SCHOOL OF OPERATIONS RESEARCH
AND INDUSTRIAL ENGINEERING
COLLEGE OF ENGINEERING
CORNELL UNIVERSITY
ITHACA, NY 14853-7501

TECHNICAL REPORT NO. 1127

May 1995

Disjoint Paths in Densely Embedded Graphs

By

Jon Kleinberg[1] and Éva Tardos[2]

---

[1]E-mail: kleinber@theory.lcs.mit.edu. Laboratory for Computer Science, MIT, Cambridge MA 02139 USA.
Supported by an ONR Graduate Fellowship.

[2]E-mail: eva@cs.cornell.edu. School of ORIE, Cornell University, Ithaca NY 14853.
Supported in part by a Packard Fellowship and an NSF PYI award.

Abstract

We consider the following *maximum disjoint paths problem* (MDPP). We are given a large network, and pairs of nodes that wish to communicate over paths through the network — the goal is to simultaneously connect as many of these pairs as possible in such a way that no two communication paths share an edge in the network. This classical problem has been brought into focus recently in papers discussing applications to routing in high-speed networks, where the current lack of understanding of the MDPP is an obstacle to the design of practical heuristics.

We consider the class of *densely embedded, nearly-Eulerian graphs*, which includes the two-dimensional mesh and many other planar and locally planar interconnection networks. We obtain a constant-factor approximation algorithm for the maximum disjoint paths problem for this class of graphs; this improves on an $O(\log n)$-approximation for the special case of the two-dimensional mesh due to Aumann–Rabani and the authors. For networks that are not explicitly required to be "high-capacity," this is the first constant-factor approximation for the MDPP in any class of graphs other than trees.

We also consider the MDPP in the on-line setting, relevant to applications in which connection requests arrive over time and must be processed immediately. Here we obtain an asymptotically optimal $O(\log n)$-competitive on-line algorithm for the same class of graphs; this improves on an $O(\log n \log\log n)$-competitive algorithm for the special case of the mesh due to Awerbuch, Gawlick, Leighton, and Rabani.

1 Introduction

We consider the following *maximum disjoint paths problem* (MDPP). We are given a large network, and pairs of nodes that wish to communicate over paths through the network — the goal is to simultaneously connect as many of these pairs as possible in such a way that no two communication paths share an edge in the network. This problem is well-known to be computationally difficult. Deciding whether all pairs can be so connected is one of Karp's original NP-complete problems [11]; it remains NP-complete even when the underlying graph is the two-dimensional mesh [13].

Our interest in this problem comes from two main sources. First, establishing disjoint paths is fundamental to routing in high-speed networks (see for example the applications mentioned in [5, 7, 19], as well as applications to optical routing in [1, 2, 22]). Although the types of routing problems that arise in such settings tend to have additional side constraints (e.g. connections have limited duration and can bring varying amounts of "profit"), the formulation described in the first paragraph contains the essence of virtually all such real-life routing problems in which each connection consumes a large fraction of the bandwidth on a link. As such, the current lack of understanding of the disjoint paths problem is a major obstacle to the design of practical heuristics. Indeed, [5] notes that in practice, the greedy algorithm tends to be used for routing, despite its bad performance on a number of very common interconnection patterns. Moreover, robust ways are known for converting algorithms for the MDPP into algorithms that can handle connections of limited duration or variable value [4]; thus, the difficulties contained in these more elaborate routing problems seem to stem mainly from the intractability of the MDPP.

This problem is also of basic interest in algorithmic graph theory. A lot of work has been done on identifying special cases of the disjoint paths problem that can be solved in polynomial time, or for which simple min-max conditions can be stated; see the survey by Frank [9]. Much less work has been done, however, on approximation algorithms for the MDPP; we are interested in extending the classes of graphs for which good approximations can be obtained.

1.1 Our Results

To be precise, let $G = (V, E)$ be a graph on n vertices and $\mathcal{T} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ a collection of *terminal pairs* — pairs of vertices of G. We say that $\mathcal{T}$ is *realizable* in G if there exist mutually edge-disjoint s_i-t_i paths, for $i = 1, \ldots, k$. The problem is then to find a realizable subset of $\mathcal{T}$ of maximum cardinality.

Our first main result is a constant-factor approximation for the maximum disjoint paths problem in the class of *densely embedded, nearly-Eulerian graphs* (defined below), which includes many common planar and locally planar interconnection networks. This improves on an $O(\log n)$-approximation for the case of the two-dimensional mesh due to Aumann and Rabani [2] and an $O(\log n)$-approximation for a class of planar graphs including the mesh due to the authors [12]. Our present algorithm makes use of variants of a number of the techniques developed in our earlier paper [12].

The assumption that we know all the terminal pairs in advance is not reasonable in situations in which connection requests between pairs of nodes arrive over time and must be processed immediately. In such a setting, it makes sense to consider *on-line* routing algorithms. Such an algorithm is given the graph G, terminal pairs arrive in an arbitrary order, and for each such pair it must irrevocably reject it, or assign it a path in G. As is standard, we refer to the approximation ratio achieved by an on-line algorithm as its *competitive ratio*; such an algorithm is said to be c-competitive if its competitive ratio is at most c.

Our second main result, then, is an $O(\log n)$-competitive randomized on-line algorithm for the MDPP in densely embedded, nearly-Eulerian graphs. This improves on an $O(\log n \log\log n)$-competitive algorithm for the special case of the two-dimensional mesh due to Awerbuch, Gawlick, Leighton, and Rabani [5]; moreover, [5] proves that no randomized on-line algorithm for the two-dimensional mesh can be better than $\Omega(\log n)$-competitive, implying that our algorithm is asymptotically optimal.

We feel that the size of the constants in our algorithms as presented here, while not astronomical, pushes them outside the range of immediate practical utility. It is important to note, however, that the previous best bounds — both off-line and on-line — for the two-dimensional mesh [2, 5, 12] involve similarly large constants inside the $O(\cdot)$ notation. Moreover, some of the ideas used by the algorithms here may well be of use in designing practical heuristics. In particular, they suggest convenient ways of handling networks that are mesh-like in structure, without placing many requirements on the "fine structure" around each node in the network.

1.2 Previous Work

Much of the previous work on this problem has dealt with the case in which each path consumes only a small fraction of the available bandwidth on an edge; this can be modeled by requiring $\Omega(\log n)$ parallel copies of each edge. In this case, the randomized rounding technique of Raghavan and Thompson [21, 20] can be used to obtain an off-line constant-factor approximation. Awerbuch, Azar, and Plotkin give an on-line $O(\log n)$-competitive algorithm for this case [3], which they show is asymptotically tight for deterministic on-line algorithms.

As noted in [5] however, there are many applications in which each communication path consumes a large fraction of the available bandwidth on a link; thus it makes sense to consider approximation algorithms for graphs without a large number of parallel edges. The results here are much more restricted. For trees with parallel edges, Garg, Vazirani, and Yannakakis [10] obtain an off-line 2-approximation (the maximization problem is NP-complete, though deciding realizability is easy); Awerbuch et. al. [5] give an $O(\log d)$-competitive randomized on-line algorithm for trees of diameter d, extending an earlier result of Awerbuch et. al. [4]. Essentially the only approximation results known for graphs other than trees are those mentioned earlier for the mesh and related planar graphs [2, 5, 12]. Thus our result here is the first constant-factor approximation for any class of graphs other than trees, when one does not require $\Omega(\log n)$ parallel copies of each edge.

A different approach is taken in papers of Peleg and Upfal [19] and Broder, Frieze, and Upfal [7] (see also Broder et. al. [6]). Here the underlying graph G is assumed to have strong expansion properties; in this case one can prove that any set of terminal pairs of at most a given size must be realizable in G, and that corresponding paths can be found in (randomized) polynomial time. The results in [7] are strong enough that they implicitly provide a polylogarithmic approximation for the MDPP in sufficiently strong expanders of bounded degree.

Cases in which the MDPP can be solved in polynomial time are surveyed in [9]; here we only discuss two specific results that we will use. First, suppose G is planar, the terminals $\mathcal{T}$ lie on a single face of G, and the pair $(G, \mathcal{T})$ satisfies the following *parity condition*: the *augmented graph* formed by adding to G the edges corresponding to $\mathcal{T}$ must be Eulerian. In this case, a theorem of Okamura and Seymour [18] says that the realizability of $\mathcal{T}$ in G can be decided in polynomial time; and in fact the following *cut condition* is sufficient for realizability: one cannot remove j edges from G and separate more than j terminal pairs. A linear-time algorithm for this problem has recently been obtained by Wagner and Weihe [29]. We will use an extension of the Okamura–Seymour, due to Frank [8], which concerns the case in which the parity condition does need not to hold on the face containing the terminals.

We also use a theorem of Schrijver [27] that provides an algorithm for finding vertex-disjoint paths in a graph embedded on a compact surface Σ, such that the paths satisfy given homotopy constraints.

1.3 Extensions of Our Results

In the on-line algorithm we assume that (i) all connections have infinite duration, and (ii) all connections have the same "value" (i.e. our objective function could have been a weighted sum of the set of pairs we accept, rather than an unweighted sum). There are general transformation techniques due to Awerbuch et. al. [4] that allow us to convert our results to on-line algorithms that can handle connections of limited duration and variable value, at the cost of additional logarithmic terms in the competitive ratio. (We pay $O(\log T)$ and $O(\log P)$, where T and P are the ratios between the largest and smallest durations and values respectively.)

Also, for any fixed value of d, it is not difficult to use the techniques developed here to obtain a constant-factor off-line approximation and an $O(\log n)$-competitive on-line algorithm for the MDPP on the d-dimensional mesh.

2 Sketch of the Algorithms

The on-line and off-line algorithms have a number of similarities; to give the reader a sense of their structure, we give a sketch of them here for the special case of the two-dimensional mesh. (Note that we are still obtaining improved bounds in this case.) We begin with the on-line algorithm, which is somewhat easier.

So suppose that G is the two-dimensional mesh. We partition G in the natural way into disjoint *subsquares* of dimensions $\log n \times \log n$ each. The algorithm makes an initial random decision whether to route *short connections* (of length less than $c \log n$, for a constant c that is, say, greater than 12) or *long connections* (of length at least $c \log n$). The case of short connections is not very difficult.

We handle long connections as follows. For a given $\log n \times \log n$ subsquare S, we call the block of nine subsquares centered at S its *enclosure*. We first choose a maximal collection $\mathcal{C}$ of subsquares subject to the condition that their enclosures be disjoint. We call a subsquare in $\mathcal{C}$

a *cluster*. We now only route connections both of whose ends lie in such clusters; if we choose $\mathcal{C}$ using a randomized algorithm such as [15], we can ensure that we are, in expectation, within a constant factor of optimal after doing this.

Let C_i denote one cluster in $\mathcal{C}$, and D_i its enclosure. For each subsquare of G that does not belong to the enclosure of some C_i, we add it to its closest enclosure D_i (ties broken according to some fixed rule); now the union of the enclosures is all of G. We now build a *simulated graph* $\mathcal{N}$ on the set of enclosures, by contracting each enclosure into a single node; i.e. we join two if they touch. In this simulated graph, all capacities are at least $\log n$, since if two enclosures touch then there are at least $\log n$ edges in their common boundary. This is very useful, since the on-line algorithm of Awerbuch, Azar, and Plotkin [3] (the "AAP algorithm"), which requires large capacity edges, can now be applied to $\mathcal{N}$.

When given a request (s_i, t_i), our algorithm accepts it if (i) no connections have yet left the clusters containing s_i and t_i, and (ii) the AAP algorithm running on the graph $\mathcal{N}$ accepts the connection (s_i, t_i). The crucial point is that we argue that our algorithm can route any connection it accepts as follows. The "global route" in $\mathcal{N}$ specified by the AAP algorithm consists of a sequence of neighboring enclosures; from this, we would like to produce a path in G using the natural crossbar structures in each enclosure. To guarantee that this routing in G is feasible, we have to run the AAP algorithm on a "scaled-down" version of $\mathcal{N}$. We divide each capacity down by an appropriate constant; the resulting network has at most $\varepsilon \log n$ capacity entering each node for a small constant ε, and therefore the enclosures are large enough to handle all such accepted connections.

To argue that the resulting algorithm is $O(\log n)$-competitive we view it as being obtained from the "cooperation" of two maximization algorithms, A_1 and A_2. A_1 is specified by rule (i); it only allows one terminal to leave each cluster. A_2 is the AAP algorithm running on the scaled down version of $\mathcal{N}$. We claim each of these algorithms is $O(\log n)$-competitive. A_1 comes with this guarantee since no more than $O(\log n)$ connections can leave a single cluster. To show A_2 is $O(\log n)$-competitive, we extend the AAP analysis to networks where edges have only $\varepsilon \log n$ capacity for a fixed $\varepsilon > 0$, and show that the algorithm is $O(\log n)$-competitve against a fractional off-line solution. Thus A_2 is $O(\log n)$-competitive against the optimum in $\mathcal{N}$ (not scaled down), which by construction is an upper bound on the optimum in G. Now, why is the combined algorithm $O(\log n)$-competitive? We charge each rejected connection to the algorithm that rejected it. One of the algorithms A_i is charged for at least half the rejections; but then the number of accepted connections must be at least $1/O(\log n)$ times this amount, or it would contradict the performance guarantee of A_i. (Note that the $O(\log n)$ bound depends crucially on the fact that each of A_1 and A_2 bases its decision at each step on what has been *jointly* accepted so far.)

In the off-line algorithm (see Section 6), we must be much more aggressive than algorithm A_1 at getting connections out of individual clusters. In particular, we extend our simulated network so that it captures the notion of *local routing* out of a cluster, as well as the notion of *global routing* between clusters. Thus, we define the following simulated network $\mathcal{N}'$ which contains $\mathcal{N}$ as a subgraph. Recall that $\mathcal{N}$ had one node z_i for each cluster $C_i \in \mathcal{C}$. We form $\mathcal{N}'$ by simply attaching C_i to z_i using an edge from z_i to *each* boundary node of C_i. The edges between nodes of $\mathcal{N}$ will still have capacity $\Theta(\log n)$; but other edges (with at least one end in the clusters) will have unit capacity.

