Degree Planar Graphs. Short of a result for all undirected graphs, what is a broad intermediate class of graphs for which we might hope to obtain strong

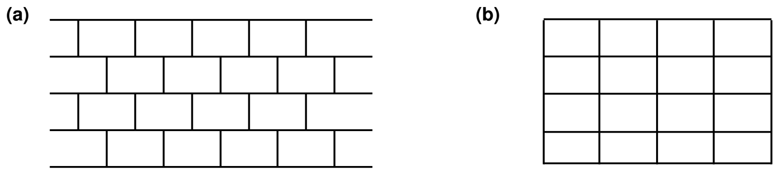

Figure 1. The brick-wall graph (a) and the grid (b) are structurally very similar, differing primarily in the degrees of their internal nodes (three versus four respectively).

approximation results? In our work with Éva Tardos on disjoint paths in grid-like graphs [19, 20], we made extensive use of structural properties of grids (the quadratic growth rate of balls, for example), but we observed that the success of the algorithm seemed to depend most intrinsically on two much more basic properties of grids:

(i) that they are planar graphs, and, more subtly,

(ii) that (almost) all their nodes have even degree.

Motivated by this, we say that an *even-degree planar graph* is one in which each node's degree is divisible by two.[1] We will also refer to even-degree graphs as *Eulerian*.

To understand the role of this even-degree condition in the disjoint paths problem, it helps to consider three points. First, a canonical approach to routing on grids is to produce large sets of mutually crossing paths, which can act as "switching" structures for connecting up terminal pairs. On a graph whose nodes have degree three, edge-disjoint paths cannot cross, and so such an approach clearly cannot be applied. Second, the gap between the true optimum $\text{OPT}(G, \mathcal{T})$ and its fractional relaxation $\text{OPT}^*(G, \mathcal{T})$ is bounded by a polylogarithmic factor in grids [19, 20], but there are instances on 3-regular planar graphs for which it can be as large as $\Omega(\sqrt{n})$, providing essentially no help in bounding the optimum. And third, there are many exactly solvable special cases known for the disjoint paths problem in planar graphs [12], but almost all require some type of evenness assumption on the node degrees; thus, assuming even degrees gives one access to this body of results.

To appreciate these issues in a very concrete setting, consider the disjoint paths problem on the *brick wall graph* pictured in Figure 1(a). (A general brick wall graph has n rows and n columns; the figure shows $n = 5$.) Structurally, this graph is extremely similar to the $n \times n$ grid, but because its internal nodes have degree three instead of four, one cannot use structures based on crossing paths for routing, and one also cannot usefully relate the optimal solution to its fractional counterpart. In fact, obtaining a polylogarithmic approximation algorithm for disjoint paths on the brick wall graph is an open problem that appears to be quite difficult; it is very close in nature to the comparably hard node-disjoint paths problem for grids.[2]

Based on considerations like this, we posed the following open question in [19]:

> $(*)$ Is it possible to obtain a polynomial-time, polylogarithmic approximation algorithm for the disjoint paths problem in the class of all even-degree planar graphs?

Our Results. In this paper, we settle this question in the affirmative, providing an $O(\log^2 n)$-approximation algorithm for the disjoint paths problem in even-degree planar graphs. Our approximate solution is in fact within an $O(\log^2 n)$ factor of the fractional optimum, showing the requirement of even degrees is all that is needed to reduce the general polynomial gap between the fractional and true optima in planar graphs down to a polylogarithmic factor. (The algorithm can in fact handle planar graphs in which a single face may contain nodes with odd degrees; in this way, the class of graphs under consideration includes the grid itself.)

Our algorithm builds on very interesting recent work of Chekuri, Khanna, and Shepherd [7, 8], who considered the routing problem with capacity two. This can be viewed as the variation of the disjoint paths problem in which each edge is allowed to carry two paths, rather than just one; Chekuri et al. gave an $O(\log n)$-approximation algorithm (henceforth referred to as the CKS algorithm) for routing with capacity two in arbitrary planar graphs. This was considerably stronger than previous results, which required edge capacity $\Omega(\log n / \log\log n)$ in order to achieve comparable guarantees (albeit in arbitrary graphs) [30, 31].

[1] While the grid does not completely satisfy this, due to the degree-three nodes on its boundary, we will see later that this is a type of exception that can be handled within our main result.

[2] It is natural to ask at this point why we do not consider the weaker condition that the minimum degree be four — at a superficial level, this too rules out the problems posed by degree-three nodes. But in fact this weaker restriction would not gain us anything. Consider an instance of the disjoint paths problem on an arbitrary planar graph G that may have degree-three nodes, and attach by two edges to each node in G a constant-sized planar graph of minimum degree five. This new graph G' has minimum degree five, but the resulting instance of the disjoint paths problem is clearly equivalent to the original one in G. Due to examples like this, it appears that restrictions weaker than the even-degree condition are too "brittle" to avoid the qualitative problems associated with degree-three nodes.

There is a natural connection between the CKS algorithm and question $(*)$, which can be appreciated as follows: One way to represent a planar graph G with edge capacity two is to replace every edge of G with two parallel copies, obtaining a planar graph G'. The disjoint paths problem on G' is then equivalent to the routing problem G with capacity two; and clearly any such graph G' has even node degrees (since all its edges come in pairs). Of course, such graphs form a particular special case of the class of all even-degree planar graphs; thus, our result can be interpreted in this way as a generalization of the capacity-two case. (One should note, however, that our bound for this more general problem is weaker by a logarithmic factor: $O(\log^2 n)$ instead of $O(\log n)$.)

If one considers the disjoint paths problem in terms of its motivating routing applications, it is natural to ask how crucial disjointness is, compared with relaxed constraints like capacities of two. The answer seems to vary depending on the application. For settings in which each connection consumes a large fraction of the bandwidth on a link, a capacity of two may be a reasonable assumption – essentially, one is positing that each connection uses at most half the bandwidth. In other settings, however, the disjointness requirement is motivated by specific circuit-switched models, by fault-tolerance, or by the need for failure-independence among connections, and here it differs in qualitative ways from the condition of a small edge capacity $c > 1$.

