Submitted to SIAM Journal on Computing. Preliminary version appeared in STOC '96.

Node-Disjoint Paths on the Mesh and a New Trade–Off in VLSI Layout

Alok Aggarwal* **Jon Kleinberg†** **David P. Williamson***

Abstract

A number of basic models for VLSI layout are based on the construction of node-disjoint paths between terminals on a multi-layer grid. In this setting, one is interested in minimizing both the number of *layers* required and the *area* of the underlying grid. Building on work of Cutler and Shiloach, and Aggarwal, Klawe, et al., we prove an upper-bound trade-off between these two quantities in a general multi-layer grid model. As a special case of our main result, we obtain significantly improved bounds for the problem of routing a full permutation on the mesh using node-disjoint paths; our new bound here is within polylogarithmic factors of the bisection bound. Our algorithms involve some new techniques for analyzing the structure of node-disjoint paths in planar graphs, and indicate some respects in which this problem, at least in the planar case, is fundamentally different from its edge-disjoint counterpart.

1 Introduction

The basic node–disjoint paths problem is as follows. We are given a graph G, and pairs of vertices in G. The two natural objectives we consider are

(1) simultaneously route as many pairs as possible on node–disjoint paths in G.

(1′) route all pairs in as few "rounds of communication" as possible, where all paths routed in a single round must be node–disjoint.

Finding node-disjoint paths between designated pairs of vertices in a planar graph is fundamental to a number of routing and embedding problems. In particular, in many models of VLSI layout, effective layout strategies rely on heuristics for finding node–disjoint paths; many current routing software systems rely extensively on such an approach (see also [12]). Node–disjointness constraints can arise as well in the context of virtual circuit routing in networks, when one requires paths to be node–disjoint for the sake of fault–tolerance or because of bottlenecks involving switches rather than links.

*I.B.M. T.J. Watson Research Center, P.O. Box 218, Yorktown Heights NY 10598.

†Laboratory for Computer Science, MIT, Cambridge, MA 02139. Supported by an ONR Graduate Fellowship. Part of this work was performed while visiting the IBM T.J. Watson Research Center. Current address: IBM Almaden Research Center, San Jose CA 95032, on leave from Department of Computer Science, Cornell University, Ithaca NY 14853.

At the same time, the basic node–disjoint paths problem is NP-hard even for the two–dimensional mesh [9], and very little is known in the way of reasonable approximation algorithms for the problem. Given the basic nature of the problem, we are interested in looking for techniques with provable performance guarantees for constructing node–disjoint paths in planar graphs, as such techniques may help in answering questions of the form (1) and (1′) as they arise in practice.

The study of this problem is also interesting to us precisely because its structure, in the case of planar graphs, is very different from the analogous edge–disjoint paths problem. This is true even in a technical sense; for example, work of Éva Tardos and the second author [8] has shown that for many planar graphs including the mesh, a natural *cut condition* is sufficient to within polylogarithmic factors for the feasibility of edge–disjoint paths. But this is far from true in the node–disjoint case, where one encounters a variety of "topological" obstructions to the existence of the paths. See Section 1.1 for a discussion of some of these issues.

Let us discuss a very basic problem that arises in this setting. We say a set of terminal pairs in a graph is a *permutation* if each vertex of the graph appears in exactly one pair. If G denotes the $n \times n$ two–dimensional mesh graph, and we are given a permutation in which every pair needs to cross the center column, then the standard bisection bound says that at most n terminal pairs can be routed simultaneously, and so at least $\frac{1}{2}n$ rounds, in the sense of (1′), are required.

The lack of understanding of this area is such that the following has remained open. Again, let G denote the mesh.

> (2) For every permutation on G, does there exist a set of $\Omega(n^{1-\varepsilon})$ pairs that can be simultaneously routed on node–disjoint paths, for every $\varepsilon > 0$?
>
> (2′) Can every permutation on G be routed in $O(n^{1+\varepsilon})$ rounds, for every $\varepsilon > 0$?

Again, by way of contrast, these questions are straightforward to resolve in the affirmative when one only requires edge–disjoint paths.

These latter questions turn out to be closely related to open problems of Aggarwal, Klawe, Lichtenstein, Linial, and Wigderson [1] in the setting of *multi–layer* VLSI layout. In multi–layer VLSI layouts, typically a *pin* (a node, in our terminology) goes through several layers at the same location, and the routing among pins at each layer is planar. If a routing wire needs to change a layer, then a *via* (or *contact cut*) has to be made between two layers, and this usually affects the yield of the output chip (as well as affecting the general quality of the circuit). Consequently, it is practically useful to route as many connections on a single layer as possible (in order to reduce the vias); for more details see [1, 5, 14]. These constraints thus result directly in problems of the type (1) and (1′), with *layers* now playing the role of "rounds."

One typically reduces the number of layers by introducing spacing between the terminals. In the model of [1, 5], for a parameter d, consider a $dn \times dn$ mesh G_d, with an $n \times n$ grid of terminals at a *uniform spacing* of d. By this we mean that terminals are placed at the set of nodes of G_d whose row and column numbers are multiples of d; thus, $d-1$ paths can now be routed between adjacent terminals. The goal in this model is to find the tightest trade–off possible between the spacing d and the number of layers required in the worst case.