An algorithm of Raghavan based on *randomized rounding* [21, 20] is sufficient to obtain a constant-factor approximation to the MDPP when all capacities are $\Omega(\log n)$. In our case, this is fine for the high-capacity subgraph $\mathcal{N}$, but problematic within the clusters, where edges have unit capacity. Nevertheless, we show in Section 6 that by running Raghavan's algorithm on $\mathcal{N}'$, we can find alternative edge-disjoint routes out of the clusters for a constant fraction of the terminal pairs it selects. This is accomplished by analyzing the *escape problem* induced in each (square) cluster by the terminals that want to get out to the high-capacity subgraph $\mathcal{N}$. Since an escape problem in a mesh is completely characterized by the *rectangular cuts*, of which there are only polynomially many, we can show by a simple summation that with high probability a constant fraction of the selected terminals can in fact escape to the boundary.

Finally, we can use the crossbar structures in the enclosures to connect the ends of these local "escape paths" with the paths in $\mathcal{N}$ returned by the randomized rounding algorithm; in this way we obtain a feasible routing for a set of terminal pairs whose size is within a constant factor of optimal.

3 Densely Embedded Graphs

We begin with a definition. If H is a graph and $X \subset V(H)$, let $H[X]$ denote the subgraph induced by the vertices of X, and $\delta(X)$ denote the set of edges with one end in X and the other in $V(H) - X$.

Definition 3.1 *A graph H is an α-semi-expander if for every $X \subset V(H)$ for which $|X| \leq \frac{1}{2}|V(H)|$, we have $|\delta(X)| \geq \alpha\sqrt{|X|}$.*

We wish to define a class of graphs that generalizes the two-dimensional mesh; to this end, we point out the following properties of the mesh.

(i) It is a planar graph with bounded degree, and (aside from one "exceptional face") it is Eulerian and has bounded face size.

(ii) It is an α-semi-expander, for a constant $\alpha > 0$ based on the ratio of the two side lengths of the mesh.

(iii) Square sub-meshes of the mesh satisfy (i) and (ii).

In the arguments to follow, it is quite cumbersome — though not technically difficult — to deal with "exceptional faces" of the type in (i). Thus, for most of the paper we will work with the more restricted class of *uniformly densely embedded graphs*, where *all* faces have bounded size; and we will assume further assume that G is Eulerian. In Section 7, we show how to handle graphs with an exceptional face; in this way, our class of graphs will include the two-dimensional mesh.

First we need some preliminary topological definitions. Let Σ denote a compact orientable surface; it is well-known (see e.g. [16]) that Σ may be obtained from the 2-sphere by attaching a finite number of handles. We use the terms *disc* (homeomorph of $[0,1]\times[0,1]$), *arc* (homeomorph of $[0,1]$), *curve* (continuous image of $[0,1]$) and *closed curve* (continuous image of S^1). By a

Σ-*disc*, we mean a subset of Σ homeomorphic to a disc. Our definition of graph embedding is standard; a *face* of an embedded graph G is a connected component of $\Sigma - G$, and we say G is *strongly embedded* if the closure of each face is a Σ-disc, and each face is bounded by a simple cycle of G.

Our class of graphs is defined to satisfy analogues of properties (i), (ii), and (iii) locally. For $u, v \in V$, let $d(u,v)$ denote the least number of edges in a u-v path, and $B_r(v) = \{u : d(u,v) \leq r\}$. Then

Definition 3.2 *A graph $G = (V, E)$ is* uniformly densely embedded *with parameters α, λ, Δ, and ℓ if:*

(i) G is strongly embedded on a compact orientable surface Σ, it has maximum degree Δ, and each face is bounded by at most ℓ edges.

(ii) For each $r \leq \lambda \log n$ and each $v \in V$, the drawing of $G[B_r(v)]$ is contained in a Σ-disc.

(iii) For each $r \leq \lambda \log n$ and each $v \in V$, the graph $G[B_r(v)]$ is an α-semi-expander.

Thus, for the remainder of the paper aside from Section 7, we will assume that G is a simple Eulerian graph that is *uniformly densely embedded* on a surface Σ with parameters α, λ, Δ, and ℓ. In Section 7, we show how our algorithms can be adapted to handle graphs satisfying the following weaker definition; it is the same as the definition above, except that we allow an exceptional face.

Definition 3.3 *A graph $G = (V, E)$ is* densely embedded *and* nearly-Eulerian *with parameters α, λ, Δ, and ℓ if:*

(i) G is strongly embedded on a compact orientable surface Σ and has maximum degree Δ.

(i)′ G contains a face Φ^* such that all faces other than Φ^* are bounded by at most ℓ edges, and every vertex not incident to Φ^* has even degree.

(ii) For each $r \leq \lambda \log n$ and each $v \in V$, the drawing of $G[B_r(v)]$ is contained in a Σ-disc.

(iii) For each $r \leq \lambda \log n$ and each $v \in V$, the graph $G[B_r(v)]$ is an α-semi-expander.

The classes of graphs satisfying these definitions are incomparable to the class considered in our earlier paper [12]. The *semi-expansion* condition above will be shown to imply the *uniformly high-diameter* condition of [12] (see Lemma 3.4); however, in the current paper, we only require planarity and semi-expansion locally, and essentially no restrictions are placed here on the "global" structure of the graph. The examples of uniformly high-diameter graphs constructed in [12] are densely embedded as well; and in Section 3.2 we will discuss some related classes of graphs that are densely embedded.

3.1 Some Basic Properties

We now show that our definition implies G has some additional mesh-like properties. First of all, for any $v \in V$ and $r \leq \lambda \log n$, the fact that $G[B_r(v)]$ is a bounded-degree semi-expander implies that the set $B_r(v)$ has size at least quadratic in r; by also using the planarity of $G[B_r(v)]$, one can show an analogous upper bound. We summarize this as follows.

Lemma 3.4 *There are constants $\bar{\alpha}$ and β depending only on α and Δ such that the following holds. For each $r \leq \lambda \log n$ and each $v \in V$, we have $\bar{\alpha} r^2 \leq |B_r(v)| \leq \beta r^2$.*

Proof. Fix $r \leq \lambda \log n$ and $v \in V$, and let $S = B_r(v)$. To see the lower bound, note that for any $i \leq r$, if $x_i = |B_i(v)|$, then by the semi-expansion of H we have

$$x_i \geq x_{i-1} + \frac{\alpha}{\Delta - 1}\sqrt{x_{i-1}}.$$

For at least $\alpha\sqrt{x_{i-1}}$ edges leave $B_{i-1}(v)$, and at most $\Delta - 1$ are incident to any one vertex. Let $\nu = \frac{\alpha}{\Delta-1}$; now one verifies by induction that $x_i \geq \frac{1}{16}\nu^2 i^2$:

$$\begin{aligned} x_i &\geq \frac{1}{16}\nu^2(i-1)^2 + \frac{1}{4}\nu^2(i-1) \\ &= \frac{1}{16}\nu^2(i-1)(i+3) \\ &\geq \frac{1}{16}\nu^2 i^2. \end{aligned}$$

To see the upper bound, we observe that $G[S]$ is planar and has diameter at most $2r$. Let $n = |S|$. By the Lipton-Tarjan planar separator theorem [14], there is a set of at most $4r + 1$ vertices whose removal breaks H into components each of size at most $\frac{2}{3}n$. Let X be a union of these components of size between $\frac{1}{3}n$ and $\frac{1}{2}n$. Then

$$\begin{aligned} \frac{\alpha\sqrt{n}}{\sqrt{3}} &\leq \alpha\sqrt{|X|} \\ &\leq |\delta(X)| \\ &\leq (\Delta - 1)(4r + 1) \end{aligned}$$

from which the result follows. ∎

We introduce some additional notation. If $S \subset V$, let $\pi(S)$ denote the set of vertices of S incident to an edge in $\delta(S)$, and $S^o = S - \pi(S)$. Observe that removing $\pi(S)$ from S disconnects it from the rest of the graph, and $\pi(B_r(v))$ consists of vertices at exactly distance r from v. If C is a connected subset of $G - S$, we use $\Gamma(S, C)$ to denote the (unique) connected component of $G - S$ containing C. The set of vertices in $\pi(S)$ which have a neighbor in $\Gamma(S, C)$ will be called the *segment of* $\pi(S)$ *bordering* C and denoted $\sigma(S, C)$. Finally, we say that a set $S \subset V$ is *simple* if $G - S$ is connected.

The following two facts are quite useful; the first essentially relates the size of the "perimeter" of a set $B_r(v)$ ($r \leq \lambda \log n$) to its radius.

Lemma 3.5 *Let $c > 1$ and r a positive integer be such that $cr < \lambda \log n$. Then for some r' between r and cr, we have $|\pi(B_{r'}(v))| \leq \beta \cdot \frac{c^2}{c-1} \cdot r$.*

Proof. Since $\pi(B_{r'}(v)) \subset \{u : d(v, u) = r'\}$, the sets $\pi(B_r(v)), \pi(B_{r+1}(v)), \ldots, \pi(B_{cr}(v))$ are all disjoint and contained in $B_{cr}(v)$. Since $|B_{cr}(v)| \leq \beta c^2 r^2$, one of these sets has size at most $\beta \cdot \frac{c^2}{c-1} \cdot r$. ∎

Using this, we show that we can extend any small enough set U to a simple set with at most a constant-factor increase in its radius.

Lemma 3.6 *There is a constant ξ such that the following holds. Let $U \subset B_r(v)$, where $r \leq \frac{1}{\xi}\lambda \log n$. Then there is a component Γ of $G - U$ and a planar simple set U' such that $U \subset U' \subset B_{\xi r}(v)$, $G - U' = \Gamma$, and $\sigma(U, \Gamma) = \sigma(U', \Gamma)$.*

Proof. Choose r' between r and $2r$ so that $|\pi(B_{r'}(v))| \leq 4\beta r$. Let $U_0 = B_{r'}(v)$, and $G - U_0$ have components $\Gamma_1, \ldots, \Gamma_p$.

Now set $s = 8\bar{\alpha}^{-1/2}\alpha^{-1}\beta\Delta$ and $\xi = 2s + 2$. We claim that all but one of the components Γ_i are contained in $B_{\xi r}(v)$. For suppose not; then for $i \neq j$ there are $w \in \Gamma_i$ and $w' \in \Gamma_j$ such that w and w' are each at distance s from U_0, $B_s(w) \subset \Gamma_i$, and $B_s(w') \subset \Gamma_j$. Now consider the edge cut of size at most $4\beta\Delta r$ formed by $\delta(U_0)$; one of the two spheres $B_s(w)$ and $B_s(w')$, say the latter, is contained in a small component of this cut in $G[B_{\xi r}(v)]$. But then the semi-expansion of $G[B_{\xi r}(v)]$ requires that $4\beta\Delta r \geq \alpha\sqrt{|B_s(w')|}$, which is a contradiction since by Lemma 3.4 we have $|B_s(w')| \geq \bar{\alpha}s^2$.

So for some i, only Γ_i is not contained in $B_{\xi r}(v)$. Now let $\Gamma'_1, \ldots, \Gamma'_q$ denote the components of $G - U$; so Γ_i is contained in one of these, say Γ'_1, and $\Gamma'_2, \ldots, \Gamma'_q$ are all contained in $B_{\xi r}(v)$. Thus we have

$$U' = U \cup \bigcup_{j>1} \Gamma'_j \subseteq B_{\xi r}(v).$$

In particular, U' is planar since $\xi r \leq \lambda \log n$, and it is simple since $G - U'$ has only the component Γ'_1. Thus U' satisfies the conditions of the lemma. ∎

Finally, we show a general property of planar graphs H with small face size: if the distance between two nodes in H is large, than the value of any edge cut which contains both in the same segment of its boundary must also be relatively large.

Lemma 3.7 *Let H be a planar graph, with distinguished faces $\Phi_1, \ldots, \Phi_r$ bounded by cycles $Q_1, \ldots, Q_r$ respectively. Suppose that all faces other than $\Phi_1, \ldots, \Phi_r$ are bounded by at most ℓ edges, and for a constant d' and all $i \neq j$ we have $d(Q_i, Q_j) \geq d'$.*

Let $U \subset V(H)$ and $v, w \in \sigma(U, C)$ for some component C of $G - U$. Then

$$|\delta(U)| \geq \min(\ell^{-1}d', \ell^{-1}d(v, w)).$$

Proof. Let $S = \sigma(U, C) \subset U$. In the graph $H[U]$, S lies on a single facial cycle Q. Traversing Q in a clockwise direction starting at v, we encouter faces $R_1, \ldots, R_p$ whose boundaries contain vertices both of U and of $H - U$.

Suppose that among the $\{R_i\}$ there are two distinct large faces Φ_m and $\Phi_{m'}$. Choose such a pair for which $R_a = \Phi_m$, $R_b = \Phi_{m'}$, and $R_c \notin \{\Phi_i\}$ for $a < c < b$. Let P denote the corresponding maximal subpath of Q whose internal vertices are incident only to faces R_c, for $a < c < b$. Then among every ℓ consecutive vertices of P, there must be one incident to an edge in $\delta(U)$; since $|P| \geq d'$ by the hypotheses of the lemma, this implies the claimed bound.

Otherwise, there is a single large face Φ_m among the $\{R_i\}$; note that Φ_m may appear several times on the traversal of Q. Now there are two sub-paths of Q from v to w, which we denote P_0 and P_1. Since v, w border the same component of $H - U$, the face Φ_m does not appear in a traversal of one of P_0 or P_1 — suppose it is P_0. So as above, among every ℓ consecutive vertices of P_0, there must be one incident to an edge in $\delta(U)$; and we have $|P_0| \geq d(v, w)$. ■

3.2 Related Classes of Graphs

In this section, we show a natural construction which produces uniformly densely embedded graphs; it is related to the definition of *geometrically well-formed graphs* in our earlier paper [12]. The material in this section is independent of the rest of the paper.

We wish to define a notion of a surface being locally planar, in the following sense. Let Σ be a compact orientable surface, embedded in $\mathbf{R}^3$. For $x \in \Sigma$, let $B'_d(x)$ denote the set of all points of Σ whose distance from x (as measured on Σ) is at most d. We say a set $X \subset \Sigma$ is *flat* if there are positive constants γ_0, γ_1 and a Σ-disc D such that

(i) $X \subseteq D$.

(ii) If $B'_s(x) \subset X$, then the surface area of $B'_s(x)$ is at least $\gamma_0 s^2$ and at most $\gamma_1 s^2$.

(iii) If $D' \subseteq D$ is a Σ-disc whose boundary has length s, then the surface area of D' is at most $\gamma_1 s^2$.

Of course all these properties hold if Σ, for example, is the unit sphere in $\mathbf{R}^3$. We say that Σ is (r, γ_0, γ_1)-locally flat if it is orientable, and for all $x \in \Sigma$ and positive $s \leq r$, the set $B'_s(x)$ is flat.

