Path Coloring on the Mesh

Yuval Rabani*

Abstract

In the minimum path coloring problem, we are given a list of pairs of vertices of a graph. We are asked to connect each pair by a colored path. Paths of the same color must be edge disjoint. Our objective is to minimize the number of colors used. This problem was raised by Aggarwal et al [1] and Raghavan and Upfal [22] as a model for routing in all-optical networks. It is also related to questions in circuit routing.

In this paper, we improve the $O(\ln N)$ *approximation result of Kleinberg and Tardos [14] for path coloring on the* $N \times N$ *mesh. We give an* $O(1)$ *approximation algorithm to the number of colors needed, and a poly*$(\ln \ln N)$ *approximation algorithm to the choice of paths and colors. To the best of our knowledge, these are the first sub-logarithmic bounds for any network other than trees, rings, or trees of rings. Our results are based on developing new techniques for randomized rounding. These techniques iteratively improve a fractional solution until it approaches integrality. They are motivated by the method used by Leighton, Maggs, and Rao [18] for packet routing.*

1 Introduction

The problem. An instance of the minimum path coloring problem (MPCP) in a graph G is specified by listing pairs of vertices of G, (s_1, t_1), (s_2, t_2), $\ldots$, (s_n, t_n). A solution to this instance specifies n paths, path i connecting s_i and t_i, and a color for each path. The assigned paths and colors must satisfy the condition that paths of the same color are edge disjoint. The objective function in the optimization version of the problem is to minimize the number of colors used. A trivial reduction from the disjoint paths problem shows that MPCP is NP-hard, even for the special case of the mesh (see [19] regarding the complexity of the disjoint paths problem on the mesh).

This problem has obvious application to circuit routing and optical routing. Consider a communication network, modeled as the graph G, where the pairs s_i, t_i denote requests for connection between the source s_i and the target t_i. The network might interconnect the processors of a parallel supercomputer, or provide telecommunication services, such as video on demand.

In an all-optical network, the source transmits a light signal at a certain frequency, which the target is tuned to. Optical switches are used to set the path for this signal. Proposed switches do not modify the wavelength of the transmission (see [1, 22]). Wavelength division multiplexing (WDM) is used to enhance the capacity of the network. By WDM, many signals can be carried simultaneously along a link, as long as they do not use the same wavelength. In this case, MPCP is equivalent to the problem of minimizing the number of different wavelengths required to transmit a batch of requests.

In a circuit routed network, a circuit (path) has to be reserved connecting the source to the target. In this case, MPCP is equivalent to the problem of minimizing the number of rounds of communication needed to process all communication requests.

Results and methods. We consider the minimum path coloring problem on the two dimensional mesh.

*Work supported by a David and Ruth Moskowitz academic lectureship award and by grants from the fund for the promotion of sponsored research and from the fund for the promotion of research at the Technion. Address: The Technion, Computer Science Department, Haifa 32000, Israel. Email: `rabani@cs.technion.ac.il`. URL: `http://www.cs.technion.ac.il/~rabani`.

(Our results can be extended to handle some mesh-like networks, such as the hex. Due to lack of space, we do not present these extensions here.) We formalize MPCP as a sequence of natural integer linear programs. We then show that the solutions to their LP relaxations provides a constant factor approximation to the number of colors needed. The integrality gap is established via a complicated randomized rounding procedure. We round the fractional solution iteratively. Each iteration has a double purpose. It gets us closer to an integral solution, and in doing so reduces the dependencies among bad events in the randomized rounding sample space. These dependencies prevent us from reaching an integral solution at a single rounding step. Given a B^{-1}-integral solution, the iteration produces a $\ln^{-3} B$-integral solution. After a small number of iterations, the solution is near-integral, and we can convert it into an integral solution without much loss in the number of colors. This gives a polynomial time constant factor approximation to the number of colors needed.