Cutler and Shiloach [5] showed that spacing $d = O(n^2)$ suffices to route all pairs in a single layer; and [1] showed that this value of d was tight up to constant factors. In terms of a general trade-off between d and the minimum number of layers required, the techniques of [5, 1] are only sufficient to show that, for spacing d, $O(\left(\frac{n}{\sqrt{d}}\right)^{4/3})$ layers are sufficient. However, guided by $(2')$, a reasonable question was the following.

> $(3')$ With spacing d, are $O(\left(\frac{n}{\sqrt{d}}\right)^{1+\varepsilon})$ layers sufficient to route any permutation on the mesh, for every $\varepsilon > 0$?

This contains the result of [5] in the case $d = \Theta(n^2)$ and question $(2')$ in the case $d = 1$.

The main results of this paper provide affirmative answers to questions (2), $(2')$, and $(3')$. Specifically, we prove

Theorem 1.1 *Every permutation on the mesh contains a set of $\Omega(n/\log n)$ terminal pairs that can be simultaneously routed on node–disjoint paths. Moreover, every permutation on the mesh can be routed in $O(n \log^2 n)$ rounds.*

Theorem 1.2 *With spacing d, every permutation on the mesh contains a set of $\Omega(n\sqrt{d}/\log n)$ pairs that can be routed in one layer, and the entire permutation can be routed in $O(nd^{-\frac{1}{2}} \log^2 n)$ layers.*

This strengthens the bounds of [5, 1] on the number of layers required for routing a permutation. Within the context of VLSI, where the number of layers is typically a fixed small number, the "dual" maximization versions of these theorems may in fact be of greater interest. Namely, these are the first algorithms that can provably extract from any permutation on the mesh a routable set of terminal pairs whose size is within polylogarithmic factors of the bisection bound. Moreover, the approach we develop is fundamentally different from that of Cutler and Shiloach [5], which was essentially the only previous method applicable in this setting. Given the basic importance of the problem, we feel that the introduction of new techniques for constructing node–disjoint paths with improved performance guarantees could be of value in suggesting additional heuristic approaches to these problems.

We mentioned above that questions (2) and $(2')$ are easy in the case of edge–disjoint paths; in fact, one can construct a set of paths meeting the bisection bound using only one bend in each path. On the other hand, the paths in the Cutler–Shiloach construction, as well as our constructions in Theorems 1.1 and 1.2, use many bends. Thus, it is natural to ask how many rounds are required with a routing that uses only a constant number of bends. Getting $o(n^2)$ rounds is not immediate, and it appears that no previous construction in the literature achieves it. We show

Theorem 1.3 *Every permutation on the mesh can be routed in $O(n^{7/5})$ rounds, with each path in the routing using only 6 bends.*

1.1 The Node–Disjoint Paths Problem

For the sake of concreteness, we give some definitions. Throughout this paper, let $G = (V, E)$ denote the $n \times n$ two–dimensional mesh. Let $\mathcal{T} = \{(s_1, t_1), \ldots, (s_m, t_m)\}$ denote a set of *terminal pairs*, where each $s_i, t_i \in V$. A *permutation* is a set $\mathcal{T}$ of terminal pairs that partitions the vertex set; a *partial permutation* is a set of terminal pairs in which each node appears at most once. These are the only types of sets $\mathcal{T}$ we will be considering.

We say that a subset $\mathcal{T}'$ of $\mathcal{T}$ is *realizable* in G if all pairs in $\mathcal{T}'$ can be simultaneously connected by node-disjoint paths, and and define $\chi(\mathcal{T})$ to be the minimum size of a partition of $\mathcal{T}$ into realizable sets. This corresponds to the minimum number of rounds of communication — or the minimum number of layers — required, in the sense of (1′).

In this language, Theorem 1.1 simply says that for any permutation $\mathcal{T}$, we have $\chi(\mathcal{T}) = O(n \log^2 n)$. For the model of [5, 1], with a $dn \times dn$ mesh and terminals evenly spaced at distance d, we use $\chi_d(\mathcal{T})$ to denote the minimum size of a partition of $\mathcal{T}$ into routable sets. Thus, our second result is that $\chi_d(\mathcal{T}) = O(nd^{-\frac{1}{2}} \log^2 n)$ for any permutation $\mathcal{T}$.

In addition to connections with VLSI and network routing, we mention that in the context of edge–disjoint paths the quantity $\chi(\mathcal{T})$ can also be viewed as the minimum number of *wavelengths* required to route $\mathcal{T}$ under an optical routing model in which two paths can share an edge as long as they are assigned different wavelengths. Bounds for $\chi(\mathcal{T})$ for permutations $\mathcal{T}$ on a d-dimensional mesh is a result of Raghavan and Upfal [13], and our work here can be seen to provide node–disjoint analogues of some of these bounds. However optical routing is not, strictly speaking, a motivation of this work, as we do not know of proposals for architectures based on optical routing in which paths sharing a wavelength must remain node–disjoint.

When the number of terminal pairs is restricted to be a fixed constant, and the graph is planar, a polynomial-time algorithm for (1) and (1′) was first given by Robertson and Seymour in [15]. A more practically efficient algorithm for this case was subsequently given by Schrijver [16], who also gave a polynomial-time algorithm for the case in which all terminals lie on the boundaries of a designated set of faces, and the homotopies of the paths with respect to these faces is fixed. This extends work of Cole and Siegel [4] and Leiserson and Maley [11] for the mesh. This approach based on homotopy appears not to be directly applicable to our problem, for although we could test any fixed choice of homotopies for our paths in polynomial time, there are exponentially many choices of homotopies to try.