Now we say that a graph is *locally well-formed* if it is drawn on a locally flat surface, and each face has geometrically about the same (small) size.

Definition 3.8 *A graph H drawn on Σ is* locally well-formed *with parameters $\Delta, \ell, \gamma_0, \gamma_1, \rho_0, \rho_1$ if it has maximum degree Δ and there is an $r > 0$ such that*

(i) Σ is $(r \log n, \gamma_0, \gamma_1)$-locally flat,

(ii) The maximum face size in the drawing of H is ℓ, and

(iii) for each face Φ of G there is an $x \in S$ so that $B'_{\rho_0 r}(x) \subset \Phi \subset B'_{\rho_1 r}(x)$.

We now want to show that every locally well-formed graph is uniformly densely embedded. To show this, the following routine lemma is useful: in Definition 3.1 it is enough to require semi-expansion for cuts that produce only two components.

Lemma 3.9 *H is an α-semi-expander if and only if the condition of Definition 3.1 holds for all sets X for which $H[X]$ and $H - X$ are both connected.*

Proof. Suppose H satisfies Definition 3.1 for all sets X with $H[X]$ connected. Then consider X with $H[X]$ is not connected, and let $\Gamma_1, \ldots, \Gamma_p$ be its components. Then

$$\sum_i |\delta(\Gamma_i)| \geq \alpha \sum_i \sqrt{|\Gamma_i|} \geq \alpha \sqrt{|\cup_i \Gamma_i|}.$$

Now suppose H satisfies Definition 3.1 for all sets X for which both $H[X]$ and $H-X$ are connected; we show it satisfies it for all sets X with possibly only $H[X]$ connected. Let X be a set of the latter kind, and $\Gamma_1, \ldots, \Gamma_p$ the components of $H-X$. Now if one of the Γ_i has size greater than $\frac{1}{2}n$, then since $(\cup_{j\neq i}\Gamma_j)\cup X$ meets the semi-expansion condition, so does X. Otherwise, for each i we have $|\delta(\Gamma_i)| \geq \alpha\sqrt{|\Gamma_i|}$, whence

$$|\delta(X)| \geq \alpha \sum_i \sqrt{|\Gamma_i|} \geq \alpha\sqrt{|X|}.$$

■

Proposition 3.10 *If G is locally well-formed with parameters $\Delta, \ell, \gamma_0, \gamma_1, \rho_0, \rho_1$, then there are positive constants α and λ such that G is uniformly densely embedded with parameters α, λ, Δ, and ℓ.*

Proof. Let G be locally well-formed with the given parameters. Then for any $v \in V$, if $s \leq \rho_1^{-1}\log n$, the set $B_s(v)$ is contained in $B'_{r\log n}(v)$ and hence in a Σ-disc. Now let $X \subset B_s(v)$; we wish to show that it satisfies the semi-expansion requirement in $G[B_s(v)]$. By Lemma 3.9, we may assume that both $G[X]$ and $G[B_s(v)-X]$ are connected. Thus $\delta(X)$ lies on a single face of $G[X]$. Let $q = |\delta(X)|$; then there is a closed curve L on Σ of length at most $\rho_1 rq$ that bounds a Σ-disc containing X. Thus, X is contained in a disc of area at most $\gamma_1\rho_1^2r^2q^2$. But each face in $G[X]$ has area at least $\gamma_0\rho^2r^2$, so X has at most $\gamma_1\rho_1^2\gamma_0^{-1}\rho_0^{-2}q^2$ faces, and hence at most ℓ times this many vertices. ■

In a series of papers proving, among other things, that the disjoint paths problem for a fixed number of terminal pairs is solvable in polynomial time [26], Robertson and Seymour make use of another notion of "denseness" of surface embeddings — namely *representativity.* It turns out that our definition of uniformly densely embedded graphs could also have been expressed in these terms. We say that a closed curve on Σ is *null-homotopic* if it is homotopic to a point; it is well-known (see e.g. [24]) that a closed curve is null-homotopic if and only if it is contained in a Σ-disc. Now we say that a drawing of G on Σ is *c-representative* [24, 25] if any non-null-homotopic closed curve on Σ meets the drawing of G at least c times.

In this terminology, we could have replaced the condition that each $G[B_r(v)]$ ($r \leq \lambda\log n$) be contained in a Σ-disc by the condition that the drawing of G be $\Omega(\log n)$-representative. More precisely,

Proposition 3.11 *If G satisfies parts (i) and (iii) of Definition 3.2, and the drawing of G is $(\lambda\log n)$-representative, then there is a constant λ' such that G is uniformly densely embedded with parameters α, λ', Δ, and ℓ. Conversely, if G is uniformly densely embedded with parameters α, λ, Δ, and ℓ, then there is a constant λ' such that the drawing of G is $(\lambda'\log n)$-representative.*

Proof. The converse statement is easier. If G is uniformly densely embedded with parameters α, λ, Δ, and ℓ, then any closed curve $\mathcal{R}$ on Σ meeting G at fewer than $\ell^{-1}\lambda\log n$ vertices meets it only at vertices contained in $B_{\lambda\log n}(v)$ for some $v \in V$. Thus $\mathcal{R}$ is contained in a Σ-disc and is null-homotopic.

Now suppose G satisfies parts (i) and (iii) of Definition 3.2, and the drawing of G is $(\lambda \log n)$-representative. We must show that for some λ', every $G[B_{\lambda' \log n}(v)]$ is drawn in a Σ-disc. Choose $\lambda' < \frac{1}{4}\lambda$ and let $U = B_{\lambda' \log n}(v)$ for some $v \in V$. We claim that every simple cycle of $G[U]$ is null-homotopic in Σ. For suppose not, and choose the shortest non-null-homotopic cycle Q contained in $G[U]$. Say for simplicity that Q contains an even number of vertices, $v_0, \ldots, v_k, \ldots, v_{2k} = v_0$, and let Q_0 and Q_1 denote the two sub-paths of Q with ends equal to v_0 and v_k. Now suppose there were some path P in $G[U]$ with ends equal to v_0 and v_k of length less than k; then one of $Q_0 \cup P$ or $Q_1 \cup P$ would contain a non-null-homotopic simple cycle of $G[U]$ of length less than $2k$, contradicting our choice of Q. Thus $k \leq 2\lambda' \log n$, and so $|Q| \leq 4\lambda' \log n < \lambda \log n$. Now since G is strongly embedded, there is a simple closed curve $\mathcal{R}$ on Σ, meeting G at precisely the vertices of Q, that is homotopic to Q in Σ; but since $\mathcal{R}$ meets G fewer than $\lambda \log n$ times, it is null-homotopic in Σ. This contradicts our assumption that Q is non-null-homotopic.

Thus $G[U]$ contains only null-homotopic simple cycles. By Theorems (11.2) and (11.10) of [24], this implies that $G[U]$ is contained in a Σ-disc. ■

4 Pre-Processing the Graph

Much of the algorithm described in Section 2 for the mesh can be applied in the general case. Of course, we cannot define "squares" in G anymore; but we can use sets of the form $B_r(v)$ instead, and we have seen above that these behave in much the same way. We similarly may choose a maximal set of mutually distant spheres and grow enclosures around them. There are two immediate problems with this approach. (1) We used the natural crossbars inside a mesh for routing; do these enclosures have similar crossbars inside them? (2) Where is the high-capacity simulated network $\mathcal{N}$?

To build crossbars inside the enclosures we use the Okamura-Seymour theorem [18], analogously to a construction in our earlier paper [12]. To define the high-capacity simulated network $\mathcal{N}$ we want to grow the enclosures out until they touch. However, at this point their boundaries might not be "smooth" enough to allow us to build crossbars inside them; additionally, there is no reason for enclosures that do touch to have $\Omega(\log n)$ edges in their common boundary.

Nevertheless, it is still possible to build a simulated network $\mathcal{N}$, as follows. We grow enclosures that have smooth boundaries, and are large enough that they contain large crossbars, but we keep them mutually distant from one another. Then we define the notion of a *Voronoi partition* of G to allow us to determine which clusters are "neighbors"; we define the simulated network $\mathcal{N}$ by putting $\Omega(\log n)$ parallel edges between neighboring clusters (whether or not they have that many edges in the common boundary of their Voronoi regions).

We show that the collection of these paths "represents" the graph G well enough that it can be used as the network $\mathcal{N}$. In particular, we need to show that all paths accepted by the simulated network can be routed in G. For this we make use of a theorem of Schrijver [27]; we show that there exist $\Omega(\log n)$ paths in G between the neighboring enclosures, such that all paths between all pairs are mutually disjoint.

We make no attempt to optimize constants here. Set $\lambda_0 = \lambda$, and choose positive constants

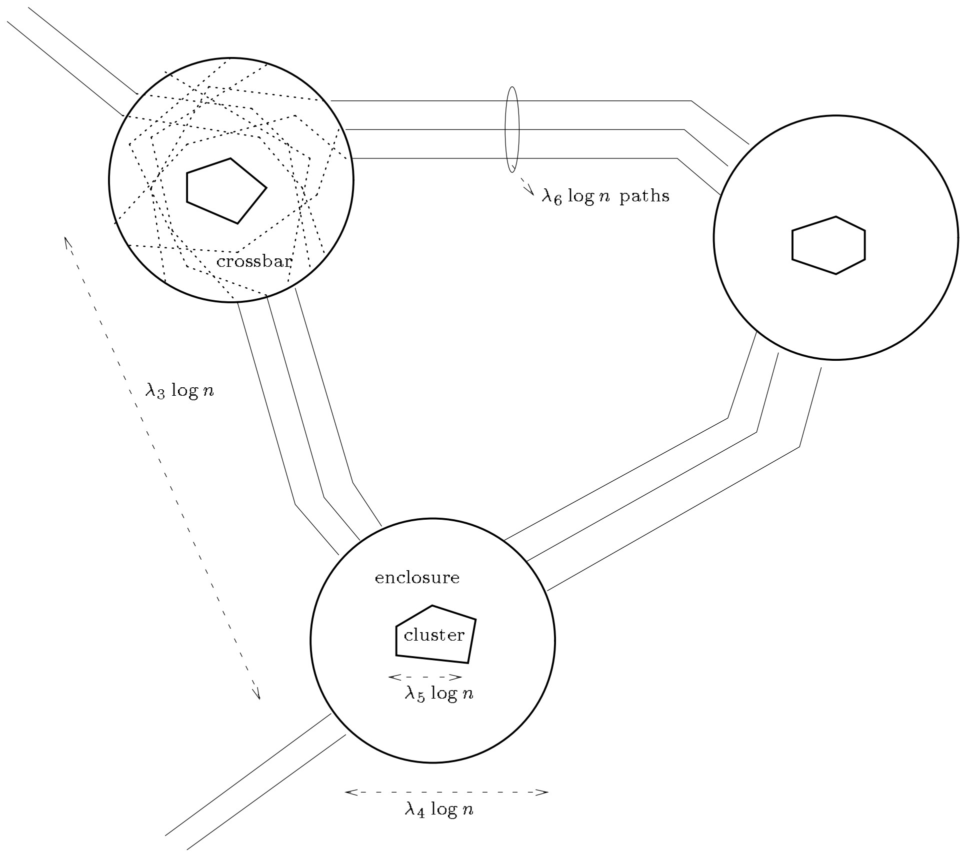

Figure 1: Building the simulated network

$\lambda_1, \lambda_2, \ldots$ so that $\lambda_{j+1} \ll \lambda_j$ (the exact relationship between these constants is easy to determine from the analysis below). A connection (s_i, t_i) is *short* if $d(s_i, t_i) \leq \lambda_2 \log n$ and *long* otherwise. Our on-line algorithm makes an initial random decision, whether to route short connections or long connections. The case of short connections is fairly straightforward, and will be discussed in Section 5. In this section, we are concerned with pre-processing the graph for routing long connections.

4.1 Clusters and Enclosures

We wish to choose a maximal set of mutually distant vertices around which to grow clusters. Let G^r denote the graph obtained from G by joining u and v if $d(u,v) \leq r$. We first run a randomized version of Luby's maximal independent set algorithm [15] in $G^{\lambda_3 \log n}$. That is, each vertex picks a random number between 1 and j, where j is large enough that the probability of ties is small. If v has a number higher than any of its neighbors', it enters the MIS and its neighbors drop out. We then iterate.

Let M denote the resulting MIS. For any $x \in V$, some vertex within $\lambda_5 \log n$ of it will enter M on the first iteration if the largest number chosen in $B_{2\lambda_3 \log n}(x)$ is chosen by a vertex in $B_{\lambda_5 \log n}(x)$. This happens with constant probability, by Lemma 3.4. Moreover, if $d(x,y) \geq \lambda_2 \log n$, then these events are independent for x and y. Thus,

Lemma 4.1 *Let $x, y \in V$ be such that $d(x,y) \geq \lambda_2 \log n$. Then with constant probability there are $u, v \in M$ such that $d(x,u) \leq \lambda_5 \log n$ and $d(y,v) \leq \lambda_5 \log n$.*

Around each $v \in M$ we now grow a *cluster* of radius roughly $\lambda_5 \log n$, and an *enclosure* around each cluster, with "smooth" boundaries. We need the following facts. Let H denote an arbitrary graph, and Q a simple cycle of H. For $u, v \in Q$, let $d_Q(u,v)$ denote the shortest-path distance from u to v on Q — that is, the length of the shorter of the two u-v paths on Q.

Definition 4.2 *We say that Q is ε-*smooth *if for all $u, v \in Q$ we have $\varepsilon d_Q(u,v) \leq d(u,v)$.*

Definition 4.3 *If U and W are two subsets of $V(H)$, we say that U is ε'-close to W if for each $u \in U$ there is a $w \in W$ such that $d(u,w) \leq \varepsilon'|W|$.*

The following fact is quite similar to, but more general than, Theorem 4.4 of our earlier paper [12]; the proof is very similar as well. In effect, it says that given a cycle Q in a planar graph H that encloses (in the sense of homotopy) the "hole" formed by some internal face, then for a small $\varepsilon > 0$ we can find a cycle Q' of no greater length that is ε-close to Q, $\Omega(\varepsilon)$-smooth, and also encloses this hole. See Figure 2.

We will use this theorem to smooth out the boundaries of the clusters and the enclosures around them.