The basic step in an iteration uses the Lovász Local Lemma [11]. The choice of paths and colors is therefore non-constructive. Unfortunately, algorithmic versions of the Local Lemma [10, 2, 17] do not seem to apply to our results, primarily because of our use of the asymmetric version of the Local Lemma (see [3]). Nevertheless, we are able to modify our arguments to get a weaker guarantee, which has an algorithmic implementation. We present a poly$(\ln \ln N)$ approximation algorithm to MPCP on the $N \times N$ mesh. This improves over the previously best bound of $O(\ln N)$ (see below). We note that our techniques are equally useful in approximating the maximum disjoint paths problem (MDPP) on the mesh, though the results obtained are inferior to those of Kleinberg and Tardos in [15]. We further note that while the hidden constants in our bounds are huge, previous results suffer from the same problem. We make no attempt to optimize these constants.

Previous work. The minimum path coloring problem drew attention in the context of recent theoretical work on optical routing. Aggarwal et al [1] followed by Raghavan and Upfal [22] examine suggested optical switches and the routing models they imply. The latter give, among other results, constant approximation algorithms to MPCP in trees, rings, and trees of rings.

MPCP has close connection to MDPP. As observed in [5], an MDPP approximation algorithm can be used to obtain an MPCP approximation algorithm, with a logarithmic factor loss in the performance guarantee. There are excellent approximations to MDPP in networks with edge capacities in $\Omega(\log |E|)$ (or, in other words, where there are $\Omega(\log |E|)$ parallel copies of each edge), using Raghavan and Thompson's randomized rounding technique [21, 20]. In fact, assuming the large number of parallel edges, MPCP can be approximated with similar guarantees on any graph, using randomized rounding. Good on-line approximations to MDPP assuming large capacities are also known [6] (see also [4]). However, in many applications to circuit routing, the assumption that edge capacities are large is false (see [8]). Similarly, the assumption of numerous parallel edges is unrealistic for optical network topologies.

Despite considerable attention to MDPP (see the collection edited by Korte et al [16]), only recently has there been some work on approximation algorithms to it. Earlier work centered mainly on identifying polynomial time solvable cases. Garg, Vazirani, and Yannakakis [12] were apparently the first to consider approximation algorithms for the problem. They give a 2-approximation to MDPP in trees with arbitrary capacities. This gives an $O(\log |V|)$ approximation to MPCP in a slightly more general setting than the constant approximation of [22]. On-line approximation algorithms for trees with logarithmic factor guarantees are also known [7, 8].

On the mesh, Awerbuch et al [8] present an on-line $O(\ln N \ln \ln N)$ approximation algorithm to MDPP. Kleinberg and Tardos [14], and independently Aumann and Rabani [5] give an (off-line) $O(\ln N)$ approximation to MDPP on the mesh. The former extend this result to a class of mesh-like planar graphs. They also show that their methods give a similar approximation guarantee to MPCP, without the loss of the extra logarithmic factor. In a sequel paper, Kleinberg and Tardos [15] give a constant approximation, and an on-line $O(\ln N)$ approximation, to MDPP on the mesh and a (different) class of mesh-like graphs. This implies another $O(\ln N)$ approximation algorithm to MPCP. Their result, too, involves embedding a simulated network. We note that their embedding is insufficient for our purposes, because it cannot efficiently cover more than a constant fraction of the terminals. Very recently, Bartal and Leonardi [9] have announced an on-line $O(\ln N)$ approximation algorithm to MPCP on the mesh.

Our iterative rounding procedure is mathematically similar to, and is motivated by the argument given by Leighton, Maggs, and Rao [18] for packet routing in $O(\text{congestion} + \text{dilation})$. Viewed as an optimization problem, their procedure iterates through non-feasible integral solutions and gradually approaches feasibility. Our approach, which requires overcoming some additional technical difficulties, is to maintain feasibility and gradually reach integrality.

Srinivasan [23, 24] recently used correlation analysis, as well as some extensions to the Local Lemma, to improve approximation results to sparse packing and covering integer programs.