Let us also indicate some ways in which the node–disjoint paths problem on the mesh differs fundamentally from its edge–disjoint counterpart. First of all, letting $v_{i,j}$ denote the node at row i and column j of the mesh, consider the set $\mathcal{T}$ of consisting of all pairs of the form $(v_{i,1}, v_{n,i})$, $i = 1, \ldots, n$. This set of terminal pairs satisfies the usual *cut condition*, stating that there is no way to delete k nodes and separate $k+1$ pairs of terminals, and in fact the set $\mathcal{T}$ is routable in one round on edge–disjoint paths. However, since any pair of paths in a routing of $\mathcal{T}$ must cross, $\mathcal{T}$ requires n rounds when paths must be node–disjoint.

This example actually does not rely on the terminals being positioned near the boundary of the mesh. Consider an infinite grid graph, let $p = m^2$ for some parameter m, and $\mathcal{T}$ denote the set of all pairs of the form $(v_{im+j,0}, v_{jm+i,p})$, with $1 \leq i, j \leq m$. Then it is not hard to show by a crossing number argument (using a lemma of [10]) that $\chi(\mathcal{T}) = \Theta(\sqrt{p})$, even though $\mathcal{T}$ is again routable in a single round using edge–disjoint paths.

As a final example, we show that while Theorem 1.3 implies that 6 bends are sufficient to route any permutation in $O(n^{7/5})$ rounds, there exist partial permutations that require $\Omega(n^2)$ rounds in any one–bend routing.

Proposition 1.4 *There exists a partial permutation $\mathcal{T}$ on G for which there is no one–bend routing using fewer than $\frac{1}{8}n^2$ rounds.*

Proof. Assume for simplicity that n is even; so $n = 2m$. Let $\mathcal{T}$ consists of all pairs of the form $(v_{i,j}, v_{m+j,m+i})$, where $1 \leq i, j \leq m$. Let

$$X = \{v_{i,j} : 1 \leq i \leq m < j \leq 2m\}$$

$$Y = \{v_{i,j} : 1 \leq j \leq m < i \leq 2m\}.$$

Then in any one–bend routing of $\mathcal{T}$, at least $\frac{1}{8}n^2$ of the paths pass though either X or Y, and every pair of such paths must cross at some node. ■

1.2 Sketch of the Algorithm

We will show that it is possible, with a constant–factor loss in the bound, to reduce the problem for an arbitrary set of terminals to one in which there are two $k \times k$ subsquares of G, denoted A and B, which are spaced $4k$ apart from one another; and every terminal pair has one end in A and the other in B. If there are p terminal pairs, we will show how to route $p/(k \log k)$ of them simultaneously; applying this process greedily gives the bound of Theorem 1.1.

The approach of Cutler and Shiloach [5] requires, in order to route q terminals in a single round, that the inter–terminal spacing be at least q. Using this, one can therefore only be sure of routing a set of size $p^{1/3}$, resulting in an overall bound of $n^{4/3}$ for $\chi(\mathcal{T})$.

We start by considering the following simple notion: suppose we tried choosing at most one terminal from each row of A and at most one from each row of B, and then joining as many of these terminals as possible in the region between A and B. There are a number of difficulties with this; in particular, there may be no way to choose terminals in this way so that nearly p/k can be joined in node–disjoint fashion. Let us call this above approach *row–selection* — one tries to pick a unique terminal from each row so that the selected terminals in A can be routed out of A on their rows, and then joined with their selected counterparts in B, in such a way that no paths cross.

Our main technique is the following generalization of row–selection: although there may be no good way to choose unique terminals from the rows of A and B, we show that there exists a "rotation" of A and a "rotation" of B, so that the resulting set of rows allows for such a good selection. Now, meshes are discrete objects, so the notion of rotation has to be defined carefully — we use monotone random walks of varying biases as a basic way of "rotating" A and B.

Having chosen a candidate pair of rotations, we decompose A and B into *pseudo–rows.* The task of *pseudo–row selection* — choosing a unique terminal from each pseudo–row — is based on analyzing the bipartite graph H on the set of pseudo–rows, in which a pseudo-row of A is joined to a pseudo-row of B if there is some terminal pair with one end in each. We show that one can choose rotations of A and B so that this bipartite graph contains a large *monotone matching* — one in which no edges cross. The pairs corresponding to the edges of this matching can be routed in a single round.

With inter–terminal spacing of $d > 1$, a very similar approach is applicable. The main difference is that we can now afford to choose edge sets of the bipartite graph H which are not necessarily monotone matchings; we can handle a limited amount of non-monotonicity using the technique of Cutler–Shiloach within each pseudo–row.

2 Preliminaries

In this section, we develop some basic lemmas that will be useful in analyzing our algorithms. In Section 3, we consider the basic problem of a permutation on the mesh, and prove Theorem 1.1. In Section 4, we give our algorithm for routing with inter-terminal spacing in the model of [5, 1]. Finally, in Section 5, we discuss algorithms for routing with only a constant number of bends per path.

Lemma 2.1 *Let $x_1, \ldots, x_d$ be positive integers for which $\sum_i x_i = a$ and $\sum_i x_i^2 = b$. Then $d \geq a^2 b^{-1}$.*

Proof. $a^2 = (\sum_i x_i)^2 \leq d \sum_i x_i^2 = db$. ∎

The following is easily proved by induction on b.