Theorem 4.4 *For each $\varepsilon > 0$ the following holds. Let H be a planar graph drawn in $\mathbf{R}^2$, Φ the outer face of H, and Φ' an internal face of H. Let Q a simple cycle of H that is non-null-homotopic in the cylinder $\mathbf{R}^2 - (\Phi \cup \Phi')$. Then in polynomial time one can find an $\left(\frac{\varepsilon}{1+\varepsilon}\right)$-smooth simple cycle Q' such that*

(i) $|Q'| \leq |Q|$,

(ii) Q' is ε-close to Q, and

(iii) Q' is also non-null-homotopic in $\mathbf{R}^2 - (\Phi \cup \Phi')$.

Proof. For $u, v \in Q$, let $[u,v]_Q$ denote the shorter of the two u-v paths contained in Q (ties broken arbitrarily). Let $r = |Q|$, $\bar{\varepsilon} = \frac{\varepsilon}{1+\varepsilon}$, and Λ denote the cylinder $\mathbf{R}^2 - (\Phi \cup \Phi')$.

If Q is not $\bar{\varepsilon}$-smooth, then there are $u, v \in Q$ such that

$$\bar{\varepsilon} d_Q(u,v) > d(u,v). \tag{1}$$

Moreover, we can efficiently find such a u and v so that there is a shortest u-v path P_{uv} in H that is vertex-disjoint from Q (for example, the pair u, v satisfying (1) for which $|P_{uv}|$ is minimum).

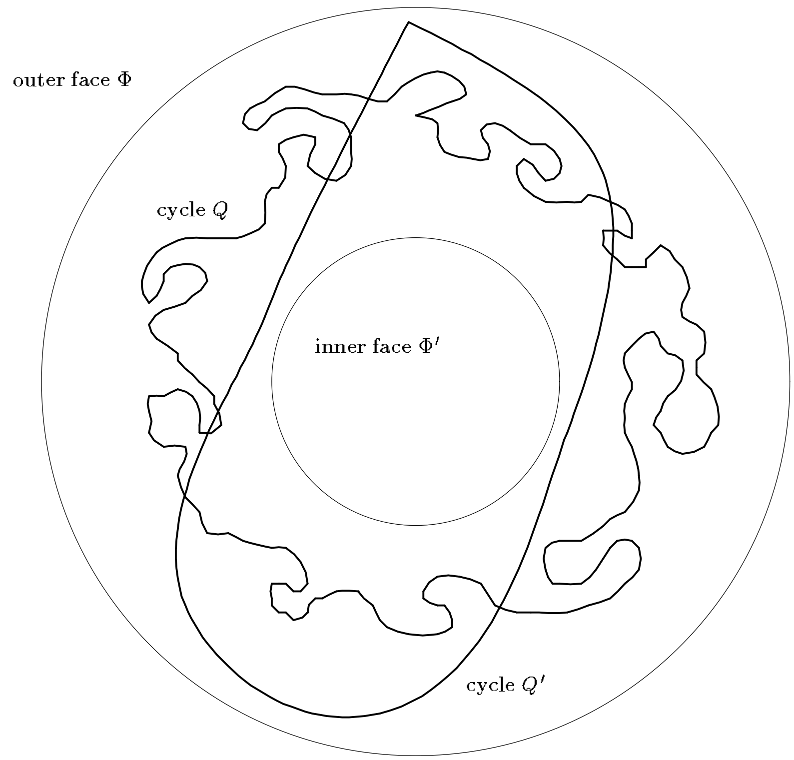

Figure 2: Smoothing a cycle

Now one of the two simple cycles $[u,v]_Q \cup P_{uv}$ and $(Q - [u,v]_Q) \cup P_{uv}$ is not null-homotopic in Λ; and each is shorter than Q. We thus update Q, replacing it with the cycle from among these two that is not null-homotopic.

We now iterate this process of "slicing off" parts of Q using short paths through H. Since the length of the cycle decreases with each iteration, this process must terminate in a cycle Q' that is $\bar{\varepsilon}$-smooth. Moreover, each iteration maintains the invariant that the current cycle is non-null-homotopic in Λ. Thus, we only have to verify that the final cycle is ε-close to Q.

This is clearly true after the first iteration: since $|P_{uv}| < \bar{\varepsilon} d_Q(u,v) < \varepsilon|Q|$, every vertex on the updated cycle can reach a vertex of Q by a path of length at most $\varepsilon|Q|$. Now, let Q_i denote the cycle obtained after i iterations of slicing off. As long as portions of Q remain on Q_i, we say that we are in the "first phase"; other phases will be defined below. In the first phase, Q_i consists of alternating intervals $Q_{i1}, P_{i1}, Q_{i2}, \ldots, Q_{ir}, P_{ir}$, where $Q_{ij} \subset Q$, and the interval Q'_{ij} of Q lying between Q_{ij} and $Q_{i,j+1}$ has been sliced off by P_{ij}. We can show by induction on the number of iterations that $|P_{ij}| \leq \bar{\varepsilon}\left|Q'_{ij}\right|$ — as was true after the first iteration.

This is done by the following case analysis. In the $(i+1)^{\text{st}}$ iteration, we find a new path; there are three cases to consider.

1. One end of P lies on Q_{ij} and the other on Q_{ik}, where possibly $j = k$. Then the property clearly continues to hold, since $|P|$ is at most $\bar{\varepsilon}$ times the number of current cycle vertices cut off, which is in turn at most the number of original vertices of Q between the endpoints of P.

2. One end of P lies on P_{ij} and the other on Q_{ik} (so P_{ij} is lengthened). Suppose that the amount of original cycle cut off *in addition to* Q'_{ij} is equal to x, and the amount of P_{ij} that is cut off by P is y. Then if $P_{i+1,j}$ denotes P_{ij} after this iteration, we have

$$\begin{aligned} |P_{ij}| &\leq \bar{\varepsilon}\left|Q'_{ij}\right| \\ |P| &\leq \bar{\varepsilon}(x+y) \\ |P_{i+1,j}| &= |P_{ij}| + |P| - y \\ &\leq \bar{\varepsilon}(\left|Q'_{ij}\right| + x + y) - y \\ &\leq \bar{\varepsilon}(\left|Q'_{ij}\right| + x) \end{aligned}$$

3. One end of P lies on P_{ij} and the other lies on P_{ik} (so P glues some of the new paths together). There are two subcases.

   (i) $j = k$. Then $|P_{ij}|$ goes down while $|Q_{ij}|$ is not affected, so the property still holds.

   (ii) $j \neq k$. Again suppose that the amount of original cycle cut off in addition to Q'_{ij} and Q'_{ik} is equal to x, the amount of P_{ij} cut off by P is y, the amount of P_{ik} cut off by P is z, and the new interval is denoted $P_{i+1,j}$. Then

$$\begin{aligned} |P_{ij}| &\leq \bar{\varepsilon}\left|Q'_{ij}\right| \\ |P_{ik}| &\leq \bar{\varepsilon}|Q'_{ik}| \\ |P| &\leq \bar{\varepsilon}(x+y+z) \\ |P_{i+1,j}| &= |P_{ij}| + |P| + |P_{ik}| - y - z \\ &\leq \bar{\varepsilon}(\left|Q'_{ij}\right| + x + y + z + |Q'_{ik}|) - y - z \\ &\leq \bar{\varepsilon}(\left|Q'_{ij}\right| + x + |Q'_{ik}|) \end{aligned}$$

If the iterations come to an end before the end of the first phase, then indeed Q' is ε-close to Q — any vertex on P_{ij} can reach Q by a path of length at most $\bar{\varepsilon}\left|Q'_{ij}\right| \leq \bar{\varepsilon}|Q|$. Otherwise, consider the iteration in which the first phase comes to an end. By analogous arguments, we obtain a cycle Q^1 such that $|Q^1| \leq \bar{\varepsilon}|Q|$ and every vertex on Q^1 can reach Q by a path of length at most $\bar{\varepsilon}|Q|$.

Each phase now proceeds exactly like the previous one, except that it begins with a cycle whose length has been reduced by at least a factor of $\bar{\varepsilon}$. Thus when the process terminates, all vertices on Q' will be able to reach Q by a path of length at most

$$|Q| \cdot \sum_{i=1}^{\infty} \bar{\varepsilon}^i = \varepsilon|Q|.$$

Thus Q' is ε-close to Q. ■

We use the following procedure to build the clusters and the enclosures around each node v in M. Let $K_v = B_{\lambda_5 \log n}(v)$.

(i) Choose a radius r between $2\lambda_5 \log n$ and $3\lambda_5 \log n$ so that $|B_r(v)| \leq 9\beta\lambda_5 \log n$. Set $C_v = B_r(v)$.

(ii) Now extend C_v to a simple set as in Lemma 3.6; since $\lambda_3 > 2c\lambda_5$, no C_v is grown enough that it overlaps any other by this process.

(iii) We now apply the ε-smoothing algorithm of Theorem 4.4 to the facial cycle Q_v of $G[C_v]$ containing $\pi(C_v)$. Here $H_v = G[B_{\lambda_3 \log n}(v)] - K_v^o$ plays the role of H, and the "hole" left by the deletion of K_v^o plays the role of the internal face Φ'. Now for a constant ε the resulting cycle Q_v' is ε-smooth in this subgraph H_v of G, and it is also $\frac{\varepsilon}{1-\varepsilon}$-close to Q_v. If we choose $\varepsilon < \frac{1}{18\beta+1}$, then since Q_v is initially at least $\frac{1}{9\beta}|Q_v|$ away from K_v, we know that any path with both ends on Q_v' that passes through K_v must have length at least $\frac{1}{9\beta}|Q_v| \geq \frac{1}{9\beta}|Q_v'|$. Thus, there are no "short cuts" between vertices of Q_v' that make use of K_v; hence Q_v' is in fact ε-smooth in G.

The smooth cycle Q_v' encloses a set S of vertices containing K_v. Update C_v to be this set S.

We now grow an *enclosure* $D_v \supset C_v$ by the same three-step process, except that we now use the constant λ_4 in place of λ_5, and the set C_v^o in place of K_v^o as the internal face Φ'. Thus, we have clusters of radius $\approx \lambda_5 \log n$, enclosures of radius $\approx \lambda_4 \log n$, and they are separated by a mutual distance of $\approx \lambda_3 \log n$.

Following the outline of Section 2, we now must build crossbar structures in the enclosures to replace the natural crossbars of the mesh. We build crossbars using an extension of the Okamura–Seymour theorem [18] due to Frank [8], along the same lines as was done in [12]. To be precise,

Definition 4.5 *If $X \subset V$, we say a* crossbar *anchored in X is a set of edge-disjoint paths, each with both ends in X, such that every pair of paths meets in at least one vertex.*

Let $\tau_v = \pi(D_v)$, and φ_v the facial cycle of D_v containing τ_v. We wish to build a crossbar in $G[D_v - C_v^o]$, anchored in τ_v, of size at least a constant fraction of $|\tau_v|$. For a large enough constant κ depending on ε, we choose a set S of $|\tau_v|/\kappa$ vertices on τ_v spaced about κ apart. Let us require that $|S| \equiv 2 \bmod 4$; so for each $u \in S$ we can identify a unique "antipodal" vertex $\tilde{u}$ in S, and we can then choose one vertex from each antipodal pair so that every second vertex in S is chosen. Let τ_v' denote the set of chosen vertices.

We now set up a disjoint paths problem in the (planar) graph $G[D_v - C_v^o]$, with the set $\mathcal{T}_v$ of terminal pairs equal to all $(u, \tilde{u})$ for $u \in \tau_v'$.

Lemma 4.6 *$\mathcal{T}_v$ is realizable in $G[D_v - C_v^o]$.*

Proof. Say that a cut is *non-trivial* if it separates at least one pair of terminals. In a planar graph with all terminal pairs on a single face Φ, and in which all vertices not incident to Φ have even degree, Frank's extension [8] of the Okamura-Seymour theorem [18] says that the following *strict cut condition* is sufficient for the realizability of the terminal pairs: every non-trivial cut has more capacity than the number of terminal pairs it separates.

Now, writing $G_v = G[D_v - C_v^o]$, we essentially have such a disjoint paths problem: all terminal pairs in $\mathcal{T}_v$ belong to the outer facial cycle φ_v of G_v; and G_v has even degree everywhere but on this cycle φ_v *and* on the inner facial cycle ι_v left by the deletion of C_v^o. Thus, to satisfy the hypotheses of the theorem we need only modify G_v so that it contains no odd-degree vertices on ι_v. Now there are necessarily an even number of such odd vertices on ι_v, and we can remove sub-paths between consecutive pairs of such vertices so as reduce each of their degrees by 1 (and reduce all other degrees on ι_v by either 0 or 2).

Following this modification, G_v has even degree everywhere but on the face φ_v containing the terminal pairs; so by Frank's theorem it is now enough to verify the strict cut condition. For this we use Lemma 3.7. Write $G_v = G[D_v - C_v^o]$; note that the only faces of G_v that are bounded by more than ℓ edges are the outer face bounded by φ_v, and the inner face left by the deletion of C_v^o. Moreover, these two faces are a distance at least $(\lambda_4 - 4\lambda_5)\log n$ apart.

It is enough to consider non-trivial cuts of the form $\delta(U)$ with both $G_v[U]$ and $G_v - U$ connected. For such a set U, there must be two vertices $v, w \in \pi(U)$ such that $v, w \in \varphi_v$. Suppose that the distance from v to w in G_v is d; then by Lemma 3.7, we have

$$|\delta(U)| \geq \min(\ell^{-1}d, \ell^{-1}(\lambda_4 - 4\lambda_5)\log n)).$$

Since the facial cycle φ_v is ε-smooth, the number of terminal pairs disconnected by $\delta(U)$ is at most $\varepsilon^{-1}d/\kappa$. Since $d \leq 8\lambda_4 \log n$, taking $\kappa > 8\varepsilon^{-1}\ell\lambda_4(\lambda_4 - 4\lambda_5)^{-1}$ ensures that the strict cut condition will be satisfied. ■

Now in the set of paths realizing $\mathcal{T}_v$, let Y_v^u denote the u-$\tilde{u}$ path. Then the collection $\{Y_v^u\}$ is a crossbar, since every pair of such paths must cross in the plane drawing of $G[D_v - C_v^o]$.

4.2 The simulated network $\mathcal{N}$

We now construct a *simulated network*; the nodes of this network are the clusters, which we represent by the vertices in M. We define a neighbor relation on the clusters using the notion of a *Voronoi partition*; two clusters will be joined by an edge in $\mathcal{N}$ if they are neighbors in this sense.