Ultimately, however, as suggested above, understanding the tractability of the disjoint paths problem is also a fundamental issue in graph algorithms, and progress in mapping out the boundary between approximability and inapproximability is a crucial issue in this light as well. Our result here illustrates the surprisingly powerful role that the even-degree condition plays for approximability in planar graphs, enabling the use of fractional relaxations and leading to strong performance guarantees.

2 Preliminaries and Overview

Grids and Crossbars. In addition to their importance as a special case, grid graphs have played a significant role in the development of disjoint paths algorithms due to the special routing properties they exhibit. One key property is their usefulness as "crossbar" structures: any instance of the disjoint paths problem on an $n \times n$ grid in which the terminals reside at distinct nodes in the first column is realizable. (Essentially, one routes each terminal pair out to a distinct column, and then connects them up there — hence the term "crossbar," to suggest that the grid can act as a large "switching station" for terminals on its boundary.) As a result of this simple observation, many disjoint paths algorithms try to identify a "grid-like" subgraph in the input graph, route many terminals to the boundary of this subgraph, and then use the subgraph as a crossbar to link them up inside [3, 7, 8, 17, 19, 20]. This general strategy helps form the basis for our algorithm here as well.[3]

The CKS algorithm. Our algorithm builds on the CKS algorithm, adapting it to make use of the even-degree condition so as to produce paths that are edge-disjoint. It is therefore useful to briefly review their approach first. The crucial ingredient in the CKS algorithm is the following. For a set of nodes $S \subseteq V$, let $\delta(S)$ denote the set of edges with exactly one end in S. We say that a set of nodes Z in G is *well-linked* if for every set S containing at most half of Z, we have $|\delta(S)| \geq |S \cap Z|$. Chekuri et al. show that for any input $(G, \mathcal{T})$, one can efficiently compute node-disjoint induced subgraphs $G_1, \ldots, G_r$ of G, and corresponding disjoint subsets $\mathcal{T}_1, \ldots, \mathcal{T}_r$ of $\mathcal{T}$, so that (a) the terminal pairs in $\mathcal{T}_i$ belong to G_i; (b) the members of the terminal pairs in $\mathcal{T}_i$ are well-linked as a set of nodes in G_i; and (c) the total size of the sets $\mathcal{T}_i$ is at least $\text{OPT}^*(G, \mathcal{T})/\lambda(n) \geq \text{OPT}(G, \mathcal{T})/\lambda(n)$, for a polylogarithmic function $\lambda(n)$. In [8], it is shown that one can bound $\lambda(n)$ by $O(\log n)$. We will leave the value of the function $\lambda(n)$ implicit, since improvements in this bound translate directly to improvements in the performance guarantee of the overall algorithm (as they do in [7, 8] as well). We will refer to each instance $(G_i, \mathcal{T}_i)$ as a *well-linked instance*, and the division of $(G, \mathcal{T})$ into these instances as a *well-linked decomposition*.

By standard reduction techniques [7, 12], one can also modify all instances so that the maximum node degree is four (and this can be done in such a way that no odd-degree nodes are introduced).

Given a well-linked instance $(G_i, \mathcal{T}_i)$, with G_i planar, Chekuri et al. [7] adapt a technique of Robertson, Seymour, and Thomas [35] to construct a minor of G_i isormorphic to an $\Omega(|\mathcal{T}_i|) \times \Omega(|\mathcal{T}_i|)$ grid.[4] By routing $\Omega(|\mathcal{T}_i|)$ terminals to the boundary of this grid minor, and then using the grid minor itself as a crossbar structure, they are able to route a constant fraction of all terminal pairs in the instance. A capacity of two is required because they are using a grid minor rather than an actual grid subgraph; for example, if the grid minor is built from a brick wall graph in G_i, then clearly some edges of G_i will in fact carry two paths when

[3]Although it is not crucial to our discussion here, it is worth noting that grid structures also play a central role in Robertson and Seymour's algorithm for the disjoint paths problem with a constant number of terminal pairs [34], but for a slightly different reason: Rather than using a grid as a crossbar directly, they implicitly use its crossbar properties to declare certain internal nodes of a large grid structure "irrelevant" to the solvability of the input instance.

[4]For graphs G and H, we say that G contains an *H-minor* if one can identify disjoint connected subgraphs of G ("super-nodes") corresponding to the nodes of H, such that two super-nodes have an edge of G connecting them whenever the corresponding nodes in H are joined by an edge. We give a more extensive discussion of minors in Section 4.

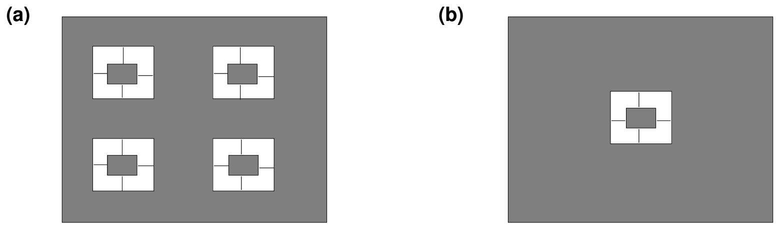

Figure 2. In these examples, a well-linked decomposition must separate the weakly-attached subgraphs from the larger enclosing graph, producing odd-degree nodes in the process.

used as part of this crossbar.

Making Use of the Even-Degree Condition. Intuitively, one should use the even-degree condition on G_i to construct a better type of grid. For example, if G_i contained a subgraph H actually isomorphic to a grid, or at least to a subdivision of a grid (i.e., a grid in which each edge is replaced by a chain of degree-two nodes), then one could use H as a crossbar in which the paths remained edge-disjoint. While our algorithm follows this intuition in a very general sense, the situation is in fact more complex than this for two fundamental reasons.