Lemma 2.2 *For non-negative integers a and b, one has*

$$\int_0^1 x^a (1-x)^b \, dx = (a+b+1)^{-1} \binom{a+b}{b}^{-1}.$$

Lemma 2.3 *Consider the following random experiment: we first choose a bias α for a coin uniformly from the interval $[0, 1]$, and then flip the coin n times. The probability that the number of heads is k is equal to $\frac{1}{n+1}$.*

Proof. The probability is equal to

$$\binom{n}{k} \int_0^1 \alpha^k (1-\alpha)^{n-k} \, d\alpha,$$

which by Lemma 2.2 is equal to $\frac{1}{n+1}$. ∎

There is also a purely combinatorial proof of this, which we leave as an exercise for the reader.

2.1 Monotone Matchings

We make use of a notion that we call a *monotone matching.* Here we define and develop some basic facts about monotone matchings at a general level.

In this section, let H denote a bipartite graph with bipartition (U, V), $U = \{u_1, \ldots, u_n\}$, $V = \{v_1, \ldots, v_n\}$; and let e denote the number of edges of H. As usual, by a *matching* in H we mean a set of edges that have no endpoints in common, and we let $\mu(H)$ denote the size of the largest matching in H.

Definition 2.4 *A matching in H is said to* monotone *if it consists of edges $(u_{i_1}, v_{j_1}), \ldots, (u_{i_k}, v_{j_k})$ such that the sequence $i_1, \ldots, i_k$ is increasing, and the sequence $j_1, \ldots, j_k$ is either increasing or decreasing. We say the matching is* increasing *in the former case and* decreasing *in the latter.*

We give two lower bounds on the size of the largest monotone matching in H.

Lemma 2.5 *H has a monotone matching of size at least $\sqrt{\mu(H)}$.*

Proof. Let M be a matching of maximum size in H, and order its edges so that their endpoints in U have increasing indices. Consider the resulting sequence of indices in V. By a lemma due to Erdös and Szekeres [6], this sequence must have either an increasing sequence or a decreasing sequence of length at least $\sqrt{|M|} = \sqrt{\mu(H)}$. The edges corresponding to this sequence constitute a monotone matching. ∎

Lemma 2.6 *H has an increasing monotone matching of size at least $e/2n$.*

Proof. Let M_k denote the set of all edges (u_i, v_j) for which $j = i + k \bmod n$. Let M_k' denote the subset of M_k in which $j \geq i$, and M_k'' the subset of M_k in which $j < i$. Now, each M_k' and M_k'' is a monotone matching, and since they are all disjoint, one has size at least $e/2n$. ∎

An example due to Don Coppersmith shows that for every $d \leq n$, there exist d-regular bipartite graphs on $2n$ nodes with no monotone matching of size greater than $O(\max(d, \sqrt{n}))$.

Finally, it is also useful to establish "covering" versions of these lemmas. We state these here.

Lemma 2.7 *The edges of H can be covered by $O(\Delta\sqrt{n})$ monotone matchings.*

Lemma 2.8 *The edges of H can be covered by $2n$ increasing monotone matchings.*

3 The bound for $\chi(\mathcal{T})$

For vertices $x, y \in G$, let $d(x, y)$ denote the shortest-path distance between them. As indicated above, we will first consider the following special case: there are $k \times k$ subsquares A and B of G, with $d(A, B) \geq 4k$, each terminal pair has one end in A and the other in B, and we must use paths that stay within $d(A, B)$ of $A \cup B$. Below we show how a bound for this special case this can be used to prove Theorem 1.1.

(0,120)(140,120)(140,0) (0,0)(0,120) (280,120)(420,120)(420,0) (280,0)(280,120) (0

Figure 1: Decomposing into pseudo-rows

We represent A as a union of *pseudo-rows*, as follows. Let v_1 denote the lower left vertex of A. We choose $\alpha \in [0,1]$ uniformly at random, and perform the following random walk for k time steps starting from v_1:

(i) With probability $1-\alpha$, move one step to the right.

(ii) With probability α, move one step up and then one step to the right.

Let L_1^α denote the path traversed by this walk; note that it is contained in A. For $-k \leq i \leq k$, define L_i^α to be a copy of this path translated $(i-1)$ steps upward and then intersected with A. The sets L_i^α will be called *pseudo-rows.*

For $x \in A$, let $R_x^\alpha = \{i : x \in L_i^\alpha\}$. The following is immediate from the construction, but worth noting.

Lemma 3.1 *If $v \in A$ belongs to both L_i^α and L_j^α, then $|j-i| = 1$. Thus $|R_v^\alpha| \leq 2$ (v belongs to at most two of these sets); and in particular, $E|R_v^\alpha| \leq 2$.*

We say that $x, y \in A$ are *collinear* if there is some pseudo-row that contains them both; in this case we write $x \sim y$.