2 Preliminaries

Let M_N denote the $N \times N$ mesh. We denote its nodes by integer pairs (i, j), where $0 \le i < N$, $0 \le j < N$, and $(0, 0)$ denotes the lower left corner. A *λ-tile* is a subset of M_N $\{(i, j) \mid x \le i < x + \lambda \text{ and } y \le j < y + \lambda\}$, for some integer pair (x, y). The pair (x, y) is the *location* of the tile. Notice that this definition makes sense even if (x, y) is outside the range of pairs that are nodes of M_N. A *μ-neighborhood* of a λ-tile τ located at (x, y) is the $(\lambda + 2\mu)$-tile located at $(x - \mu, y - \mu)$. The λ-partition of M_N is the collection of disjoint λ-tiles, including the one located at $(0, 0)$, which covers the nodes of M_N.

We begin by observing that MPCP on M_N can be approximated by considering the same problem on a simpler network. Let G_N be the following network: Take a λ-partition of M_N. We shall fix $\lambda = \lambda(N)$ later on. (For simplicity, we ignore the issue of N's divisibility by λ.) For each tile in the partition, G_N contains a *representative* node adjacent to the representatives of adjacent tiles. Each edge between two adjacent representatives has capacity λ. In addition to these nodes, G_N contains all the nodes of the original mesh. (We refer to these as *leaves*.) The edges of the original mesh are maintained within each λ-tile. Additional edges connect the boundary nodes of each tile to the representative of that tile. Notice that we are interested in MPCP instances on G_N where the terminals are restricted to leaves only. Thus, when we speak of MPCP on G_N, we implicitly assume this restriction to the problem.

Lemma 1. Let α, c be constants. Let A be a polynomial time MPCP algorithm, which is an $f(N)$-approximation algorithm to instances where each pair of terminals is at least $\alpha \ln^c N$ apart (for all N). Then, there is an $O(f(N))$-approximation algorithm to MPCP.

Proof Sketch. Call connections whose terminals are at least $\alpha \ln^c N$ apart *long*, and the other connections *short*. We use separate sets of colors for the long and short connections. We route the long connections using A. For the short connections, we use the fact that there is a fixed number of collections of $\beta \ln^c N$-tiles ($\beta > \alpha$ a constant), such that (i) for every short connection, both terminals are contained in a single tile in at least one of the collections; and (ii) the $3\beta \ln^c N$-neighborhoods of the tiles in a collection are all disjoint. These collections induce a partition of the short connections into classes. Each class contains the connections whose terminals are contained in a single tile of the corresponding collection. (If a connection fits into more than one collection, pick one arbitrarily.) We use a separate set of colors for each class of short connections. In each class, every $\beta \ln^c N$-tile is routed separately within its $3\beta \ln^c N$-neighborhood. (Colors are reused for each tile.) A simple argument shows that we do not increase the number of colors needed to route the connections in a tile by restricting the routes to its neighborhood. Routing the connections in each tile is done by calling A recursively. Notice that one level of recursion is enough. The next level has to deal with poly$(\ln \ln N)$-tiles, and there the problem can be solved by exhaustive search. ■

Lemma 2. Consider a ρ-partition of M_N. Given an MPCP instance on M_N, consider solutions satisfying the following additional constraint: For each color c, for each tile τ in the partition, at most one path colored c leaves τ. Then the optimal solution with this additional constraint is within a factor of $O(\rho)$ of the optimal solution without this constraint. The same holds for G_N, where we take the ρ-partitions of each of the λ-tiles.

Proof. Given a feasible solution to MPCP on M_N, we convert it to a feasible solution satisfying the additional constraint as follows: For each color c, consider the graph whose nodes are the paths colored c with two paths adjacent iff they leave the same ρ-tile. The maximum degree of a node in this graph is bounded by $8\rho - 2$, so its chromatic number is at most $8\rho - 1$. So, by replacing each color in the original solution by at most $8\rho - 1$ distinct colors (according to the coloring of the nodes of the constructed graph), we impose the additional constraint. ■

For any α, we relate the leaves of G_N to the tiles of the α-partition of $M_{\alpha N}$ in the obvious way.