(i) First, an even-degree planar graph may contain a large well-linked set of nodes and yet not contain a subdivision of a grid of more than constant size. (For example, consider a copy of the brick wall graph in which every alternate edge in each row is replaced with two parallel copies.) So if we want to use a stronger type of grid structure, we need to look for something more complicated than a subdivision.

(ii) Second, we cannot in fact assume that each G_i in a well-linked decomposition has even degrees: the process of dividing G into the subgraphs G_i will change the degrees of nodes in hard-to-control ways.

We deal with the first of these problems by defining a structure that we call a *transparent minor* — roughly, this is a minor whose super-nodes are well-connected enough that instances of the disjoint paths problem defined inside them are realizable. Essentially by definition, a transparent grid minor can be used as a crossbar with edge-disjoint paths; and we show that an even-degree planar graph with a well-linked set of size k contains a transparent minor isomorphic to an $\Omega(k) \times \Omega(k)$ grid. We do this by adapting the technique of Robertson, Seymour, and Thomas mentioned above, combining it with the Okamura-Seymour theorem [27] to produce edge-disjoint paths that cross extensively.

However, as noted in (ii), the graphs G_i may not in fact be even-degree, and we describe next how to deal with this further difficulty.

Handling Odd-Degree Nodes. To understand how odd-degree nodes can arise in G_i — and to see that they cannot be avoided just by choosing a well-linked decomposition carefully — consider the examples illustrated schematically in Figure 2: These examples are constructed by taking a grid-like planar graph, removing one or more subgraphs from the interior, and reconnecting these subgraphs by very sparse sets of edges to the rest of the graph. With terminals defined appropriately, any well-linked decomposition must separate out these subgraphs, and thereby introduce odd-degree nodes. As a richer example, consider a recursive version of Figure 2(a): within each of the four weakly-attached subgraphs, there are four more weakly-attached subgraphs each, and four more within each of these, and so forth recursively. Any well-linked decomposition may have to break apart all these subgraphs along the sparse cuts.

Our goal will be to try partially "restoring" the even-degree condition, by modifying each subgraph in the well-linked decomposition. The challenge here is that these modifications may destroy the property that all subgraphs are node-disjoint, but we will try to control how they overlap so that a large subset of them remains mutually disjoint. In the recursive version of Figure 2(a), we could carry out these modifications as follows: for each subgraph G_i at level ℓ of the recursion, we "fill back in" all its weakly-attached subgraphs from deeper levels (but don't introduce any new terminals). We thus have a well-linked instance $(G'_i, \mathcal{T}_i)$ with odd-degree nodes only on the outer face, so we can use an extension of the Okamura-Seymour theorem due to Frank [11, 12] to construct a large transparent grid minor in G'_i. Moreover, the subgraphs G'_i at a fixed level remain node-disjoint, so we can produce paths in all of them independently; and since there are only $O(\log n)$ levels of recursion, we can work just with the level on which we route the most terminal pairs and lose only a further $O(\log n)$ factor in the approximation.

This won't work for the recursive version of Figure 2(b), however, where we introduce just one internal weakly-attached subgraph at each level of the recursion. Here the recursion goes on for $\Theta(n)$ levels, and filling in the nodes at level ℓ interferes with the routing at all deeper levels. But here, things work out for the following completely different reason. The recursion runs very deeply because there is only one internal subgraph per level, and so the subgraphs G_i in the well-linked decomposition have odd nodes on only two faces. We therefore use Okamura's extension of the Okamura-Seymour theorem [26], which can be adapted to allow odd-degree nodes on up to two faces, and we construct large transparent grid minors in all the subgraphs G_i without having to modify them.

Of course, this discussion has been in terms of the single pair of recursive examples derived from Figure 2. But it gives the basic idea in general: for an arbitrary planar graph, we define a notion of "level," and take advantage of the trade-off between having a small number of levels (as in (a)) and having many subgraphs G_i with odd-degree nodes on only two faces (as in (b)).

Outline. The remainder of the paper is organized as follows. In Section 3, we show how to implement the approach just discussed, partitioning the instances and partially restoring the evenness condition. In Section 4, we provide basic definitions and properties of transparent minors, and related structures that we call *weavings*. In Sections 5 and 6, we describe how to construct large transparent grid minors in the subgraphs arising from the well-linked decomposition, as modified by the construction in Section 3. Finally, we summarize the full routing algorithm in Section 7.

3 Partially Restoring the Evenness Condition

Suppose we are given a well-linked decomposition of an instance $(G, \mathcal{T})$, where G is an even-degree planar graph. Let $\mathcal{G}$ denote the set of all subgraphs G_i in this decomposition; we may assume that each $G_i \in \mathcal{G}$ is connected. We wish to establish the following fact.

Theorem 3.1 *There is a polynomial-time algorithm to construct a set of subgraphs $\{G'_i : G_i \in \mathcal{G}\}$, and partition these subgraphs into $O(\log n)$ classes, such that*

(i) G_i is a subgraph of G'_i, which in turn is a subgraph of G.

(ii) Each G'_i has nodes of odd degree on at most two faces, and

(iii) All the subgraphs G'_i in a single class are node-disjoint.

Note that since each $(G_i, \mathcal{T}_i)$ is a well-linked instance, so is $(G'_i, \mathcal{T}_i)$ (as we produce the new instance $(G'_i, \mathcal{T}_i)$ by adding nodes and edges to the graph, but not defining any new terminals). In the following sections, we show how to take a well-linked instance $(G'_i, \mathcal{T}_i)$, where G'_i satisfies (ii), and route a constant fraction of all its terminal pairs. Since we may do this for all the instances in a single class simultaneously (by (iii) their underlying graphs are disjoint), and since one of the classes contains at least $\text{OPT}(G, \mathcal{T})/(\lambda(n) \log n)$ terminal pairs, we thereby obtain an approximation guarantee of $O(\lambda(n) \log n) = O(\log^2 n)$ by simply doing this for each class, and taking the one in which we route the most terminal pairs.