For a vertex $x \in G$, let $a(x)$ denote its row number and $b(x)$ denote its column number. For two vertices x and y, let $d_\infty(x,y)$ denote the L_∞ distance between them; that is,

$$d_\infty(x,y) = \max\left(|a(x)-a(y)|, |b(x)-b(y)|\right).$$

Lemma 3.2 *The probability that $x, y \in A$ are collinear is at most $\frac{2}{d_\infty(x,y)}$.*

Proof. Suppose without loss of generality that $b(x) \leq b(y)$. We have

$$\begin{aligned}
Pr(x \sim y) &= \int_0^1 Pr(x \sim y \mid \alpha = t)\; dt \\
&\leq \int_0^1 \sum_i Pr(x, y \in L_i^t)\; dt \\
&= \int_0^1 \sum_i Pr(x \in L_i^t) \cdot Pr(y \in L_i^t \mid x \in L_i^t)\; dt \\
&= \int_0^1 Pr(y \in L_i^t \mid x \in L_i^t) \cdot \sum_i Pr(x \in L_i^t)\; dt \\
&= \int_0^1 Pr(y \in L_i^t \mid x \in L_i^t) \cdot E|R_x^t|\; dt \\
&\leq 2 \int_0^1 Pr(y \in L_i^t \mid x \in L_i^t)\; dt.
\end{aligned}$$

If it is not the case that $b(y) - b(x) \geq a(x) - a(y) \geq 0$, then this last integral is 0. Otherwise, it is simply the probability, over a uniformly distributed bias, that a given number of heads will come up when a coin with that bias is flipped $d_\infty(x, y)$ times. By Lemma 2.3, this is $\frac{1}{1+d_\infty(x,y)}$. The lemma follows. ■

Lemma 3.3 *There is an absolute constant c' so that if there are p terminal pairs with ends in A and B, then there is a realizable subset of size at least $p/(c' k \log k)$.*

Proof. We first choose a random α and decompose A into pseudo-rows. We do the same for B, using a random β. Now by Lemma 3.1, we can choose half the pseudo-rows of A and half the pseudo-rows of B so that all pseudo-rows are disjoint and at least $q = p/4$ of the terminal pairs have both ends in these chosen pseudo-rows. Let $\mathcal{T}'$ denote a set of q such pairs.

Define a *collineation* to be a pair i, j such that s_i and s_j are collinear in A and t_i and t_j are collinear in B. We wish to bound the expected number of collineations among the pairs in $\mathcal{T}'$; this is simply the sum

$$\sum_{i,j} Pr(s_i \sim s_j) \cdot Pr(t_i \sim t_j).$$

This can be viewed as the dot product of two vectors of dimension $\binom{q}{2}$, one with coordinates $Pr(s_i \sim s_j)$ and the other with coordinates $Pr(t_i \sim t_j)$. We observe that this dot product will be maximized if $Pr(s_i \sim s_j) = Pr(t_i \sim t_j)$ for all pairs i, j. For the dot product is certainly maximized, once the lengths of the vectors is fixed, by having them parallel to one another; and in this case, if their lengths differ, one can increase the dot product by changing the coordinates of the shorter one to be equal to the longer one.

Thus, the expected number of collineations is bounded by the largest possible sum

$$\sum_{x,y \in S} Pr(x \sim y)^2,$$

where S is a subset of A of size q. We can write this sum as

$$\frac{1}{2} \sum_{x \in S} \sum_{y \in S \setminus \{x\}} Pr(x \sim y)^2 \leq \frac{1}{2} \sum_{x \in S} \sum_{y \in S \setminus \{x\}} \frac{4}{d_\infty(x, y)^2},$$

with the inequality following from Lemma 3.2. This inner sum is maximized by having the vertices y fill in a ball centered at x, in which case its value is bounded by

$$\sum_{i=1}^{\frac{1}{2}\sqrt{q}} \frac{32i}{i^2} \leq 32 + 16 \ln q \leq c \ln q$$

for an absolute constant c. Thus, summing over all x, we see that the expected number of collineations in $\mathcal{T}'$ is at most $\frac{1}{2} cq \ln q$.

Thus, there exist α, β and decompositions of A and B into pseudo-rows for which there is a subset $\mathcal{T}'$ of q terminal pairs whose number of collineations is at most $\frac{1}{2} cq \ln q$. For this set $\mathcal{T}'$, let us build the following bipartite multigraph H. There is one vertex u_i on the left for each chosen pseudo-row A, and one vertex v_j on the right for each chosen pseudo-row B. We now put w_{ij} parallel edges from u_i to v_j if there are w_{ij} terminals with one end in the pseudo-row u_i and the other end in v_j. From H, we build a simple graph H' which contains an edge (u_i, v_j) iff $w_{ij} > 0$.

We claim that the number of edges of H' is at least $q/(2c \ln q)$. To see this, note that the number of collineations in $\mathcal{T}'$ is precisely

$$\sum_{i,j} \binom{w_{ij}}{2},$$

whence we have $\sum_{i,j} w_{ij}^2 \leq 2cq \ln q$. Since $q = |T'| = \sum_{i,j} w_{ij}$, the claim follows from Lemma 2.1.