Let H be a graph and $S \subset V(H)$. We fix a lexicographic ordering $\preceq$ on the elements of S. For $s \in S$, let

$$\mathcal{U}_s = \{v \in G : \forall s' \in S : d(v,s) \leq d(v,s') \text{ and } \forall s' \preceq s : d(v,s) < d(v,s')\}.$$

That is, $\mathcal{U}_s$ is the set of vertices that are at least as close to s as to any other s', with ties broken based on $\preceq$.

Definition 4.7 *The* Voronoi partition $\mathcal{V}(H,S)$ *of H with respect to S is the partition* $\{\mathcal{U}_s : s \in S\}$.

The following fact is immediate.

Lemma 4.8 *For each $s \in S$, $H[\mathcal{U}_s]$ is connected.*

Proof. Suppose $v \in \mathcal{U}_s$; we claim that any shortest s-v path P is contained in $\mathcal{U}_s$. For suppose not, and let $v' \in P$ be the closest vertex to s that lies in $\mathcal{U}_{s'}$ for some $s' \neq s$. Then $d(s', v) \leq d(s, v)$, and in fact $d(s', v) < d(s, v)$ if $s \preceq s'$. It follows that $v \in \mathcal{U}_{s'}$, a contradiction. ■

We can now build a graph $\mathcal{N}(H, S)$ on the vertices in S, joining two if their Voronoi cells share an edge.

Definition 4.9 *The* neighborhood graph *of S in H, denoted $\mathcal{N}(H, S)$, is the graph with vertex set S, and an edge (s, s') iff there is an edge of H with endpoints in $\mathcal{U}_s$ and $\mathcal{U}_{s'}$.*

The simulated graph we use will be the neighborhood graph $\mathcal{N}(G, M)$ with every edge given capacity $\approx \lambda_6 \log n$. Let $\mathcal{V}$ and $\mathcal{N}$ denote $\mathcal{V}(G, M)$ and $\mathcal{N}(G, M)$ respectively, and $\mathcal{N}(\gamma)$ the graph $\mathcal{N}$ in which each edge is given capacity γ.

The following two facts about $\mathcal{N}$ are easy to establish. First, by the maximality of M, we have

Lemma 4.10 *For all $v \in M$, $\mathcal{U}_v \subset B_{\lambda_3 \log n}(v)$.*

Proof. Suppose $u \in \mathcal{U}_v$ but $d(v, u) > \lambda_3 \log n$. Then $d(v', u) > \lambda_3 \log n$ for *all* $v' \in M$; this contradicts the fact that M is a maximal independent set in $G^{\lambda_3 \log n}$. ■

This, along with Lemma 3.4, implies

Lemma 4.11 *The degree of a vertex in $\mathcal{N}$ is at most $\Delta' \leq 16\bar{\alpha}^{-1}\beta$.*

Proof. Let U denote the neighbors of v in $\mathcal{N}$, including v itself. Then by Lemma 4.10,

$$\bigcup_{u \in U} \mathcal{U}_u \subset B_{2\lambda_3 \log n}(v),$$

and hence

$$\sum_{u \in U} |\mathcal{U}_u| \leq 4\beta\lambda_3^2 \log^2 n.$$

But around each $u \in U$ there is a disjoint ball of radius $\frac{1}{2}\lambda_3 \log n$, which contains at least $\bar{\alpha}\lambda_3^2 \log^2 n/4$ vertices. Thus $|U| \leq 16\bar{\alpha}^{-1}\beta$. ■

We now attempt to make explicit a sense in which the network $\mathcal{N}(\Theta(\log n))$ "represents" the graph G sufficiently well: for any pair of neighboring Voronoi cells $\mathcal{U}_v, \mathcal{U}_w$, we show that at most $O(\log n)$ paths in the optimal routing can cross an edge in $\delta(\mathcal{U}_v, \mathcal{U}_w)$.

Lemma 4.12 *If $(s_i, t_i) \in \mathcal{T}$, let $\psi(s_i, t_i)$ denote the pair of clusters containing (s_i, t_i). There is a constant γ such that for any realizable subset $\mathcal{T}'$ of $\mathcal{T}$, $\psi(\mathcal{T}')$ can be routed in $\mathcal{N}(\gamma \log n)$.*

Proof. Set $\gamma = 9\beta(\Delta'^3\lambda_5 + \lambda_3)$. For each s_i-t_i path P in the optimal routing, construct the following path for (u_i, v_i) in $\mathcal{N}$ — when P crosses from $\mathcal{U}_w$ into $\mathcal{U}_{w'}$, add an edge from w to w'. Now consider how many paths in our constructed routing use the edge (w, w'). Each such corresponds to a path in the original routing that used an edge in $\delta(\mathcal{U}_w,\mathcal{U}_{w'})$. We can't bound the size of this set directly, since it could be quite "meandering." But consider the following argument. $\delta(\mathcal{U}_w,\mathcal{U}_{w'}) \subset B_{2\lambda_3 \log n}(x)$ for some vertex $x \in G$; thus there is some r between $2\lambda_3 \log n$ and $3\lambda_3 \log n$ so that $|\pi(B_r(x))| \leq 9\beta\lambda_3 \log n$. So at most this many paths with both ends more than r away from x can use edge in $\delta(\mathcal{U}_w,\mathcal{U}_{w'})$. But closer than this, there are at most Δ'^3 clusters, each of which is the origin of at most $9\beta\lambda_5 \log n$ paths. ■

4.3 Inter-Cluster Paths

The goal of this part is, for a constant λ_6, to construct $\lambda_6 \log n$ disjoint paths between each pair of enclosures D_v, D_w where (v, w) is an edge in $\mathcal{N}$. This will allow us to convert a routing in the simulated network $\mathcal{N}(\lambda_6 \log n)$ into actual disjoint paths in G. Recall that the outer facial cycle of $G[D_v]$ is denoted φ_v, and it contains a set τ'_v of vertices evenly spaced at distance κ.

Theorem 4.13 *There exist vertex-disjoint paths in G, each with ends in sets τ'_v and otherwise disjoint from all D_v, such that for $(v, w) \in \mathcal{N}$, there are at least $\lambda_6 \log n$ such paths with one end in τ'_v and the other in τ'_w.*

Proof. The proof is based on the following theorem of Schrijver [27]. Let Σ_1 be a surface (possibly with boundary), H a graph embedded on Σ_1, and $\{A^i : i = 1, \ldots, k\}$ a set of disjoint curves on Σ_1 each of which is either closed or anchored at vertices of H on the boundary of Σ_1. The problem is to find vertex-disjoint paths and cycles $\{P^i\}$ in G so that P^i is homotopic to A^i for each i.

Call a collection of curves $\overline{\mathcal{R}}$ on Σ_1 *essential* if it consists either of a single closed curve that is not null-homotopic, or it is a finite union of curves each with endpoints on the boundary of Σ_1. Schrijver [27] proves that such vertex-disjoint paths and cycles exist if for each essential collection of curves $\overline{\mathcal{R}}$, there are curves $\{B^i\}$, where B^i is homotopic to A^i, such that $\overline{\mathcal{R}}$ intersects the drawing of H more than it intersects the curves $\{B^i\}$. (The main result of [27] is in fact a necessary and sufficient condition; this weaker statement suffices for our purposes. Additionally, [27] is stated for the special case of surfaces without boundary, but the extension we use here follows immediately from [27].)

Say that a curve is G-normal if it meets the drawing of G only at vertices, and define its G-length to be the number of times it meets the drawing. For each v, v' that are neighbors in $\mathcal{N}$, we draw a G-normal arc $\mathcal{A}_{vv'}$ on Σ with endpoints v and v'. We can ensure that all these arcs are disjoint, since each $\mathcal{U}_v$ is connected, and for each $(v, v') \in E(\mathcal{N})$, there is at least one edge of G with endpoints in $\mathcal{U}_v$ and $\mathcal{U}_{v'}$. Choose a small constant $\lambda_6 \leq |\tau'_v|/\Delta'$ (say, less than $\frac{1}{8}\alpha^2\bar{\alpha}\Delta'^{-2}\Delta^{-2}\lambda_3^{-1}\lambda_4^2$; the reason for this will become clear below). Now suppose we have in fact $\lambda_6 \log n$ copies of each $\mathcal{A}_{vv'}$, all running "parallel" to one another. By pushing them apart appropriately, we can assume that each arc runs through a different vertex in τ'_v and $\tau'_{v'}$. Let $A_{uu'}$ denote the G-normal arc that runs through $u \in \tau'_v$ and $u' \in \tau'_{v'}$.

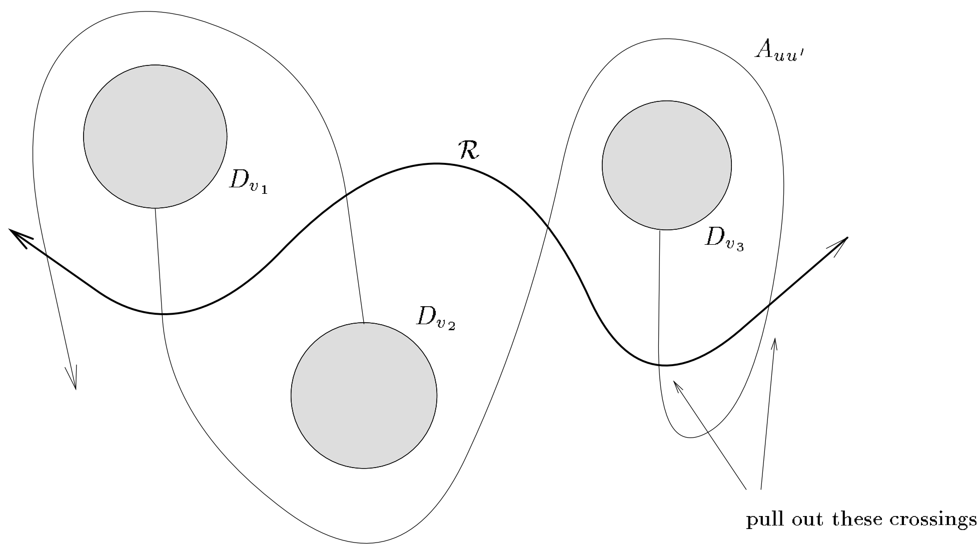

Figure 3: Pulling out a crossing

Define G' to be the graph obtained by deleting D_v^o for each $v \in M$. We now cut Σ along the facial cycles φ_v to obtain Σ', a surface with boundary. Note that G' is properly embedded on Σ'. The theorem is now a consequence of the following claim. ■

Claim 4.14 *There exist vertex-disjoint paths $P_{uu'}$ in G' such that $P_{uu'}$ is homotopic to $A_{uu'}$.*

Proof. If curves $\mathcal{R}$ and $\mathcal{R}'$ are homotopic, we write $\mathcal{R} \sim \mathcal{R}'$. We extend this notation to finite collections of curves $\overline{\mathcal{R}}, \overline{\mathcal{R}}'$ in the obvious way. Following notation of Schrijver [27], the number of crossings of $\mathcal{R}_0$ and $\mathcal{R}_1$ is denoted $\mathrm{cr}(\mathcal{R}_0, \mathcal{R}_1)$, and we define

$$\mathrm{mincr}(\mathcal{R}_0, \mathcal{R}_1) = \min\{\mathrm{cr}(\mathcal{R}_0', \mathcal{R}_1') : \mathcal{R}_0 \sim \mathcal{R}_0', \mathcal{R}_1 \sim \mathcal{R}_1'\}.$$

By the theorem of Schrijver [27] given above, the desired paths exist if for each essential collection of curves $\overline{\mathcal{R}}$ on Σ', one has

$$\mathrm{cr}(\overline{\mathcal{R}}, G') > \sum_{(u,u')} \mathrm{mincr}(\overline{\mathcal{R}}, A_{uu'}). \tag{2}$$

Note that in verifying this inequality, we may assume $\overline{\mathcal{R}}$ is G'-normal, and that $\overline{\mathcal{R}}$ has no self-crossings.

For a collection of curves $\overline{\mathcal{R}}$, define its *index* to be

$$\mathrm{cr}(\overline{\mathcal{R}}, G') - \sum_{(u,u')} \mathrm{mincr}(\overline{\mathcal{R}}, A_{uu'}).$$

So it is enough to consider collections of curves $\overline{\mathcal{R}}$ whose index is minimum in their homotopy class, and to show that such $\overline{\mathcal{R}}$ in fact have positive index.

Set $r = \frac{1}{2}\Delta^{-1}\alpha\bar{\alpha}^{1/2}\lambda_4 \log n$. We claim that no G-normal closed curve of G-length less than $2r$ can enclose a set of the form D_v. For if it did, then D_v could be disconnected by an edge cut of size less than $\alpha\bar{\alpha}^{1/2}\lambda_4 \log n$, which is not possible since $|D_v| \geq \bar{\alpha}\lambda_4^2 \log^2 n$. From this it follows that any G'-normal closed curve of G'-length less than $2r$ must be null-homotopic in Σ'.

Note that we can view the expression $\mathrm{cr}(\overline{\mathcal{R}}, G') - \sum_{(u,u')} \mathrm{cr}(\overline{\mathcal{R}}, A_{uu'})$ as a sum over the finitely many arc-components of $\overline{\mathcal{R}}$; if the value of this expression is not positive, we show how modify the curves $A_{uu'}$ so that it increases. We do this by considering each arc-component of $\overline{\mathcal{R}}$ in turn. Let $\mathcal{R}$ denote a single arc-component of $\overline{\mathcal{R}}$; we consider two cases, based on the G'-length of $\mathcal{R}$.

Case 1. $\mathrm{cr}(\mathcal{R}, G') \leq r$. Then $\mathcal{R}$ must have both endpoints on the same facial cycle (it is too short to touch two such cycles, and if it were a closed curve it would have to be null-homotopic, by the above argument.) But then it is easy to produce arcs $\{A'_{uu}\}$ for which $\mathrm{cr}(\mathcal{R}, G') > \sum_{(u,u')} \mathrm{cr}(\mathcal{R}, A'_{uu'})$ since φ_v is ε-smooth.

Case 2. $\mathrm{cr}(\mathcal{R}, G') > r$. Again, we just have to exhibit arcs $A'_{uu'} \sim A_{uu'}$ lying on Σ' so that

$$\mathrm{cr}(\mathcal{R}, G') > \sum_{(u,u')} \mathrm{cr}(\mathcal{R}, A'_{uu'}), \tag{3}$$

without increasing the number of crossings of these arcs with the other components of $\overline{\mathcal{R}}$. If the set $\{A_{uu'}\}$ satisfies (3), we are done; otherwise, we show how to modify this set of arcs so that it does. See Figure 3.