We now proceed to prove Theorem 3.1. We fix a drawing of the graph G in the plane $\mathbf{R}^2$; we will also use G to denote to the drawing of the graph, when there is little risk of confusion. A *face* of G is a connected component of $\mathbf{R}^2 \setminus G$; the *outer face* of G is the unbounded component of $\mathbf{R}^2 \setminus G$, and all other faces are *internal*. Each face is bordered by the set of nodes and edges incident to it, which are naturally ordered in a *facial walk*: this is a cycle that may repeat nodes and edges.

From a drawing of G, we can define a drawing for each subgraph $G_i \in \mathcal{G}$, obtained by simply deleting all nodes in $G \setminus G_i$. Now, every internal face Γ of G_i is either also a face of G, or else it is the result of deleting some number of nodes of $G \setminus G_i$ that were drawn inside Γ. In the former case, we will refer to Γ as a *basic face* of G_i, and in the latter case we will refer to it as an *exceptional face*. (We will not refer to the outer face of G_i as either basic or exceptional.) We note the following simple observation.

Lemma 3.2 *If a node v has odd degree in G_i, then it is incident to the outer face of G_i or to at least one exceptional face.*

As suggested earlier, we will define each subgraph G'_i by "filling in" the nodes missing from some of the exceptional faces of G_i. We now investigate how this interferes with the node-disjointness condition, by defining the following partial order. Given subgraphs G_i and G_j, since they are connected and node-disjoint, it is either the case that one is drawn inside an exceptional face of the other, or each is drawn in the outer face of the other. We define a partial order on the subgraphs in the decomposition, writing $G_i \preceq G_j$ if G_i is drawn inside an exceptional face of G_j. Here is the key property of this partial order.

Lemma 3.3 *If G_i, G_j, and G_k are subgraphs such that $G_i \preceq G_j$ and $G_i \preceq G_k$, then G_j and G_k are comparable with respect to $\preceq$.*

This lemma allows us to represent the partial order $\preceq$ on $\mathcal{G}$ as follows. We say that a partial order $\sqsubseteq$ on a set U is *tree-representable* if there exists a rooted forest H with node set U, and connected components equal to rooted trees $T_1, \ldots, T_r$, such that $u \sqsubseteq v$ if and only if u and v belong to the same component and u is a descendent of v. By Lemma 3.3, we have

Lemma 3.4 *The partial order $\preceq$ on $\mathcal{G}$ is tree-representable.*

Let $T_1, \ldots, T_r$ be the trees in the representation provided by Lemma 3.4. We now use the structure of these trees to guide the construction of the subgraphs G'_i and their partition into a small number of classes.

The main tool for this is the following simple decomposition result for trees; it is reminiscent of the *caterpillar decomposition* [24, 25], though different in its specifics. Given a rooted tree T, and a subset of nodes X, we use $T[X]$ to denote the rooted forest induced on the nodes of X; we say that $T[X]$ is a *rooted path* if it is a connected subset of a single root-to-leaf path in T.

Lemma 3.5 *Given a rooted tree T with n nodes, one can partition its nodes into sets $X_1, X_2, \ldots, X_\ell$ such that $\ell = O(\log n)$, and for all i we have the following:*

(i) Each component of $T[X_i]$ is a rooted path.

(ii) If $u, v \in X_i$ are such that one is a descendent of the other in T, then they belong to the same component of $T[X_i]$.

The proof is omitted due to lack of space. Essentially, the construction that proves the lemma is as follows. Viewing T as an undirected graph, we can find a *separator node*: a node v such that no component of $T \setminus \{v\}$ has more than $n/2$ nodes. We define X_1 to be any root-to-leaf path containing v. In general, having produced $X_1, \ldots, X_b$, we define $T_b = T \setminus (X_1 \cup \cdots \cup X_b)$. We apply the above construction in each component of T_b, producing one rooted path in each component that together form the set X_{b+1}.

We now prove Theorem 3.1 by applying this decomposition result to the tree representation of $\preceq$.

Proof Sketch for Theorem 3.1. As above, let $T_1, \ldots, T_r$ be the trees in the representation of the partial order $\preceq$ provided by Lemma 3.4. We apply Lemma 3.5 to each tree T_a, producing sets $X_{a,1}, X_{a,2}, \ldots, X_{a,\ell(a)}$. Let $\ell = \max_a \ell(a)$, and define $Y_b = \cup_a X_{a,b}$ for $b = 1, 2, \ldots, \ell$. We note that $\ell = O(\log n)$.

Observe that for each b, the set Y_b consists of a collection of rooted paths in the partial order, with elements in different paths incomparable. It follows that for any subgraph $G_i \in Y_b$, all the subgraphs $G_j \in Y_b$ for which $G_j \preceq G_i$ are drawn inside a single exceptional face Γ_i of G_i. We define G'_i by adding to G_i all nodes drawn inside all exceptional faces other than Γ_i. Thus, Γ_i is the only exceptional face of G'_i.

One can now verify that this collection of subgraphs satisfies the conditions of the theorem. ■

Following the plan suggested at the beginning of this section, we will assume in the following sections that we are dealing with a single instance $(G, \mathcal{T})$, where $\mathcal{T}$ is a well-linked set of k terminals and G is a planar graph (of maximum degree four) with odd-degree nodes on at most two faces.