Finally, the number of vertices in each half of the bipartition of H' is at most k, so by Lemma 2.6, H' has a monotone matching of size at least

$$\frac{q}{4ck \ln q} \geq \frac{p}{16ck \ln q} \geq \frac{p}{32ck \log k}.$$

We can route one terminal for each edge of the monotone matching, which implies the lemma with $c' = 32c$. Note that we need $d(A, B) = \Omega(k)$ in order to ensure that we correctly align the matched pseudorows of A with the corresponding pseudorows of B. ■

Lemma 3.4 *All terminal pairs with ends in A and B can be routed in $O(k \log^2 k)$ rounds.*

Proof. We produce the partition into realizable sets greedily, using Lemma 3.3. The bound follows since the number of remaining terminals goes down by a $(1 - \frac{c'}{k \log k})$ fraction for each realizable set we produce. ■

Proof of Theorem 1.1. For an integer τ, let $\mathcal{T}_\tau$ denote the set of terminal pairs (s_i, t_i) for which $\frac{1}{2}\tau \leq d(s_i, t_i) \leq \tau$. We prove that $\mathcal{T}_\tau$ can be routed in $O(\tau \log^2 \tau)$ rounds; the overall bound on the number of rounds then follows since, for an absolute constant c'',

$$\begin{aligned}
\chi(\mathcal{T}) &\leq \sum_{i=0}^{\lceil \log n \rceil} \chi(\mathcal{T}_{2^i}) \\
&\leq \sum_{i=0}^{\lceil \log n \rceil} c'' 2^i \log^2(2^i) \\
&\leq 4c'' n \log^2 n.
\end{aligned}$$

To prove that $\chi(\mathcal{T}_\tau) = O(\tau \log^2 \tau)$, we show that $\mathcal{T}_\tau$ can be partitioned into a constant number of sets, and each of these sets can be partitioned into pairs of well–separated squares; we then invoke the algorithm of Lemma 3.4 on all the pairs in a single set simultaneously.

Choose $\tau' = \frac{1}{16}\tau$, and define S_{ij} to be the $\tau' \times \tau'$ subsquare of G whose upper–left corner lies in row $i\tau'$ and column $j\tau'$. Each pair of squares $(S_{ij}, S_{\ell m})$ defines a subset of $\mathcal{T}_\tau$, consisting of those terminal pairs with one end in S_{ij} and the other in $S_{\ell m}$. We call such a pair *active* if it contains a terminal pair in $\mathcal{T}_\tau$. By definition, any active pair $(S_{ij}, S_{\ell m})$ satisfies

$$6 \leq |\ell - i| + |m - j| \leq 16.$$

Thus, by Lemma 3.4, we can route all the terminal pairs in the set defined by any active pair in $O(\tau \log^2 \tau)$ rounds.

We say that active pairs $(S_{ij}, S_{\ell m})$ and $(S_{i'j'}, S_{\ell' m'})$ *interfere* with one another if there is some vertex within distance τ of both $S_{ij} \cup S_{\ell m}$ and $S_{i'j'} \cup S_{\ell' m'}$. Let us build a graph $\mathcal{K}$ on the set of active pairs, joining two by an edge if they interfere with one another. This graph has constant degree, and hence can be colored with a constant number of colors. But we can use the same set of rounds for routing *all* the active pairs in a single color class, and thus $\chi(\mathcal{T}_\tau) = O(\tau \log^2 \tau)$.

The proof of the other statement of Theorem 1.1, that every permutation contains a routable set of size $\Omega(n/\log n)$, is strictly analogous. We first argue, using the interference graph $\mathcal{K}$ and Lemma 3.3 instead of Lemma 3.4, that $\mathcal{T}_\tau$ contains a routable set of size $\Omega(|\mathcal{T}_\tau|/\tau \log \tau)$. The bound now follows, since for some $\tau \in \{2^i : 0 \le i \le \lceil \log n \rceil\}$ we have $|\mathcal{T}_\tau| = \Omega(n\tau)$. ■

4 Terminals with Spacing

The proof of Theorem 1.2 is essentially that of Theorem 1.1, with one additional step added. As before, we can specialize to the case of two $kd \times kd$ squares A and B that contain p terminal pairs. To construct pseudo-rows, we first partition A and B into disjoint $d \times d$ subsquares, contract these subsquares, and then use the previous construction.

Lemma 4.1 *There is an absolute constant c' so that if there are p terminals with ends in A and B, then there is a realizable subset of size at least $p\sqrt{d}/(c' k \log k)$.*

Proof. We follow the proof of Lemma 3.3 up through the construction of the bipartite graph H' with at most k vertices in each half of its bipartition, and with at least $q/(c \ln q)$ edges. Now define a *slice* in H' to be a set of the form $U' \cup V'$, where U' is a set of $\sqrt{d}$ consecutive vertices on the left side of H' and V' is a set of $\sqrt{d}$ consecutive vertices on the right side. The crucial point is that there are only d edges of H' with both ends in a given slice, and so using the technique of [1, 5], we can route one terminal pair for each of these edges in a single round.

We now construct a graph H'' by identifying blocks of $\sqrt{d}$ consecutive vertices of H' on the left and on the right. (We can essentially think of edges of H'' as corresponding to particular slices of H'.) Let us give each edge (u^*, v^*) in H'' a weight equal to the number of edges of H' that have ends in the super-nodes u^* and v^*; thus these weights are integers between 0 and d. By the weighted analogue of Lemma 2.6, H'' has a monotone matching of weight at least

$$\frac{\text{weight}(H'')}{2|V(H'')|} = \frac{|E(H')|}{2|V(H'')|} \ge \frac{q\sqrt{d}\log k}{4ck \ln q} \ge \frac{p\sqrt{d}}{32ck\log k}.$$

We can route a number of terminals equal to the weight of any monotone matching in H'', as argued above; this implies the lemma. ■

Proof of Theorem 1.2. Again reducing to the case of $kd \times kd$ squares A and B, we use a greedy approach based on Lemma 4.1. Each time we reduce the number of remaining cccccccc 0 0 0 cccccccc 0 0 0 cccccccc 0 0 0 cccccccc 0 0 0 cccccccc 0 0 0 cccccccc 0 0 0 cccc

Figure 2: Routing with a constant number of bends

terminals by a factor of at least $(1-\frac{c'\sqrt{d}}{k\log k})$; thus we route all terminal pairs in $O(kd^{-\frac{1}{2}}\log^2 k)$ rounds. ■

5 Constant-Bend Routing

As in the previous sections, let us first consider the case in which all terminal pairs have ends in $k \times k$ squares A and B, with $d(A,B) \geq 4k$. Here, we additionally assume that A and B are each at a distance of at least $\Omega(k^{7/10})$ from the boundary of the mesh. Let us suppose that the lowest column number of a node in A is less than or equal to the lowest column number of a node in B.