Lemma 3. Consider any set of edge-disjoint paths in G_N connecting leaves. Assume that any leaf is a terminal of at most one path. Further assume that for any path, its two terminals are located in two λ-tiles whose representatives are at distance at least two apart. Consider the tiles in the α-partition of $M_{\alpha N}$ which correspond to terminals. Suppose that α is a sufficiently large constant. Then, we can choose any single node in each such tile, so that the collection of pairs of nodes in pairs of tiles that correspond to pairs of terminals connected in G_N can all be connected by edge-disjoint paths in $M_{\alpha N}$.

Proof Sketch. Choose the nodes in the α-tiles that correspond to terminals. G_N is simulated as follows. Its leaves are simulated by the tiles in the α-partition of $M_{\alpha N}$, and its representatives are simulated by the tiles in the $\alpha\lambda$-partition of $M_{\alpha N}$.

Firstly, Consider the representatives. The capacity of the connection between an $\alpha\lambda$-tile and a neighboring tile is more than sufficient. We pick one connecting edge for each boundary α-tile. In order to simulate the passing of paths through the representative, we embed in each $\alpha\lambda$-tile $\binom{4}{2}$ $\lambda \times \lambda$ crossbars connecting the boundary α-tiles of one side of the larger tile to the boundary tiles of another side, for each pair of sides. We refer to the network composed of all these crossbars plus the edges chosen to connect between $\alpha\lambda$-tiles as the *high-capacity network*.

Secondly, we embed in each $\alpha\lambda$-tile an M_λ mesh. We refer to these meshes as the *escape networks*. Thirdly, we embed in each $\alpha\lambda$-tile 4^2 $\lambda \times \lambda$ crossbars connecting the boundary α-tiles for pairs of sides (including connections from a side to itself). We refer to these crossbars as the *redirection networks*. If α is sufficiently large, then the edges of these networks can all be embedded as mutually edge-disjoint paths, and furthermore, none of the networks block the chosen terminal nodes.

The connections are routed as follows. Each terminal is connected to the escape network node in its α-tile. Then, the path connecting the corresponding terminals in G_N is simulated in $M_{\alpha N}$ using the embedded networks as follows. Consider the path in G_N. We partition it into two parts. The first part is the two portions of the path connecting the terminals to the boundaries of their respective λ-tiles. The second part is the portion of the path connecting the representatives of these λ-tiles. The first part is simulated by following the same paths on the escape networks. The second part is simulated by following the same path on the high-capacity network. Passing through representative nodes is simulated by using the crossbar structures of this network. The redirection networks are used to connect the portions of the paths in the escape networks to the portion of the path in the high-capacity network. ■

Theorem 4. For some constant α, given an $f(N)$-approximation algorithm for MPCP on G_N, one can construct an $O(f(N))$-approximation algorithm for MPCP on $M_{\alpha N}$.

Proof Sketch. Let c be a constant. Fix $\lambda = \ln^c N$.

Thus, G_N is now completely defined. We map an instance of MPCP on $M_{\alpha N}$ to an instance of MPCP on G_N as follows. A terminal in $M_{\alpha N}$ is mapped to the leaf corresponding to the tile containing the terminal in the α-partition of $M_{\alpha N}$. By Lemma 1, it is sufficient to consider instances where the requested pairs of terminals are at distance more than $2\alpha\lambda$ apart. We use the $f(N)$-approximation algorithm on G_N. By Lemma 2, we can convert the solution into an $O(f(N))$-approximation to the same problem with the additional constraint that a leaf is a terminal of at most one path in each color. In the resulting solution, each color class satisfies the conditions of Lemma 3, so the paths can be simulated on $M_{\alpha N}$. Thus we get an $O(f(N))$-approximation to the original problem, with the added constraint that each tile in the α-partition of $M_{\alpha N}$ contains at most one terminal in each color. Using Lemma 2 again, we conclude that this is also an $O(f(N))$-approximation to the original problem without the added constraint, albeit with a larger hidden constant. ■