4 Transparent Minors and Weavings

Transparent Minors. Here we define the notion of a *transparent minor* of a graph. First, we use a slightly unusual definition of a *minor* of a graph that is equivalent to the standard one: we say that H is a *minor* of a graph $G = (V, E)$ if there exist

- a collection of *super-nodes* — disjoint sets of nodes in G corresponding to the nodes of H, denoted $\{S(\alpha) : \alpha \in V(H)\}$, such that each subgraph $G[S(\alpha)]$ is connected; and
- a collection of *super-edges* — paths in G corresponding to the edges of H, denoted $\{P(\alpha, \beta) : (\alpha, \beta) \in E(H)\}$. Each path $P(\alpha, \beta)$ should have its endpoints in $S(\alpha)$ and $S(\beta)$, and its internal nodes should be disjoint from all the sets $\{S(\alpha) : \alpha \in V(H)\}$ and from all other paths $P(\gamma, \delta)$.

We will refer to the subgraph of G consisting of all super-nodes and super-edges as an *H-minor* in G.

Informally, a transparent minor of G is one in which the super-nodes are sufficiently well-connected that one can solve instances of the disjoint paths problem inside them. We say that a node of $S(\alpha)$ is a *port* if it is an endpoint of one of the incident super-edges $P(\alpha, \beta)$. For a node α in H, we let $d(\alpha)$ denote the degree of α. Now, we say that H is a *transparent minor* of G if H is a minor of G, and each super-node $S(\alpha)$ satisfies the following condition:

- Let $\mathcal{T}$ be a set of $\leq \lceil d(\alpha)/2 \rceil$ terminal pairs in $S(\alpha)$ such that at most one member of the multiset $\{s_1, \ldots, s_k, t_1, \ldots, t_k\}$ is not a port. Then $\mathcal{T}$ is realizable in $G[S(\alpha)]$.

It is easily checked that if G contains a subgraph isomorphic to a subdivision of H, then G contains a transparent H-minor; but as noted in the introduction, there exist examples of graphs containing a transparent H-minor but no subdivision of H.

But while transparent minors are more general than subdivisions, they are defined so as to share a fundamental property with them: if H is a transparent minor of G, then one can convert a routing of terminal pairs through H into a routing through G.

Lemma 4.1 *Let $\{(\alpha_1, \beta_1), \ldots, (\alpha_k, \beta_k)\}$ be a realizable set of terminal pairs in a graph H, with all $2k$ terminals distinct, and let G be a graph that contains a transparent H-minor. In G, choose arbitrary nodes $s_i \in S(\alpha_i)$ and $t_i \in S(\beta_i)$ for $i = 1, 2, \ldots, k$. Then the set of terminal pairs $\{(s_1, t_1), \ldots, (s_k, t_k)\}$ is realizable in G.*

Weavings. In constructing a transparent grid minor, we will use sets of paths that intersect one another in particular ways. This is a standard general approach to constructing grid minors (see e.g. [1, 7, 8, 10, 32, 33]), though the details tend to differ depending on the application; here we must be careful to ensure the transparency of the minor.

We begin with a basic definition. Fix a drawing of a planar graph K, and consider two simple paths P and Q that meet at a node v. Let e, e' be the edges of P incident to v, and let f, f' be the edges of Q incident to v. We say the meeting of P and Q is *oblique* if e and e' appear next to each other in the clockwise order of $\{e, e', f, f'\}$, and we say that meeting is *tranverse* otherwise.

We now define a particular configuration of crossing paths. Let K be a planar graph drawn in a disc Δ, and let $\mathcal{W}$ be a set of paths in K. We say $\mathcal{W}$ is a *weaving* in K if the following properties hold.

(W1) $\mathcal{W}$ consists of paths $P_1, P_2, \ldots, P_a, Q_1, Q_2, \ldots, Q_b$ such that the path P_i has endpoints p_i, p_i', the path Q_j has endpoints q_j, q_j', and these endpoints are all distinct and drawn on the boundary of Δ in clockwise order $q_1, q_2, \ldots, q_b, p_1', p_2' \ldots, p_a', q_b', q_{b-1}', \ldots, q_1, p_a, p_{a-1}, \ldots, p_1$.

(W2) All paths in $\mathcal{W}$ are mutually edge-disjoint.

(W3) For each $i < j$, there is no tranverse crossing between P_i and P_j, or between Q_i and Q_j.

Further, we say that $\mathcal{W}$ is a *simple weaving* in K if all paths in $\mathcal{W}$ are simple (no path revisits the same node), and there is at most one transverse crossing between any pair of paths. We refer to the paths $P_1, \ldots, P_a$ as the *rows* of the weaving and the paths $Q_1, \ldots, Q_b$ as the *columns*; and we refer to the set of pairs of endpoints of all paths as the *anchors* of the weaving. The *order* of a weaving is $\min(a, b)$.

In the next two sections, we show how to construct these objects in the graph associated with the input instance.

5 Constructing a Weaving

We are given an instance $(G, \mathcal{T})$ of the disjoint paths problem subject to the assumptions specified at the end of Section 3; in this section, we show that G contains a subgraph with a weaving of order $\Omega(k)$. In the next section, we will claim that one can construct, from this, a simple weaving in G of order $\Omega(k)$, and then from this a transparent grid minor in G of order $\Omega(k)$.

Some Sufficient Conditions for Realizability. Let K be a planar graph (without self-loops, but potentially with parallel edges), drawn in the plane $\mathbf{R}^2$. By an *arc*, we mean a homeomorphic image of the interval $[0, 1]$, and by a *loop*, we mean a homeomorphic image of the unit circle. A loop or arc in $\mathbf{R}^2$ is K-*normal* if it meets the drawing of K only at nodes; for such a loop or arc $I \subseteq \mathbf{R}^2$, we use $I \cap K$ to denote the set of nodes of K that are met by I. The *length* of a K-normal arc or loop I is defined to be $|I \cap K|$. If $L \subseteq \mathbf{R}^2$ is a loop, then its *interior* $\mathrm{Int}(L)$ is the closure of the bounded component of $\mathbf{R}^2 \setminus L$, and its *exterior* $\mathrm{Ext}(L)$ is the closure of the unbounded component of $\mathbf{R}^2 \setminus L$. An arc I is a *meridian* of L with respect to $\mathrm{Int}(L)$ (resp. $\mathrm{Ext}(L)$) if both ends of I belong to L, and the interior of I lies in the interior of $\mathrm{Int}(L)$ (resp. $\mathrm{Ext}(L)$). The two arcs obtained from L by deleting the endpoints of a meridian I will be called the *alternate arcs* of L with respect to I.