The main difficulty we encounter here is the following: since we wish to use only a constant number of bends, the approach based on pseudo–rows appears to be inherently inapplicable, as it involves partitioning A and B into paths with many bends. Thus, in this case, we build a graph H with a vertex u_i for each row of A, and a vertex v_j for each row of B. We give the edge (u_i, v_j) a weight w_{ij} equal to the number of terminal pairs with one end in each row.

To give a sense of the algorithm, we first mention a very easy algorithm for routing in $O(k^{3/2})$ rounds. This is as follows. First replace each edge (u_i, v_j) of H with w_{ij} parallel edges. Now by Lemma 2.7, the edges of H can be covered by $O(k^{3/2})$ monotone matchings. Moreover, we may assume that each matching in the cover contains at most $O(k^{1/2})$ edges. We claim that all the terminal pairs corresponding to a single one of these matchings can be routed in a single round. For if the matching is increasing, then the pairs can be routed by having each terminal leave A along its row through the east wall, and enter B along the matching row through the west wall; and if the matching is decreasing, then the pairs can be routed by having each terminal leave A along its row through the west wall and enter B along the matching row through the west wall. (See Figure 2 for a version of the latter case.) Moreover, since A and B are at least $\Omega(k^{1/2})$ distance from the boundary of the mesh, there is room for all the paths corresponding to a single matching to be routed in the region outside $A \cup B$.

To get a bound of $O(k^{7/5})$, we need a slight strengthening of this technique. We divide the edges into $\log k$ classes, by assigning each edge to the smallest power of 2 greater than its weight. Let H_w denote the subgraph of H consisting only of edges that are assigned the weight w, and let $\mathcal{T}_w$ denote the set of terminals corresponding to edges of H_w.

Lemma 5.1 *$\mathcal{T}_w$ can be routed in at most $2kw$ rounds, using at most four bends.*

Proof. Using Lemma 2.8, we can partition the edges of H_w into $2k$ (increasing) monotone matchings. Each edge of H_w corresponds to w terminals, so all terminals can be routed in $2kw$ rounds. It is easy to construct paths for these terminals using at most four bends. ■

Lemma 5.2 *$\mathcal{T}_w$ can be routed in at most*

$$\max[O(k^{3/2}/w^{1/4}), O(k^{13/10})]$$

rounds, using at most six bends.

Proof. The maximum degree of a vertex in H_w is k/w, so we can partition the edges of H_w into k/w matchings. Let us say that a matching M is *well-spaced* if for edges (u_i, v_j) and (u_k, v_ℓ) of M, the difference between i and k is at least $\sqrt{w}$, and the distance between j and ℓ is at least $\sqrt{w}$. By an easy coloring argument, we can partition any matching in H_w into $O(\sqrt{w})$ well-spaced matchings, and thus we can partition the edges of H_w into $O(k/\sqrt{w})$ well-spaced matchings. Finally, each of these well-spaced matchings is only on $O(k/\sqrt{w})$ vertices, so by Lemma 2.7 we can partition each one into $O(\sqrt{k}/w^{1/4})$ monotone matchings.

Thus, we now have a total of $O(k^{3/2}/w^{3/4})$ monotone matchings covering the edges of H_w, and each has the property that the endpoints of its edges are mutually at distance $\sqrt{w}$. We use a different set of rounds for routing each monotone matching.

Now, how many rounds are needed for a single monotone matching M? (Recall that each edge of M represents up to w terminals.) Let A_i (resp. B_i) denote the set of vertices in the i^{th} row of A (resp. B). Since the edges of M are anchored in rows of A and B that are mutually $\sqrt{w}$ apart, we can assign to each edge $e = (u_i, v_j)$ a "band" of rows of width $\Omega(\sqrt{w})$ around row A_i and a similar band around row B_j. The pairs corresponding to an edge e are themselves a matching on the vertex set $A_i \cup B_j$. Thus these pairs can be partitioned, again by Lemma 2.7, into $O(\sqrt{w})$ monotone matchings, and each of *these* monotone matchings can be routed in a single round in the bands surrounding rows A_i and B_j.