3 Approximating the Number of Colors

For any integer c, consider the directed graph $G(c)$ derived from G as follows. The nodes of $G(c)$ are $c+2$ copies of the nodes of G. The copies are numbered 0 through $c+1$. If v is a node of G, we denote by v^j its jth copy in $G(c)$. Each of the copies 1 through c induces a graph isomorphic to G (where an undirected edge is interpreted as being usable in either direction). Each vertex in copy 0 has an edge directed to each of the corresponding vertices in copies 1 through c. Also, each vertex in copy $c+1$ has an edge directed from each of the corresponding vertices in copies 1 through c.

We map an instance of MPCP in G into an instance of integral multicommodity flow in $G(c)$ in the obvious way. A pair (s_i, t_i) in the original input is replaced by a pair (s_i^0, t_i^{c+1}).

Now, suppose that there is a feasible solution to MPCP with c colors. Then, the corresponding integral multiflow instance in $G(c)$ is feasible. Since the optimal c ranges between 1 and n, we can use a decision procedure for integral multiflow in an exhaustive search (or even binary search) algorithm to find the optimal solution to MPCP. Unfortunately, feasibility of integral multiflow is NP-complete.

Formally, the integral multiflow feasibility problem can be expressed as the following (exponential size) integer linear program, denoted $(IP[c])$:

minimize γ subject to

$$\begin{array}{ll} \sum_{j,k} f_{j,k}^i \geq 1 & \forall i \in \{1,2,\ldots,n\}; \\ \sum_{i,j} f_{j,k}^i q_j^i(e) \leq \gamma c(e) & \forall k, \forall e \in E(G); \\ f_{j,k}^i \in \{0,1\} & \forall i,j,k; \end{array}$$

where j enumerates the possible paths in G for commodity i, and q_j^i denotes the characteristic function of the jth path of commodity i. The integral multiflow problem is feasible iff there is a solution to $(IP[c])$ with $\gamma \leq 1$.

We relax the last set of conditions to $f_{j,k}^i \in [0,1]$, obtaining a linear program, which we denote $(LP[c])$. Clearly, if the optimal solution to $(IP[c])$ is ≤ 1, then so is the optimal solution to $(LP[c])$. Moreover, finding the optimal solution to $(LP[c])$ can be done in polynomial time. (See section 4 below.) Let $\gamma_{IP[c]}^*$ denote the optimal solution to $(IP[c])$, and let $\gamma_{LP[c]}^*$ denote the optimal solution to $(LP[c])$. We show the following theorem:

Theorem 5. Let $G = G_N$. If $\gamma_{LP[c]}^* \leq 1$, then there exists a constant α such that $\gamma_{IP[\alpha c]}^* \leq 1$.

Before turning to the proof of Theorem 5, we introduce some notation. Define $G = G_N$ by taking $\lambda = \ln^2 N$. Fix c, and let $\{\gamma, f\}$ be a solution to $(LP[c])$ with $\gamma \leq 1$. Let $\overline{\gamma}(f)$ denote an upper bound on the capacity utilization for edges connecting representatives, and let $\widehat{\gamma}_\ell(f)$ denote an upper bound on the maximum over all rectangles with boundary capacity at least ℓ contained in any single $\ln^2 N \times \ln^2 N$ tile, over all colors, of the amount of flow of that color leaving the rectangle divided by the total capacity of edges leaving the rectangle.

We shall need the following lemmas:

Lemma 6. Let a, b, B_0 be sufficiently large constants[1]. Let $\ell \geq 1$ be an integer. Let $B \geq B_0$, let $\overline{\gamma} \geq a \ln^{-1} N \ln^{-1} B$, and let $\widehat{\gamma} \geq b\ell^{-1} \ln^{-1} B$. Given a B^{-1}-integral solution $\{\gamma, f\}$ to $(LP[c])$ with $\overline{\gamma}(f) = \overline{\gamma}$ and $\widehat{\gamma}_\ell(f) = \widehat{\gamma}$, there exists a $\ln^{-3} B$-integral solution $\{\gamma', f'\}$ to $(LP[c])$ with $\overline{\gamma}(f') \leq (1+\epsilon)^2\overline{\gamma}$ and $\widehat{\gamma}_\ell(f') \leq (1+\epsilon)^2\widehat{\gamma}$, where $\epsilon = 1/\sqrt{\ln B}$.