There are a number of theorems providing sufficient conditions for the solvability of the disjoint paths problem in planar graphs; here we use a theorem of Okamura [26], generalizing an earlier theorem of Okamura and Seymour [27]. Let $(K, \mathcal{X})$ be an instance of the disjoint paths problem, and let $K + \mathcal{X}$ denote the graph in which the pairs corresponding to $\mathcal{X}$ are added as edges to K. For a set $S \subseteq V(K)$, let $\gamma(S)$ denote the set of terminal pairs with exactly one end in S. Okamura's Theorem is the following.

Theorem 5.1 ([26]) *Let $(K, \mathcal{X})$ be an instance of the disjoint paths problem in which*

(i) K is planar;

(ii) there are faces Γ and Γ' such that each terminal pair in $\mathcal{X}$ has both members incident to one of Γ or Γ';

(iii) the graph $K + \mathcal{X}$ is Eulerian; and

(iv) the cut condition *holds: for every set $S \subseteq V(K)$, we have $|\gamma(S)| \leq |\delta(S)|$.*

Then the instance is realizable, and edge-disjoint paths joining all terminal pairs can be found in polynomial time.

(The Okamura-Seymour theorem is the variant of this theorem in which (ii) is weakened to require all terminals to be incident to a single face [27].)

To build a weaving, we will want to be able to construct edge-disjoint paths in a planar graph with all terminal pairs on the outer face (these will form the anchors of the weaving), and with nodes of odd degree potentially incident to both the outer face and an exceptional face Γ^*. This is close to the setting of Okamura's Theorem, except that $K + \mathcal{X}$ may not be Eulerian.

We deal with this using an analogue of an idea applied by Frank to the Okamura-Seymour theorem [11, 12]. We define each odd-degree node to be a terminal, and we define new terminal pairs by matching them up consecutively around each face. We now have the desired Eulerian condition, but the new terminal pairs may have caused the cut condition to be violated. The cut condition will survive this construction, however, if we assume a stronger, *extended cut condition* on the initial instance; then the extra terminals cannot push $|\gamma(S)|$ up high enough to exceed $|\delta(S)|$ for any set S. Here is the precise formulation of the theorem.

Theorem 5.2 *Let $(K, \mathcal{X})$ be an instance of the disjoint paths problem in which* *(i′) K is planar.*

(ii′) All terminals lie on the outer face Γ_0 of K;

(iii′) All odd-degree nodes of K lie on Γ_0 and at most one other face Γ^*.

(iii″) There is at most one node incident to both Γ_0 and Γ^*; and if there are no nodes incident to both then there are an even number of odd nodes incident to each.

(iv′) The extended cut condition *holds: for every set $S \subseteq V(K)$, such that both $K[S]$ and $K \setminus S$ are connected, we have $|\gamma(S)| < |\delta(S)|$; moreover, if S and $V(K) \setminus S$ each contain nodes incident to both Γ_0 and Γ^*, we have $|\gamma(S)| < |\delta(S)| + 2$.*

Then the instance is realizable, and edge-disjoint paths joining all terminal pairs can be found in polynomial time.

Constructing a Weaving. We now return to our instance $(G, \mathcal{T})$. To construct a weaving, we will find a G-normal loop L^* of length $\Omega(k)$ bounding a subgraph G^* in which the extended cut condition holds for a set of terminal pairs defined on the outer face of G^* to form the anchors of a weaving. We use Theorem 5.2 to construct edge-disjoint paths; by planarity, these paths must mutually cross, yielding the desired weaving.

Thus, the crux of the following theorem is the existence of an appropriate G^* and L^*. We sketch the proof here.

Theorem 5.3 *G contains a subgraph with a weaving of order $\Omega(k)$, and this weaving can be constructed in polynomial time.*

Proof Sketch. We adapt a greedy procedure due to Robertson, Seymour, and Thomas [35], a variant of which was also employed by Chekuri et al. [7, 8]; it was used in these papers to construct grid minors in planar graphs under assumptions that were similar, but without the even-degree condition and without the goal of ensuring edge-disjointness among the paths. The construction is also related to an approach used in our work with Tardos [19, 20].

We start with a G-normal loop L that completely encloses the drawing of G. We now shrink this loop by a sequence of local operations. Whenever the length of L is less than ck for a small constant c, we push it inward so it passes through an additional node. We can also slide it over an edge between consecutive nodes on L. Finally – the crucial operation – if L currently bounds a subgraph G', we look for a "short-cut" through G'. A short-cut here is G'-normal meridian I with respect to $\mathrm{Int}(L)$ such that both alternate arcs L_1 and L_2 are at least as long as I. In this case, we replace L with $L_i \cup I$ for the $i \in \{1, 2\}$ such that $\mathrm{Int}(L_i \cup I)$ contains more nodes of X.

Since the terminal set in G is well-linked, and the length of L is always much less than k, this process must stop while there are still $\Omega(k)$ terminals in $\mathrm{Int}(L)$. Let L^* be the loop at termination, G^* the subgraph drawn inside it, and Z the $\Omega(k)$ nodes at which L^* meets G^*. We define an instance of the disjoint paths problem in G^* by pairing members of Z in the order formed by the anchors in the definition of a weaving.