If the set $\{A_{uu'}\}$ does not satisfy Inequality (3), then there is some interval $\mathcal{R}'$ of $\mathcal{R}$ of G'-length r for which (3) is violated. Let us consider such an $\mathcal{R}'$.

Observe that each arc $A_{uu'}$ has G'-length at most $2\lambda_3 \log n$, and hence at most $\Delta'^2\lambda_6 \log n$ of these arcs can meet $\mathcal{R}'$, since at most Δ'^2 pairs of clusters have at least one end close enough to $\mathcal{R}'$. Now suppose the total number of crossings of these arcs with $\mathcal{R}'$ exceeds

$$\frac{(2\lambda_3 \log n)(\Delta'^2\lambda_6 \log n)}{r} < \mathrm{cr}(\mathcal{R}', G).$$

Then some arc $A_{uu'}$ meets $\mathcal{R}'$ more than $2\lambda_3 \log n/r$ times, and hence the interval of $A_{uu'}$ between some pair of consecutive crossings with $\mathcal{R}'$ has G'-length less than r.

Suppose that this pair of consecutive crossings occurs at vertices w and w'. Let $\mathcal{R}''$ denote the G'-normal curve formed from this interval of $A_{uu'}$ and the portion of $\mathcal{R}'$ between w and w'. $\mathcal{R}''$ has G'-length less than $2r$, and so it must be null-homotopic by the argument given above.

Now, since $\mathcal{R}$ has minimum index over all curves in its homotopy class, the portion of $A_{uu'}$ between w and w' meets G' at least as many times as the portion of $\mathcal{R}'$ between w and w'. We can therefore modify $A_{uu'}$ so that it runs along $\mathcal{R}'$ for this interval. This does not increase the G'-length of $A_{uu'}$; and it decreases the number of crossings of $\mathcal{R}$ — as well as $\overline{\mathcal{R}}$ (since it has no self-crossings) — with $A_{uu'}$.

Thus this process terminates; when it does, we have a set of arcs $\{A'_{uu'}\}$ for which Inequality 3 holds. ■

Let us denote one such path with ends u and u' by $Z_{uu'}$. Moreover, we have $u \in \tau'_v$ and $u' \in \tau'_{v'}$, and they are ends of paths Y_v^u and $Y_{v'}^{u'}$ respectively. Denote by $\bar{Z}_{uu'}$ the concatenation of the three paths Y_v^u, $Z_{uu'}$, and $Y_{v'}^{u'}$.

5 The On-Line Algorithm

5.1 Routing Short Connections

Recall that a connection is *short* if $d(s_i, t_i) \leq \lambda_2 \log n$. Our algorithm made an initial random decision, whether to accept only short connections or only long connections. To handle short connections, we require the following two facts.

Proposition 5.1 *Let $H = (V, E)$ be an arbitrary graph of diameter d. Then there is a deterministic on-line* MDPP *algorithm that is $2 \cdot \max(d, \sqrt{|E|})$-competitive.*

Proof. Let $m = |E|$. The algorithm maintains a sequence of graphs $H_1, H_2, \ldots$ as follows. $H_1 = H$. The algorithm always routes the first request on a shortest path P_1, and sets $H_2 = H_1 - P_1$. In general, when presented with request (s_i, t_i), the algorithm routes it on a shortest path P_i in H_i *if* $d(s_i, t_i) \leq \sqrt{m}$ in H_i. It then sets $H_{i+1} = H_i - P_i$. Let p denote the total number of paths routed by the algorithm.

Let $d' = \max(d, \sqrt{m})$. Consider any routing for $\mathcal{T}$, consisting of paths $Q_1, \ldots, Q_q$. At most d' of the Q_j intersect each P_i, since the Q_i are all edge-disjoint and $|P_i| \leq d'$. Also, at most $\sqrt{m}$ of the Q_j fail to intersect any of the P_i, since the pair (s_j, t_j) associated with Q_j must have been rejected by the on-line algorithm, and hence $|Q_j| > \sqrt{m}$. Thus we have $q \leq d'p + \sqrt{m} \leq 2d'p$. ■

Lemma 5.2 *Let $r \leq \lambda_1 \log n$, $U \subset B_r(v)$ for some $v \in V$, and $\mathcal{T}'$ a set of terminal pairs in U. Then the maximum size of a subset of $\mathcal{T}'$ that is realizable in $B_{8\xi^2 r}(v)$ is within a constant factor of the maximum size of a subset of $\mathcal{T}'$ that is realizable in G.*

Proof. First choose a radius r' between $2r$ and $3r$ for which $|\pi(B_r(v)| \leq 9\beta r'$. Then construct a simple set extension of $B_{r'}(v)$ as in Lemma 3.6; and ε-smooth its outer facial cycle to obtain a set $U' \supset U$ contained in $B_{4\xi r}(v)$. Let $U'' \subset B_{8\xi^2 r}$ denote a simple set extension of $B_{8\xi r}(v)$, as in Lemma 3.6. For a constant κ', we can pick a set S of vertices on the outer facial cycle of U' spaced κ' apart, and use Frank's theorem [8] as in Lemma 4.6 to construct a set of edge-disjoint paths connecting "antipodal" pairs in S, such that all paths stay within $U'' - U'$. Note that we must take care to ensure that the parity condition is met, since the outer facial cycle of U'' can contain odd-degree vertices. But this is handled as in the proof of Lemma 4.6: we remove sub-paths of this cycle between consecutive pairs of the (necessarily even number of) odd-degree vertices; U'' is large enough that the strict cut condition will remain satisfied.

Consider the set of paths in a realization of a maximum-size subset of $\mathcal{T}'$ in G. If at least half these paths never leave U', we are done. Otherwise, of the paths that meet $\pi(U')$, we can select a constant fraction of pairs of paths that can all be connected along the outer facial cycle of U' to different vertices in S. We can then use the crossbar of the previous paragraph to connect all of these pairs together; the resulting paths are within a constant fraction of the maximum number achievable in G, and they do not leave U''. ■

The algorithm for short connections is now as follows. We run a randomized version of Luby's algorithm, this time in $G^{\lambda_1 \log n}$. Let M' denote the resulting MIS. With constant probability, both ends of a short connection are within $\frac{1}{16\xi^2}\lambda_1 \log n$ of the same $v \in M'$, as in Lemma 4.1. We now let U_v denote $B_{\lambda_1 \log n/16\xi^2}(v)$ and only route connections both of whose ends lie in the same U_v. To route such connections, we run the algorithm of Proposition 5.1 in each $B_{\frac{1}{2}\lambda_1 \log n}(v)$; by Lemma 5.2, this is within $O(\log n)$ of optimal in each U_v.

5.2 The AAP Algorithm

If H is a graph with n nodes in which each edge has capacity at least $\log 2n$, then there is an on-line MDPP algorithm of Awerbuch, Azar, and Plotkin [3] that achieves a competitive ratio of $2 \log 4n$. For our purposes, we need a strengthening of this "AAP algorithm" — we want only to require capacities to be $\varepsilon \log n$, for any $\varepsilon > 0$, and to be competitive against the fractional optimum.

Proposition 5.3 *If all edge capacities are at least $(\varepsilon \log n + 1 + \varepsilon)$, there is a deterministic on-line* MDPP *algorithm that is $O(2^{1/\varepsilon} \log n)$-competitive against the fractional optimum.*

Proof. We follow the AAP algorithm and its analysis very closely. We vary a little from their notation, since we only deal here with routing a maximal number of requests, each of infinite duration. Thus, the i^{th} request is specified by a pair (s_i, t_i) of terminals. We define the "profit" of the connection to be n; thus the total profit obtained by the on-line algorithm is simply n times the number of terminal pairs routed.

Define $\mu = 2^{1+1/\varepsilon} n$, so we have

$$\varepsilon \log \mu = \varepsilon \log n + 1 + \varepsilon.$$

Let u_e denote the capacity of edge e; thus we can assume that for all e,

$$u_e \geq \varepsilon \log \mu.$$

With this value of μ, we now run the AAP algorithm — for the sake of completeness, we state this algorithm here.

For $j = 1, 2, \ldots, k$:
  Define λ_e^j to be the fraction of u_e consumed by paths already routed.
  Define $c_e^j = u_e(\mu^{\lambda_e^j} - 1)$.
  For a request (s_i, t_i), route it on any path P satisfying $\sum_{e \in P} \frac{1}{u_e} c_e^j \leq n$.
  If no such path is available, then reject the request.

First we argue why the relative load on an edge will never exceed 1. At the moment before this happened, on edge e say, we had

$$\lambda_e^j > 1 - \frac{1}{u_e} \geq 1 - \frac{1}{\varepsilon \log \mu}.$$

Thus

$$\begin{aligned} \frac{c_e^j}{u_e} &= \mu^{\lambda_e^j} - 1 \\ &> \mu^{1-\frac{1}{\varepsilon \log \mu}} - 1 \\ &= \frac{\mu}{2^{1/\varepsilon}} - 1 = 2n - 1 \\ &\geq n. \end{aligned}$$

So we have

$$\frac{c_e^j}{u_e} > n$$

and thus the connection could not have used this edge.

Suppose there are a total of k requests. Let A denote the set of requests routed by the AAP algorithm. Then we show

$$2^{1+1/\varepsilon} \log \mu \sum_{j \in A} n \geq \sum_e c_e^{k+1}. \tag{4}$$

As in the proof in [3] we show this by induction on the number of admitted requests, via proving that if the algorithm admits the j^{th} request, we have

$$\sum_e c_e^{j+1} - c_e^j \leq 2^{1+1/\varepsilon} n \log \mu.$$

So consider edge e on the j^{th} path used by the AAP algorithm. We have

$$c_e^{j+1} - c_e^j = u_e \left(\mu^{\lambda_e^j} (2^{\log \mu / u_e} - 1) \right).$$

Now the exponent on the 2 is at most $1/\varepsilon$, and for $x \in [0, 1/\varepsilon]$ we clearly have $2^x - 1 \leq 2^{1/\varepsilon} \cdot x$. Thus

$$\begin{aligned} c_e^{j+1} - c_e^j &\leq u_e \cdot \mu^{\lambda_e^j} \cdot 2^{1/\varepsilon} \cdot \log \mu / u_e \\ &= \mu^{\lambda_e^j} \cdot 2^{1/\varepsilon} \cdot \log \mu \\ &= 2^{1/\varepsilon} \cdot \log \mu \cdot \left[\frac{c_e^j}{u_e} + 1 \right]. \end{aligned}$$

Summing over all edges gives the desired bound.

Finally, we show that the expression

$$\sum_e c_e^{k+1} \tag{5}$$

is an upper bound on the profit of the *fractional* optimum minus the profit of the on-line algorithm. ([3] shows this for the integer optimum, but the proof is essentially the same.)

Let $\mathcal{Q}$ denote the set of indices which were rejected by the on-line algorithm but for which a positive fraction of the demand was routed by the optimum. For $j \in \mathcal{Q}$, suppose that the fractional optimum uses paths $P_j^1, \ldots, P_j^z$, with weights $\gamma_j^1, \ldots, \gamma_j^z$. Then since j was rejected by the on-line algorithm, and the costs are monotonic in the indices, we must have

$$n \leq \sum_{e \in P_j^i} \frac{c_e^{k+1}}{u_e}$$

for any i, j. Then for any edge e we have

$$\sum_{i,j: e \in P_j^i} \frac{\gamma_j^i}{u_e} \leq 1,$$

and hence we have

$$\begin{aligned} \sum_j \sum_i \gamma_j^i n &\leq \sum_j \sum_i \sum_{e \in P_j^i} \frac{\gamma_j^i c_e^{k+1}}{u_e} \\ &\leq \sum_e c_e^{k+1} \cdot \sum_{i,j: e \in P_j^i} \frac{\gamma_j^i}{u_e} \\ &\leq \sum_e c_e^{k+1}. \end{aligned}$$

Combining the bounds in Equations (4) and (5), we obtain the claimed competitive ratio.
■

A lower bound of [3] implies that the factor of $2^{1/\varepsilon}$ is unavoidable for deterministic on-line algorithms.

5.3 Routing Long Connections

Finally, we give the algorithm for routing long connections. First, we only consider terminal pairs with both ends in sets of the form C_v — denote this set of terminal pairs by $\mathcal{T}_M$. If $\mathcal{T}^*$ denotes a realizable subset of maximum size, then by Lemma 4.1, the expected number of terminal pairs in $\mathcal{T}^*$ that belong to $\mathcal{T}_M$ is a constant fraction of $|\mathcal{T}^*|$. Thus we only lose a constant factor in the competitive ratio by restricting attention to $\mathcal{T}_M$.

Set $\lambda_6' = \lambda_6(1 - \frac{1}{\log n})$ (the reason for this definition will become clear below); we define an on-line routing problem in the simulated network $\mathcal{N}(\lambda_6' \log n)$. If $s_i \in C_v$, then we define its image in the "simulation" to be $\psi(s_i) = v$. The input will simply be the sequence of terminal pairs $(\psi(s_i), \psi(t_i))$, where (s_i, t_i) is the sequence of pairs presented to the algorithm running on G. Our algorithm for the problem in the simulated network is as follow: we route (v, w) if (i) the AAP algorithm on $\mathcal{N}(\lambda_6' \log n)$ accepts (v, w), and (ii) no connection with an end equal to either v or w has yet been accepted.

Lemma 5.4 *The above algorithm is $O(\log n)$-competitive against the fractional optimum in $\mathcal{N}(\lambda_6' \log n)$.*

Proof. Let X denote the set of connections accepted by the on-line algorithm, and let γ^i denote the fraction of connection i routed by the fractional optimum. Let Y_1 denote the connections rejected because of rule (i), and Y_2 the connections rejected because of rule (ii). Then running the AAP algorithm by itself on the subsequence of the input consisting of X and Y_1, we see that AAP would still reject all connections in Y_1; hence $\sum_{i\in Y_1} \gamma^i \le O(\log n)|X|$. Now consider running rule (ii) by itself on the subsequence of the input consisting of X and Y_2; rule (ii) would still reject all the connections in Y_2. The set of pairs $(\psi(s_i), \psi(t_i))$ in $X \cup Y_2$ can be viewed as the edges of a graph on M; rule (ii) is running the greedy algorithm for building a maximal matching on this graph, and we know that optimum routing can put a fractional weight of at most $O(\log n)$ on the subset of edges of this graph incident to any one vertex in M. Thus $\sum_{i\in Y_2} \gamma^i \le O(\log n)|X|$. Since $\sum_{i\in Y_1\cup Y_2} \gamma^i \le \sum_{i\in Y_1} \gamma^i + \sum_{i\in Y_2} \gamma^i$, the bound follows. ■

Our on-line algorithm in G simply runs the above simulation; whenever $(\psi(s_i), \psi(t_i))$ is accepted, it routes the pair (s_i, t_i) in G using the paths constructed in Lemma 4.6 and Theorem 4.13. The following lemma says that it will not run out of "bandwidth" while doing this.