Proof Sketch. Consider the following randomized rounding procedure: select path q_j^i and color k with probability $f_{j,k}^i(1+\epsilon)\ln^3 B$, and assign to it a new flow $\overline{f}_{j,k}^i = \ln^{-3} B$, independently for all i, j, k. This defines a probability space over $\ln^{-3} B$-integral (not necessarily feasible) flows. We shall show that there is a point in this probability space that is both feasible and within the claimed capacity utilization bounds. In proving this we use the following version of the Local Lemma:

Lemma 7 (Lovász Local Lemma). Let A_1, A_2, ..., A_n be events in some probability space. Let $G = (V, E)$ be the dependency graph among these events. Let x_1, x_2, ..., x_n be reals such that for all $i = 1, 2, \ldots, n$, $0 \leq x_i < 1$, and

$$\Pr[A_i] \leq x_i \prod_{(i,j)\in E} (1 - x_j).$$

Then,

$$\Pr\left[\bigwedge_{i=1}^{n} \overline{A_i}\right] \geq \prod_{i=1}^{n}(1 - x_i).$$

■

Let F_i denote the event that less than $\ln^3 B$ flow paths were picked for commodity i (and therefore the resulting solution is non-feasible). For a $\ln^2 N$ edge e, let $A_{e,k}$ denote the event that the total flow of color k over e is more than $(1+\epsilon)^2\overline{\gamma}\ln^2 N$ after rounding. For a rectangle Q with boundary capacity $\geq \ell$ contained in a $\ln^2 N \times \ln^2 N$ square, let $E_{Q,k}$ denote the event that the total flow of color k out of Q is more than $(1+\epsilon)^2\widehat{\gamma}\nabla Q$, where ∇Q denotes the total capacity of the edges leaving Q.

[1] Their value is implicit in the proof.

Using Chernoff bounds, we have:

$$\Pr[F_i] < e^{-\frac{\epsilon^2}{2(1+\epsilon)}\ln^3 B} = e^{-\ln^2 B/2(1+\epsilon)}; \qquad (1)$$

$$\Pr[A_{e,k}] < \left(\frac{e^\epsilon}{(1+\epsilon)^{(1+\epsilon)}}\right)^{\overline{\gamma}\ln^2 N \ln^3 B}; \qquad (2)$$

$$\Pr[E_{Q,k}] < \left(\frac{e^\epsilon}{(1+\epsilon)^{(1+\epsilon)}}\right)^{\widehat{\gamma}\nabla Q \ln^3 B}. \qquad (3)$$

The dependencies among the various types of events are summarized as follows. An event F_i depends on $\leq B \cdot \frac{4N^2}{\ln^4 N}$ events $A_{e,k}$, and, for every value of ∇Q, on $\leq 2(\nabla Q)^3 \cdot B$ events $E_{Q,k}$. It is independent of all other events F_j. An event $A_{e,k}$ depends on $\leq B \cdot \overline{\gamma}\ln^2 N$ events F_i, on $\leq \frac{4N^2}{\ln^4 N}$ events $A_{e',k}$, and, for every value of ∇Q, on $\leq 2(\nabla Q)^3 \cdot B \cdot \overline{\gamma}\ln^2 N$ events $E_{Q,k}$. An event $E_{Q,k}$ depends on $\leq B \cdot \widehat{\gamma}\nabla Q$ events F_i, on $\leq \frac{4N^2}{\ln^4 N}$ events $A_{e,k}$, and, for every value of $\nabla Q'$, on $\leq 2(\nabla Q')^3 \cdot B \cdot \widehat{\gamma}\nabla Q$ events $E_{Q',k}$.