Because we iteratively updated L until no short-cuts could be found, the extended cut condition almost holds. The one difficulty is that we need $|\gamma(S)| < |\delta(S)| + 2$ when S and $V(G^*) \setminus S$ each contain nodes on both the outer face of G^* and also Γ^*. We thus define the iterative procedure to produce L^* more broadly: we allow short-cuts that pass through Γ^* to be up to two nodes longer than the alternate arcs. A long short-cut such as this can be produced only once in the whole process, however, since after constructing it we have all odd-degree nodes on the outer face of G^*. Hence the length of L^* can only grow by two nodes above the bound ck over the course of all iterations, so it remains sufficiently short for the above arguments to hold.

Thus, we are in a position to apply Theorem 5.2; one then verifies that the resulting set of paths forms the weaving. ■

In addition to having a weaving of order $\Omega(k)$, we would like one for which a large subset of terminals can be routed to the anchors using disjoint paths. We accomplish this using the folllowing argument of Chekuri et al. [8] (which they attribute to Paul Seymour).

For two sets of nodes A and B in a graph, with $|A| \leq |B|$, we say that A is *attachable* to B if there exist $|A|$ edge-disjoint paths, each with one end in A and the other in B. If the weaving constructed in Theorem 5.3 does not have the property that the anchors are attachable to the terminals, then by well-linkedness, there is a short loop L' inside G^* (derived from a small cut) that contains most of the terminals inside it. We can resume the iterations from the proof of Theorem 5.3 from L' and again run them to termination. Repeating this as often as needed, we eventually end up with a loop L^* that establishes the following.

Theorem 5.4 *G contains a subgraph with a weaving of order $\Omega(k)$ for which the anchors are attachable to the set of terminals, and this weaving can be constructed in polynomial time.*

6 Constructing a Transparent Grid Minor

We now convert the weaving into a transparent grid minor of comparable size. While it would be possible to describe the routing algorithm directly in terms of the weaving, in fact weavings are conceptually messier than grids, and it is useful to abstract this complexity inside the statements of the following two theorems. Moreover, having a grid minor allows one to directly adapt routing algorithms that assume the existence of a grid, rather than having to modify these algorithms.

We first use an iterative procedure that "pulls out" extra transverse crossings in the weaving, in a manner analogous to the proof of Theorem 4.1 of Graph Minors III [32]. (The details differ since our goals here are in a sense the opposite of [32]: there the construction sought crossing paths that shared edges, whereas here we seek to keep the paths edge-disjoint.) We argue this procedure terminates, establishing

Theorem 6.1 *G contains a subgraph with a simple weaving of order $\Omega(k)$, for which the anchors are attachable to the set of terminals, and this simple weaving can be constructed in polynomial time.*

We then take every other row and every other column in the simple weaving, and show their crossings can be used to define super-nodes in a transparent grid minor. (It would not work to use every row and column, as adjacent rows and columns may be too extensively intertwined.) Thus,

Theorem 6.2 *G contains a subgraph with a transparent grid minor of order $\Omega(k)$, and this minor can be constructed in polynomial time. Moreover, it is possible to select a node from each boundary super-node so that the resulting set A is attachable to the set of terminals.*

7 The Full Routing Algorithm

Finally, we summarize the full routing algorithm, using the components developed in earlier sections. First, we use the method of Chekuri et al. [8] to produce a well-linked decomposition, and then we apply Section 3 to partition these instances into $O(\log n)$ classes. For each class, we will produce a routing for all the instances in the class, using paths that share no edges across instances; we then choose the class in which the most terminal pairs are routed.

Thus, suppose we are working with a single instance $(G'_i, \mathcal{T}_i)$ having k_i terminal pairs. First, using Sections 5 and 6, we construct a subgraph H of G'_i that is a transparent grid minor of order $\Omega(k_i)$, such that a set A consisting of one node from each boundary super-node of H is attachable to the set X of terminals.

From here, the algorithm now closely follows the CKS algorithm, with our transparent grid minor playing the role of their arbitrary grid minor. The goal of the remainder of the algorithm is to route $\Omega(k_i)$ terminal pairs via edge-disjoint paths in G'_i. The fact that the terminal set X is attachable to the boundary of the grid minor, together with the well-linkedness of X, implies that we can in fact find members of a set $\mathcal{T}'_i$ of $\Omega(k_i)$ terminal pairs that can be routed by edge-disjoint paths to distinct nodes on the boundary of H. We then use a technique from [7] to avoid having the paths used for routing terminals to the boundary of the grid minor overlap with paths used inside the grid for linking them up: we find a minor H' corresponding to a constant fraction of the rows and columns of H, such that a constant fraction of the pairs in $\mathcal{T}'_i$ have both their members outside the drawing of H'. We route these terminals to distinct nodes on the boundary of H' without using any of its internal edges.

Thus, to conclude the routing, we are left with a subproblem in which the underlying graph is the transparent grid minor H', and the terminal pairs $\mathcal{T}''_i$ consist of the paired points on the boundary of H' where the terminals in $\mathcal{T}'_i$ were routed. By viewing H' as an actual grid (rather than a minor), we can construct edge-disjoint paths through H' connecting a constant fraction of the terminal pairs in $\mathcal{T}''_i$. Finally, since H' is transparent, we apply Lemma 4.1 to convert these grid paths into edge-disjoint paths in G'_i connecting the same pairs of terminals. We thus route $\Omega(k_i)$ terminal pairs using edge-disjoint paths in G'_i.

Acknowledgements. We thank Chandra Chekuri, Sanjeev Khanna, and Bruce Shepherd for valuable comments, and for pointing out an error in an earlier version of this work.

References

[1] N. Alon, P. Seymour, R. Thomas. "Planar separators," SIAM J. Discrete Math. 7 (1994) 184-193

[2] M. Andrews, L. Zhang. "Hardness of the undirected edge-disjoint paths problem." *Proc. 37th ACM Symp. on Theory of Computing* 2005.

[3] Y. Aumann, Y. Rabani, "Improved bounds for all-optical routing," *Proc. 6th ACM-SIAM Symp. on Discrete Algorithms*, 1995, pp. 567–576.