Lemma 5.5 *The algorithm in G can route all the connections accepted by the simulation.*

Proof. For each cluster C_v, we explicitly reserve the path of the form $\bar{Z}_{uu'}$ that passes through D_v closest to C_v; let us denote this path by $\bar{Z}^*_v$. Now when the simulation accepts $(\psi(s_i), \psi(t_i))$, it specifies a sequence of neighboring clusters $C_{v_1}, C_{v_2}, \ldots, C_{v_r}$, where $v_1 = \psi(s_i)$ and $v_r = \psi(t_i)$.

The algorithm in G routes (s_i, t_i) as follows. First it routes s_i and t_i out to their reserved paths $\bar{Z}^*_{\psi(s_i)}$ and $\bar{Z}^*_{\psi(t_i)}$. Then for each $j = 1, \ldots, r-1$, it chooses any path $\bar{Z}_{uu'}$ ($u \in \tau_{v_j}$ and $u' \in \tau_{v_{j+1}}$) that is not one of the special reserved paths and that has not yet been used for a previous terminal pair. In this way it obtains a sequence of such paths $Z_1, \ldots, Z_{r-1}$. Each path crosses its successor at some vertex; thus they can be joined together to produce a path from s_i to t_i. Since the simulation only accepts at most $\lambda_6' \log n$ terminal pairs whose routes use the edge in $\mathcal{N}$ from v to w, for any $v, w \in M$, there are enough inter-cluster paths to route all accepted terminal pairs. ■

Finally, we have to show that optimum in G is not far from the optimum in the simulation. This follows using Lemma 4.12. Since the on-line algorithm is $O(\log n)$-competitive against the fractional optimum in $\mathcal{N}(\lambda_6' \log n)$, it is also $O(\log n)$-competitive against the fractional optimum in $\mathcal{N}(\gamma \log n)$, which by Lemma 4.12 is at least as large as the maximum realizable subset of $\mathcal{T}$. Thus,

Theorem 5.6 *The on-line algorithm is $O(\log n)$-competitive in any uniformly densely embedded Eulerian graph G.*

5.4 A Digression: Combining Maximization Algorithms

The following section is fairly elementary, and not needed in the rest of the paper. However, we feel it is worthwhile making more explicit the "combining" of on-line algorithms that is being used in the proof of Lemma 5.4.

Let U denote a finite set, with $S_1, \ldots, S_n$ subsets of U such that $U = \cup_i S_i$. Let d be the least number such that for each $u \in U$, we have

$$|\{i : u \in S_i\}| \leq d. \tag{6}$$

Let $\mathcal{F}_i$ denote a collection of subsets of S_i closed with respect to inclusion. For $U' \subseteq S_i$, define

$$\mu_i(U') = \max\{|C| : C \subseteq U' \text{ and } C \in \mathcal{F}_i\}.$$

Now, on-line algorithm A_i ($i = 1, \ldots, n$) is trying to find a "large" set in $\mathcal{F}_i$; it operates as follows. The elements of some $U' \subseteq S_i$ are presented to it any order, and on each element the algorithm A_i either accepts it or rejects it. Moreover, we assume that the state of A_i is completely determined by the set of elements it has accepted so far.

We say A_i is c-competitive if on any ordered $U' \subset S_i$, A_i returns a set T_i of size at least $\frac{1}{c} \cdot \mu_i(U')$. Let c^* denote the maximum competitive ratio of any of the algorithms A_i.

Now define

$$\mathcal{F} = \{C : \forall i \; C \cap S_i \in \mathcal{F}_i\}$$

and $\mu(U')$ to be the maximum size of a member of $\mathcal{F}$ contained in U'. We define the combined algorithm $A = \wedge_{i=1}^n A_i$ as follows. As each $u \in U'$ is presented to A, it accepts u iff for each i such that $u \in S_i$, A_i accepts u. The total set accepted so far, intersected with S_i, serves as the state for each A_i.

Proposition 5.7 *A is c^*d-competitive.*

Proof. Assume the algorithm A was presented with a set U' and it returned X. Let Y denote a member of $\mathcal{F}$ contained in U' of maximum size; we show that $|Y| \leq c^* d|X|$. Let R_i' denote the elements of $Y - X$ that were rejected by algorithm A_i, $J_i = X \cap Y \cap S_i$, and $R_i = J_i \cup R_i'$. Note that $Y = \cup_i R_i$.

Now set $U_i' = (X \cap S_i) \cup R_i$. Order U_i' as it appears in U', and present it as input to A_i. Then as in the running of the combined algorithm A, A_i will accept precisely the set $X \cap S_i$. Since A_i is c^*-competitive, and $R_i \subset Y \cap S_i \in \mathcal{F}_i$, we have

$$|R_i| \leq c^* |X \cap S_i|.$$

We also have $|Y| \leq \sum_i |R_i|$, and by Inequality (6) we have $\sum_i |X \cap S_i| \leq d|X|$. Thus

$$|Y| \leq \sum_i |R_i| \leq c^* \sum_i |X \cap S_i| \leq c^* d|X|.$$

■

In the natural way, one can define a fractional version of the above model. Instead of a collection $\mathcal{F}_i$ of "allowable subsets" of S_i, we are now given an anti-blocking polytope $\mathcal{P}_i \subset [0,1]^{|S_i|}$ (if $y \in \mathcal{P}_i$ and $x \leq y$ coordinate-wise, then $x \in \mathcal{P}_i$). For $U' \subset S_i$, the function $\mu_i(U')$ now gives the maximum fractional weight of a subset of U', where the maximum is taken over the polytope $\mathcal{P}_i$.

Then the proof just given carries over to show

Proposition 5.8 *If each A_i is c^*-competitive against the fractional optimum, then A is c^*d-competitive against the fractional optimum.*

The application to routing in the simulated network is clear. Algorithm A_0 is the AAP algorithm running in $\mathcal{N}(\lambda_6' \log n)$, and for each $v \in M$, algorithm A_v is the algorithm that allows one request out of C_v and then rejects all subsequent requests. By the previous section, our on-line algorithm in $\mathcal{N}(\lambda_6' \log n)$ is simply the combined algorithm $A_0 \wedge (\wedge_{v \in M} A_v)$, $c^* = O(\log n)$, and $d = 3$. Thus the combined algorithm is $O(\log n)$-competitive.

6 The Off-Line Algorithm

For the constant-factor off-line approximation, we use the same graph $\mathcal{N}$ as before. In $\mathcal{N}(\gamma \log n)$, for any fixed constant γ, one can obtain a constant-factor approximation to the MDPP by the following *randomized rounding* algorithm of Raghavan [20]. First we solve the fractional relaxation of the MDPP instance (this can be done in polynomial time); from this, we obtain for each terminal pair (s_i, t_i) a collection of paths $P_i^1, \ldots, P_i^z$ and associated weights $y_i^1, \ldots, y_i^z \in [0,1]$ such that $x_i = \sum_j y_i^j \in [0,1]$. We now pick a *scaling factor* $\mu < 1$; independently for each terminal pair (s_i, t_i) we route it on path P_i^j with probability μy_i^j, and don't route it at all with probability $1 - \mu x_i$. If we do route it, we say that (s_i, t_i) has been *rounded up*. In [21, 20], it is shown that with constant probability, no capacity is violated by the selected paths, and the number of pairs that are rounded up is a constant fraction of the fractional optimum.

In particular, we require the following theorem from [20].

Theorem 6.1 (Raghavan) *Let $X_1, X_2, \ldots, X_r$ be independent Bernoulli trials with $EX_j = p_j$ and $\Psi = \sum_i X_i$; so $E\Psi = m = \sum_i p_i$. Then for $\delta > 0$ we have*

$$Pr[\Psi > (1+\delta)m] < \left[\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right]^m.$$

We specialize this to the form in which we will use it as follows.

Corollary 6.2 *Let $0 < \mu < 1$, and $p_1, \ldots, p_r \in [0,1]$. Let $X_1', X_2', \ldots, X_r'$ be independent Bernoulli trials with $EX_j' = \mu p_j$ and $\Psi' = \sum_i X_i'$. Let $m = \sum_i p_i$. Then*

$$Pr[\Psi' > m] < (e\mu)^m.$$

Proof. Apply the bound of Theorem 6.1 with m set to μm and δ set to $\mu^{-1} - 1$. ∎

Let us consider how to use this randomized rounding approach in routing long connections off-line. In the high-capacity network $\mathcal{N}$, this rounding approach is fine; but to get a constant-factor approximation we also have to be within a constant factor of the optimum in routing terminals out of the clusters (in the on-line algorithm it was enough to route only one). To this end, we build the following more complicated network $\mathcal{N}'$. Let z_v denote the node representing $v \in M$ in the network $\mathcal{N}$; we construct $\mathcal{N}'$ by attaching C_v to z_v via an edge from z_v to *each* node in $\pi(C_v)$. Let $\mathcal{N}'(\gamma)$ denote the network $\mathcal{N}'$ in which each edge *between nodes of the subgraph* $\mathcal{N}$ has capacity γ, and all other edges have unit capacity.

We now run the randomized rounding algorithm on $\mathcal{N}'(\lambda_7 \log n)$, for a small constant $\lambda_7 > 0$. With high probability the parts of all selected paths lying in the subgraph $\mathcal{N}(\lambda_7 \log n)$, taken together, do not violate any capacity constraint; and the number of pairs that are rounded up is within a constant factor of the fractional optimum. We now must convert the selected paths in $\mathcal{N}'(\lambda_7 \log n)$ into s_i-t_i paths in G. We can use the technique of the previous section to produce, for each selected pair (s_i, t_i), a "global" path P_i that begins at $\tau'_{\psi(s_i)} \subset \pi(D_{\psi(s_i)})$ and ends at $\tau'_{\psi(t_i)} \subset \pi(D_{\psi(t_i)})$.

The real problem is how to find paths *within* the clusters such that each s_i (resp. t_i) that has been rounded up can reach the endpoint of this associated global path on $\tau'_{\psi(s_i)}$ (resp. $\tau'_{\psi(t_i)}$). For this, the paths returned by the randomized rounding are of no value, since the edges of $\mathcal{N}'$ within the clusters C_v have only unit capacity. Instead we argue as follows.

Let S_v denote the set of terminals in C_v that are rounded up. Each is trying to "get out" to its associated path that begins at τ'_v. Recall that $G_v = G[D_v - C_v^o]$ has outer facial cycle φ_v, and a set of vertices τ'_v spaced κ apart on φ_v, each with an associated "antipodal" point on φ_v.

For a constant κ_1, we choose a set σ'_v of vertices spaced κ_1 apart on the outer facial cycle Q'_v of $G[C_v]$. First let us argue that it is enough to route the terminals in S_v to σ'_v — for then in G_v, we can construct paths linking the vertices in σ'_v to the endpoints of the global paths in τ'_v, and build a crossbar through G_v in what remains. Specifically,

Lemma 6.3 *There is a crossbar in G_v anchored in $\sigma'_v \cup \tau'_v$, such that each vertex of $\sigma'_v \cup \tau'_v$ is the endpoint of exactly one path in the crossbar.*

Proof. This will be another application of Frank's extension of the Okamura–Seymour theorem. First we "cut open" the graph G_v along a shortest path P^* from φ_v to Q'_v in the planar dual. Now all terminals in $\sigma'_v \cup \tau'_v$ lie on a single face of the resulting graph. We now partition $\sigma'_v \cup \tau'_v$ into antipodal pairs, with respect to the ordering induced by the facial cycle containing them. Finally, we set up a disjoint paths problem with the set of antipodal pairs as terminals in this "opened" copy of G_v.

Observe that the capacity of any cut in the opened copy of G_v is at least half as large as a cut separating the same set of terminals in G_v — we can simply convert it to a cut in G_v by adjoining the edges removed from P^*. Given this observation and the arguments of Lemma 4.6, the verification of the strict cut condition is routine. ∎

Since the simulated network now requires only capacity $\lambda_7 \log n$, this crossbar can used to set up paths linking σ'_v to the endpoints of all global paths in τ'_v, while still preserving enough paths for the simulated network to use for routing.

So this leaves us with the following *escape problem.* We are given the set S_v of terminals that have been rounded up, and we want to route a large fraction of them to $\sigma'_v \subset \pi(C_v)$ via disjoint paths. The following lemma, whose proof contains the central step of the algorithm, says that this can be done. For simplicity we will prove the lemma for the special case of the mesh, and then describe the extension to our more general class of graphs.

Lemma 6.4 *For a sufficiently small (constant) value of μ, there is a constant $c < 1$ and sets $S'_v \subset S_v$, such that*

(i) if one end of a pair (s_i, t_i) belongs to $\cup_v S'_v$ then so does the other,

(ii) $|\cup_v S'_v| \geq c|\cup_v S_v|$, and

(iii) each set S'_v can be linked to $\sigma'_v \subset \pi(C_v)$ via edge-disjoint paths.

Proof. We first prove this fact assuming that G is the two-dimensional mesh, each C_v is a square submesh, and a terminal can "escape" to any vertex on the boundary of C_v. First observe the following fact: an escape problem on a rectangular mesh is feasible if and only if, for all p, q, any subrectangle of size $p \times q$ contains at most $2(p+q)$ terminals. To see this, note that we are dealing with a maximum flow problem, and thus only have to verify the cut condition. On a rectangular mesh, the smallest rectangle enclosing any connected cut has no greater capacity, and contains at least as many terminals, as the original cut; thus the cut condition holds if and only if it holds for all subrectangles.

Call a rectangle *overfull* if it violates the cut condition. What is the probability that a $p \times q$ rectangle becomes overfull after the rounding? Before rounding, the total fractional weight it contains is at most $2\mu(p+q)$ (since the un-scaled fractional flow is feasible). Thus, setting $\gamma = (e\mu)^2$, the probability that the number of terminals exceeds $2(p+q)$ after rounding is at most γ^{p+q}, by Corollary 6.2.