Finally, we can do this simultaneously (i.e. using the same set of rounds) for all edges of a single well-spaced monotone matching, provided there is sufficient room between the squares A and B and the boundary of the mesh for routing all the paths. If there is sufficient room, then the number of rounds used is $O(k^{3/2}/w^{1/4})$. Otherwise, we must break up the rounds that use too many paths into sub-rounds that only route $O(k^{7/10})$ terminals at a time; the resulting bound is $O(k^{13/10})$ rounds. ■

Lemma 5.3 *The set of terminals with ends in A and B can be routed in $O(k^{7/5})$ rounds, using at most six bends.*

Proof. Let $r(\mathcal{T})$ denote the total number of rounds required for routing $\mathcal{T}$ with six bends. Then

$$\begin{aligned} r(\mathcal{T}) &\leq \sum_{w=2^i} r(\mathcal{T}_w) \\ &\leq \sum_{w=2^i \leq k^{2/5}} r(\mathcal{T}_w) + \sum_{k^{2/5} \leq w=2^i \leq k^{4/5}} r(\mathcal{T}_w) + \sum_{w=2^i \geq k^{4/5}} r(\mathcal{T}_w) \\ &\leq O(k) \sum_{w=2^i \leq k^{2/5}} w + O(k^{3/2}) \sum_{k^{2/5} \leq w=2^i \leq k^{4/5}} w^{-1/4} + O(k^{13/10}) \sum_{w=2^i \geq k^{4/5}} 1. \end{aligned}$$

Since all three of the terms in this last sum are easily seen to be $O(k^{7/5})$, the bound follows. ■

Proof of Theorem 1.3. We use the notion of an interference graph $\mathcal{K}$ on pairs of subsquares as in the proof of Theorem 1.1. The only change is that we can now only work with $k \times k$ squares that have either no row or no column within $O(k^{7/10})$ of the boundary. (It is enough to have this guarantee for either rows *or* columns, since we could just as well have built our graph H on the set of columns of one of the squares.) To handle this, we define G' to be the subgraph of G induced by all nodes within distance $O(n^{2/5})$ of the boundary, or within distance $O(n^{7/10})$ of a corner of the mesh. Since $|G'| = O(n^{7/5})$, we can use a different round for each terminal pair with an end in G'; we then run the algorithm above (using the interference graph $\mathcal{K}$) in $G \setminus G'$. ■

6 Conclusion

There are a number of natural questions suggested by this work. First, the algorithm of Theorem 1.2 routes a permutation with spacing d in $O(nd^{-1/2} \log^2 n)$ rounds, and thereby gives an upper-bound trade-off between inter-terminal spacing and the number of rounds required. However, we do not know how close to optimal this trade-off is for all values of d. At the extreme ends of the range $0 \leq d \leq n^2$, the number of rounds used by our algorithm is tight to within polylogarithmic factors — this follows from the bisection bound when $d = \Theta(1)$ and from [1] when $d = \Theta(n^2)$. While it is possible that the trade-off is tight to within polylogarithmic factors for all values of d, we are not able to prove this.

In a somewhat different direction, the algorithm of Theorem 1.1 gives a routing for a set $\mathcal{T}$ of terminal pairs for which the number of rounds used is close to the best possible in the worst case. However, it is not an *approximation algorithm*, since we do not have a result relating the number of rounds we require for a given set $\mathcal{T}$ to the optimal value $\chi(\mathcal{T})$. It would be very interesting to obtain such an algorithm, since it would appear to require new techniques for analyzing $\chi(\mathcal{T})$ in cases when the cut condition gives no information.

Finally, it would be interesting to decide whether $\chi(\mathcal{T}) = O(n)$ for every permutation on the mesh.

Acknowledgements

We would like to thank Don Coppersmith, Tom Leighton, Baruch Schieber, and Madhu Sudan for valuable discussions on these problems.

References

[1] A. Aggarwal, M. Klawe, D. Lichtenstein, N. Linial, A. Wigderson, "Multi-layer grid embeddings," *Proc. 26th IEEE FOCS*, 1985.

[2] A. Aggarwal, M. Klawe, D. Lichtenstein, N. Linial, A. Wigderson, "A lower bound on the area of permutation layouts," *Algorithmica*, 6(1991), pp. 241–255.

[3] A. Aggarwal, M. Klawe, P. Shor, "Multi-layer grid embeddings for VLSI," *Algorithmica*, 6(1991), pp. 129–151.

[4] R. Cole, A. Siegel, "River routing every which way but loose," *Proc. 25th IEEE FOCS*, 1984.

[5] M. Cutler, Y. Shiloach, "Permutation layout," *Networks*, 8(1978), pp. 253–278.

[6] P. Erdös, G. Szekeres, "A combinatorial problem in geometry," *Compositio Math.*, 2(1935), pp. 463–470.

[7] M. Klawe, F.T. Leighton, "A tight lower bound on the area of planar permutation layouts," *SIAM J. Discrete Math.*, 5(1992), pp. 558–563.

[8] J. Kleinberg, É. Tardos, "Disjoint paths in densely embedded graphs," *Proc. 36th IEEE FOCS*, 1995.

[9] 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.

[10] F.T. Leighton, *Layouts for the Shuffle–Exchange Graph and Lower Bound Techniques for VLSI*, Ph.D. Thesis, MIT Math Dept., 1981.

[11] C. Leiserson, F.M. Maley, "Algorithms for routing and testing routability of planar VLSI layouts," *Proc. 17th ACM STOC*, 1985.

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

[13] P. Raghavan, E. Upfal, "Efficient all-optical routing," *Proc. 26th ACM STOC*, 1994.

[14] R. Rivest, "The placement and interconnect system," *Proc. 19th Design and Automation Conference*, 1982.

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

[16] A. Schrijver, "Disjoint homotopic paths and trees in a planar graph," *Discrete and Computational Geometry*, 6(1991), pp. 527–574.