[4] B. Awerbuch, R. Gawlick, F.T. Leighton, Y. Rabani, "On-line admission control and circuit routing for high performance computing and communication," *Proc. 35th IEEE Symp. on Foundations of Computer Science*, 1994, pp. 412–423.

[5] A. Broder, A. Frieze, E. Upfal, "Existence and construction of edge-disjoint paths on expander graphs," *SIAM J. Computing*, 23(1994), pp. 976–989.

[6] C. Chekuri, S. Khanna, B. Shepherd, "The All-or-Nothing Multicommodity Flow Problem," *Proc. 36th ACM Symp. on Theory of Computing*, 2004.

[7] C. Chekuri, S. Khanna, B. Shepherd, "Edge-disjoint paths in planar graphs," *Proc. 45th IEEE Symp. on Foundations of Computer Science*, 2004.

[8] C. Chekuri, S. Khanna, B. Shepherd, "Multicommodity flow, well-linked terminals, and routing problems," *Proc. 37th ACM Symp. on Theory of Computing*, 2005.

[9] J. Chuzhoy, S. Khanna. "Improved hardness results and integrality gaps for edge-disjoint paths," manuscript, 2005, cited in [8].

[10] R. Diestel, K.Yu. Gorbunov, T.R. Jensen and C. Thomassen. Highly connected sets and the excluded grid theorem, *J. Combin. Theory Ser. B* 75 (1999).

[11] A. Frank, "Edge-disjoint paths in planar graphs," *J. Comb. Theory Ser. B*, 39(1985), pp. 164–178.

[12] A. Frank, "Packing paths, cuts, and circuits — a survey," in *Paths, Flows, and VLSI-Layout*, B. Korte, L. Lovász, H.J. Prömel, A. Schrijver, Eds., Berlin: Springer-Verlag, 1990, pp. 49–100.

[13] N. Garg, V. Vazirani, M. Yannakakis, "Primal-dual approximation algorithms for integral flow and multicut in trees, with applications to matching and set cover," *Proc. International Colloq. on Automata, Languages, and Programming*, 1993, pp. 64–75.

[14] V. Guruswami, S. Khanna, R. Rajaraman, F.B. Shepherd, M. Yannakakis. Near-Optimal Hardness Results and Approximation Algorithms for Edge-Disjoint Paths and Related Problems. *J. Computer and System Sci.*, 67(3):473-496, 2003.

[15] R.M. Karp, "Reducibility among combinatorial problems," *Complexity of Computer Computations*, R.E. Miller, J.W. Thatcher, Eds., Plenum Press, 1972.

[16] R.M. Karp, F.T. Leighton, R. Rivest, C. Thompson, U. Vazirani, V. Vazirani, "Global wire routing in two–dimensional arrays," *Algorithmica*, 2(1987).

[17] J. Kleinberg, "Decision Algorithms for Unsplittable Flow and the Half-Disjoint Paths Problem," *Proc. 30th ACM Symposium on Theory of Computing*, 1998.

[18] J. Kleinberg, R. Rubinfeld, "Short paths in expander graphs," *Proc. 37th IEEE Symp. on Foundations of Computer Science*, 1996.

[19] J. Kleinberg, É. Tardos, "Approximations for the Disjoint Paths Problem in High–Diameter Planar Networks," *Proc. 27th ACM Symp. on Theory of Computing*, 1995, pp. 26–35.

[20] J. Kleinberg, É. Tardos, "Disjoint Paths in Densely Embedded Graphs," *Proc. 36th IEEE Symp. on Foundations of Computer Science*, 1995, pp. 52–61.

[21] P. Kolman, C. Scheideler. Simple On-Line Algorithms for the Maximum Disjoint Paths Problem. *Proc. 13th ACM Symp. Parallel Alg. and Arch.*, 2001.

[22] F.T. Leighton, S. Rao, "Circuit switching: a multicommodity flow based approach," *Proc. 9th International Parallel Processing Symposium*, 1995.

[23] T. Lengauer. *Combinatorial Algorithms for Integrated Circuit Layout.* Wiley, 1990.

[24] N. Linial, A. Magen, and M. Saks. Low distortion Euclidean embedding of trees. *Israel J. Math*, 106(1998).

[25] J. Matousek. On embedding trees into uniformly convex Banach spaces. *Israel Journal of Mathematics*, 114:221-237, 1999.

[26] H. Okamura, "Multicommodity flows in graphs," *Discrete Applied Mathematics* 6(1983), pp. 55-62.

[27] H. Okamura, P. Seymour, "Multicommodity flows in planar graphs," *J. Comb. Theory Ser. B*, 31(1981).

[28] S. Plotkin, "Competitive Routing in ATM networks," *IEEE J. Selected Areas in Communications*, 1995.

[29] W.R. Pulleyblank, "Two Steiner tree packing problems," invited talk at *27th ACM Symposium on Theory of Computing*, 1995.

[30] P. Raghavan, "Probabilistic construction of deterministic algorithms: approximating packing integer programs," *J. Computer and System Sciences*, 37(1988).

[31] P. Raghavan, C.D. Thompson, "Randomized rounding: a technique for provably good algorithms and algorithmic proofs," *Combinatorica*, 7(1987).

[32] Neil Robertson and P.D. Seymour; Graph minors. III. Planar tree-width. *J. Combin. Theory Ser. B* 36(1984)

[33] Neil Robertson and P.D. Seymour; Graph minors. V. Excluding a planar graph. *J. Combin. Theory Ser. B* 41(1986)

[34] N. Robertson, P.D. Seymour, "An outline of a disjoint paths algorithm," in *Paths, Flows, and VLSI-Layout*, B. Korte, L. Lovász, H.J. Prömel, A. Schrijver, Eds., Berlin: Springer-Verlag, 1990, pp. 267–292.

[35] Neil Robertson, Paul Seymour, and Robin Thomas; Quickly excluding a planar graph. *J. Combin. Theory Ser. B* 62 (1994) 323-348