This suggests the following algorithm to construct the set S'_v: we go through each $s \in S_v$, deleting it if it is contained in any overfull rectangle — we also then delete its matching terminal in some other cluster. This results in the set S'_v. What is the probability that s is contained in an overfull rectangle? s is contained in pq rectangles of dimensions $p \times q$, so the probability is clearly bounded by the infinite sum

$$\sum_p \sum_q pq\gamma^{p+q} = \frac{\gamma^2}{(1-\gamma)^4}.$$

s can also be deleted if its matching terminal is contained in an overfull rectangle, so the probability of s being deleted is at most $\frac{2\gamma^2}{(1-\gamma)^4}$. By taking μ small enough, we can make this last expression a constant less than 1; this implies the lemma for the two-dimensional mesh.

We now describe the extension to uniformly densely embedded graphs in general. We now have no way to define rectangles *per se*, but we define a *round cut* to be a set of the form $B_r(u) \cap C_v$. Since by Lemma 3.4 a given $u \in C_v$ is only contained in $O(r^2)$ round cuts of radius r, the following claim shows that the argument of the previous paragraph can be applied. ■

Claim 6.5 *There is a constant ξ_1 such that the following holds. If the capacity of every round cut exceeds the number of terminals it contains by at least a factor of ξ_1, then the escape problem in C_v (with respect to σ'_v) is feasible.*

Proof. Suppose that the escape problem is not feasible; then there is *some* cut U such that the number of terminals in U exceeds $|\delta(U)|$. Moreover, we can find such a U with both $G[U]$ and $G[C_v - U]$ connected.

Set $\xi'_1 = \kappa_1 + \bar{\alpha}^{-1/2}\alpha^{-1}$, $\xi_1 = 4\Delta\xi'_1(\beta + \varepsilon^{-1})$, and write $p = |\delta(U)|$. We will be done if we can exhibit a round cut R containing U for which $|\delta(R)| \leq \xi_1 p$.

Now if we contract $G - U$ to a single vertex, we obtain a planar graph with maximum face size κ_1 (as opposed to ℓ; this is due to the large spacing of the vertices of σ'_v). So by Lemma 3.7, the maximum distance between two points on $\pi(U)$ is at most $\kappa_1 p$.

Next we claim that every vertex in U must be within distance $\bar{\alpha}^{-1/2}\alpha^{-1}p$ of $\pi(U)$. The reason for this is analogous to the proof of Lemma 3.6 — if not, then U would contain a ball of more than this radius, which would contain more than $\alpha^{-1}p^2$ vertices; a contradiction since $|\delta(U)| = p$.

Thus for any $u \in \pi(U)$, we have $U \subset B_r(u)$, where

$$r = (\kappa_1 + \bar{\alpha}^{-1/2}\alpha^{-1})p = \xi'_1 p.$$

Now by Lemma 3.5, there is an r' between r and $2r$ such that

$$|\delta(B_{r'}(u))| \leq 4\beta\Delta\xi'_1 p.$$

Now let $R \supset U$ denote the round cut $B_{r'}(u) \cap C_v$. Every edge of $\delta(R)$ is an edge of either $\delta(B_{r'}(u))$ or of $\delta(C_v \cap R)$. The former quantity was just shown to be at most $4\beta\Delta\xi'_1 p$. To bound the latter quantity, note that any two vertices in $\pi(C_v \cap R)$ are at most $2r' \leq 4\xi'_1 p$ apart; since the facial cycle containing $\pi(C_v)$ is ε-smooth, this means that $\pi(C_v \cap R)$ contains at most $4\varepsilon^{-1}\xi'_1 p$ vertices, and hence

$$|\delta(C_v \cap R)| \leq 4\Delta\varepsilon^{-1}\xi'_1 p.$$

The claim now follows since

$$|\delta(R)| \leq 4\Delta\xi'_1(\beta + \varepsilon^{-1})p = \xi_1 p.$$

■

This gives a constant-factor approximation for long connections: if (s_i, t_i) is rounded up, and $s_i, t_i \in \cup_v S'_v$, then we concatenate the paths from s_i to $\pi(C_{\psi(s_i)})$ (given by Lemma 6.4) to $\pi(D_{\psi(s_i)})$ (given by Lemma 6.3) to $\pi(D_{\psi(t_i)})$ (given by the path in $\mathcal{N}(\lambda_7 \log n)$), and now symmetrically to $\pi(C_{\psi(t_i)})$ and to t_i.

We handle short connections recursively as follows. We run the above algorithm independently on disjoint neighborhoods of each cluster C_v, as provided by Lemma 5.2. Recall from the proof of Lemma 5.2 that such neighborhoods consisted of sets $U''_v \supset U'_v \supset C_v$, such that

$U_v'' - U_v'$ contained a crossbar for routing a constant fraction of all connections whose paths left U_v'.

Now there is the following problem: $G[U_v'']$, on which we run the algorithm recursively, contains a large face. To take care of this, we can either invoke the more general algorithm of the following section, or we can use the following trick. Let Q denote the outer facial cycle of U_v'; suppose that it contains p vertices of even degree in $G[U_v']$ and q vertices of odd degree. Let H denote a two-dimensional mesh whose side lengths differ by at most 1 and whose outer face has length $2p + q + 4$; we attach H to $G[U_v']$ by edges from the degree-3 vertices on the outer face of H to the vertices of Q. In order to make sure that the resulting graph is Eulerian, we attach each even-degree vertex of Q to two such vertices on the outside of H, and each odd-degree vertex of Q to one such vertex on the outside of H. In this way, we obtain an Eulerian graph that is uniformly densely embedded — we now run the above algorithm on this graph, obtaining a collection of edge-disjoint paths. Paths that do not leave U_v' can be used directly in G; and we use the crossbar in $U_v'' - U_v'$ to route a constant fraction of the connections whose paths use the mesh H.

Call a connection "medium" if it is now a long connection in this recursive call, and "small" otherwise. Medium connections are handled as just described in the previous paragraph. Small connections take place within clusters of size $O(\log \log n)$ and therefore can be simply solved to optimality by brute force (again on disjoint neighborhoods provided by Lemma 5.2). We can then take the largest realizable set we find among the long, medium, and small connections, obtaining

Theorem 6.6 *There is a constant-factor (off-line)* MDPP *approximation in uniformly densely embedded Eulerian graphs.*

7 Graphs with an Exceptional Face

In this section, we sketch the extension of our algorithms to densely embedded, nearly-Eulerian graphs. Recall from Definition 3.3 that such a graph satisfies the properties of a uniformly densely embedded Eulerian graph, except that it is allowed to contain an "exceptional" face Φ^*, with facial cycle Q^* that may have length greater than ℓ and may contain vertices not of even degree.

For the remainder of this section, let G denote a densely embedded, nearly-Eulerian graph with parameters α, λ, Δ, and ℓ. For simplicity, we assume that the facial cycle Q^* is sufficiently large that it is not contained in any set $B_{\lambda \log n}(v)$; it is not difficult to remove this assumption.

The changes required in the algorithm come from the fact that there can now be a G-normal curve joining two distant vertices in G that intersects G relatively few times — this is because it can pass through the large face Φ^*. This has consequences in the proofs of Lemma 4.6 (and its relatives) and Claim 4.14. However, by requiring the outer cycles of clusters and enclosures to satisfy a more restrictive notion of ε-smoothness, these facts will follow as before.

We define our more restrictive type of ε-smoothness as follows. Let G/Q^* denote the graph G with a single additional node q^* joined by length-0 edges to each vertex of the long facial cycle Q^*. Then a small cut passing through two distant vertices, as described in the previous paragraph, *does* correspond to a short path in G/Q^* — it simply makes use of the additional node q^*. Now it is straightforward to show that we need only require the outer cycles of the clusters and enclosures to be ε-smooth in the graph G/Q^*; and this can be accomplished by running the ε-smoothing algorithm in this graph instead of in G.

This introduces a further difficulty, however. Say that we have just smoothed some cycle Q, obtaining a cycle Q'. While Q' will be ε-close to the original cycle Q in G/Q^*, there is no reason why this means it will be ε-close in G.

To handle this, we strengthen the statement of Theorem 4.4, as follows. For a vertex u, define the *u-restricted distance* $d^u(v, w)$ between two vertices v and w to be the minimum length of a v-w path avoiding u. From the proof of Theorem 4.4, one sees that we can always find a short path from the smooth cycle Q' back to the original cycle Q that avoids any prescribed vertex; i.e. Q' is ε-close to Q with respect to any u-restricted distance function. In particular, Q' is ε-close to Q in G/Q^* with respect to the q^*-restricted distance function; that is, Q' is ε-close to Q in the original graph G.

Thus, we can obtain clusters and enclosures with boundaries which are ε-smooth in G/Q^*, and which are not far from the original boundaries in G. The sets τ'_v will now consist of evenly spaced vertices only on the part of an enclosure's outer facial cycle that does not lie on Φ^*. The proof of Lemma 4.6 now follows exactly as before. When we use Schrijver's theorem to construct inter-cluster paths, we also cut the surface Σ along the long facial cycle Q^* so as to remove Φ^* from the surface. Now an essential curve can be anchored on the boundary of Φ^* as well as on the boundary of an enclosure; but this poses no problem since the enclosure boundaries are ε-smooth in G/Q^*.

Once the simulated network $\mathcal{N}$ has been set up, the on-line and off-line algorithms work exactly as before. (In particular, Claim 6.5 follows without modification, since the large face Φ^* is incorporated into the hypotheses of Lemma 3.7.) Thus we have,

Theorem 7.1 *There is an $O(\log n)$-competitive on-line* MDPP *approximation in any densely embedded nearly-Eulerian graph.*

Theorem 7.2 *There is a constant-factor (off-line)* MDPP *approximation in any densely embedded nearly-Eulerian graph.*

Acknowledgements

We thank Yuval Rabani for his contribution to this work. In particular, he proposed the general technique of performing randomized rounding on graphs with low-capacity edges and throwing out parts of the routing on which the capacity constraints become violated. An unpublished algorithm of Satish Rao and the second author [23] provided one of the initial motivations for this approach, suggesting the technique of contracting $O(\log n) \times O(\log n)$ subsquares of the two-dimensional mesh to simulate a high-capacity network. Thanks also to Baruch Awerbuch, Michel Goemans, David Karger, Tom Leighton, and Lex Schrijver for their comments on these problems.

References

[1] A. Aggarwal, A. Bar-Noy, D. Coppersmith, R. Ramaswami, B. Schieber, M. Sudan, "Efficient routing and scheduling algorithms for optical networks," *Proc. 5th ACM-SIAM SODA*, 1994, pp. 412–423.

[2] Y. Aumann, Y. Rabani, "Improved bounds for all-optical routing," *Proc. 6th ACM-SIAM SODA*, 1995, pp. 567–576.

[3] B. Awerbuch, Y. Azar, S. Plotkin, "Throughput-Competitive Online Routing," *Proc. 34th IEEE FOCS*, 1993, pp. 32–40.

[4] B. Awerbuch, Y. Bartal, A. Fiat, A. Rosén, "Competitive non-preemptive call control," *Proc. 5th ACM-SIAM SODA*.

[5] 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 FOCS*, 1994, pp. 412–423.

[6] A. Broder, A. Frieze, S. Suen, E. Upfal, "Optimal construction of edge-disjoint paths in random graphs," *Proc. 5th ACM-SIAM SODA*, 1994, pp. 603–612.

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

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

[9] 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.

[10] 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. ICALP*, 1993, pp. 64–75.

[11] R.M. Karp, "Reducibility among combinatorial problems," *Complexity of Computer Computations*, R.E. Miller, J.W. Thatcher, Eds., New York: Plenum Press, 1972. pp. 85–103.

[12] J. Kleinberg, É. Tardos, "Approximations for the disjoint paths problem in high-diameter planar networks," *Proc. 27th ACM STOC*, 1995.

[13] M.E. Kramer, J. van Leeuwen, "The complexity of wire routing and finding the minimum area layouts for arbitrary VLSI circuits," *Advances in Computing Research 2: VLSI Theory*, F.P. Preparata, Ed., London: JAI Press, 1984. pp. 129–146.

[14] R. Lipton, R. Tarjan, "A separator theorem for planar graphs," *SIAM J. Applied Math.* 36(1979), pp. 177–189.

[15] M. Luby, "A simple parallel algorithm for the maximal independent set problem," *SIAM J. Computing*, 15(1986), pp. 1036–1053.

[16] W. Massey, *Algebraic Topology: An Introduction*, New York: Springer-Verlag, 1967.

[17] M. Middendorf, F. Pfeiffer, "On the complexity of the disjoint paths problem," *Combina-*

torica, 13(1993). pp. 97–107.

[18] H. Okamura, P. Seymour, "Multicommodity flows in planar graphs," *J. Comb. Theory Ser. B*, 31(1981), pp. 75–81.

[19] D. Peleg, E. Upfal, "Disjoint paths on expander graphs," *Combinatorica*, 9(1989), pp. 289–313.

[20] P. Raghavan, "Probabilistic construction of deterministic algorithms: approximating packing integer programs," *J. Computer and System Sciences*, 37(1988), pp. 130–143.

[21] P. Raghavan, C.D. Thompson, "Randomized rounding: a technique for provably good algorithms and algorithmic proofs," *Combinatorica*, 7(1987), pp. 365–374.

[22] P. Raghavan, E. Upfal, "Efficient all-optical routing," *Proc. 26th ACM STOC*, 1994, pp. 134–143.

[23] S. Rao, É. Tardos, personal communication, 1993.

[24] N. Robertson, P.D. Seymour, "Graph Minors VII: Disjoint paths on a surface," *J. Combinatorial Theory Ser. B*, 45(1988), pp. 212–254.

[25] N. Robertson, P.D. Seymour, "Graph Minors XI: Circuits on a surface," *J. Combinatorial Theory Ser. B*, 60(1994), pp. 72–106.

[26] N. Robertson, P.D. Seymour, "Graph Minors XIII: The disjoint paths problem," *J. Combinatorial Theory Ser. B*, 63(1995), pp. 65–110.

[27] A. Schrijver, "Disjoint circuits of prescribed homotopies in a graph on a compact surface," *J. Combinatorial Theory Ser. B*, 51(1991), pp. 127–159.

[28] D. Sleator, R. Tarjan, "Amortized efficiency of list update and paging rules," *Comm. ACM*, 23(1985), pp. 202–208.

[29] D. Wagner, K. Weihe, "A linear time algorithm for multicommodity flow in planar graphs," *Proc. First European Symposium on Algorithms*, 1993, pp. 384–395.