We define for all i, $x_{F_i} = x_F = B^{-2}$; for all e, k, $x_{A_{e,k}} = x_A = B^{-1}N^{-2}$; for all Q, k, $x_{E_{Q,k}} = x_Q = B^{-2}e^{-\nabla Q}$.

Using elementary estimates, it is a simple matter to verify that the conditions for applying the Local Lemma hold. We demonstrate this for the events F_i:

$$\begin{aligned}
&x_F\,(1 - x_A)^{\frac{4BN^2}{\ln^4 N}} \prod_{\nabla Q=\ell}^{4\ln^2 N} (1 - x_Q)^{2(\nabla Q)^3 B} \\
\geq\;& B^{-2} \cdot e^{-5/\ln^4 N} \prod_{i=1}^{4\ln^2 N} e^{-(3i^3/Be^i)} && (4)\\
=\;& B^{-2} \cdot e^{-5/\ln^4 N} \cdot e^{-3/B \sum_{i=1}^{4\ln^2 N} i^3/e^i} && (5)\\
\geq\;& B^{-2} \cdot \left(1 - \frac{5}{\ln^4 N}\right) \cdot \left(1 - \frac{18}{B}\right) && (6)\\
\geq\;& B^{-\frac{\ln B}{2(1+\epsilon)}} && (7)\\
\geq\;& \Pr[F_i], && (8)
\end{aligned}$$

where Inequality 4 follows from $(1 - x^{-1})^{x-1} \geq e^{-1}$, and Inequality 6 follows from $e^{-x} \geq 1 - x$ and $6 \geq \sum_{i=1}^{\infty} i^3/e^i$.

Similar calculations hold for the events $A_{e,k}$ and $E_{Q,k}$. Thus, we can apply the Lovász Local Lemma and conclude that there is a point in the sample space of the above randomized rounding procedure at which none of the F_i's, $A_{e,k}$'s, $E_{Q,k}$'s occur. ■

Lemma 8. Given a B^{-1}-integral solution $\{\gamma, f\}$ to $(LP[c])$, one can find in polynomial time a B^{-1}-integral solution $\{\gamma', f'\}$ to $(LP[c])$ with $\gamma' \leq \max\{\overline{\gamma}(f), \widehat{\gamma}_1(f)\}$.

Proof Sketch. We can convert $\{\gamma, f\}$ into $\{\gamma', f'\}$ by solving the escape problem for the terminals in each $\ln^2 N \times \ln^2 N$ colored tile. This is an s–t flow problem, and the cut conditions are satisfied, taking the edge capacities to be $\widehat{\gamma}_1(f)$. Routing among the representatives is done as in $\{\gamma, f\}$. ■

Proof of Theorem 5. Let c be such that $\gamma^*_{LP[c]} \leq 1$. Let B_0 be a constant. We convert the optimal solution $\{\gamma^*, f^*\}$ to $(LP[c])$ into a feasible solution $\{\gamma, f\}$ to $(LP[4B_0c])$ satisfying the following conditions: (i) $\gamma^*_{LP[c]} \leq \frac{1}{2B_0}$; (ii) All flow paths carry the same amount of flow $f_0 \geq 2^{-N^{O(1)}}$. The bound on f_0 follows from the fact that $\{\gamma, f\}$ is computed in polynomial time. To meet the other conditions, execute the following procedure: Duplicate each color and the paths that use it $4B_0$ times, giving each flow path a fraction of $(4B_0)^{-1}$ of its original flow. Now, divide each flow path into as many identical paths as needed, so that all paths have flow within a factor of 2 of each other. Finally, assign to all paths the largest flow of a path (which is at most twice what they had before).

Now, we repeatedly apply Lemma 6 with $\ell = 1$. We begin with $B = f_0^{-1}$ and get a $\ln^{-3} B$-integral flow. Applying the lemma to this flow, we get a $(3 \ln \ln B)^{-3}$-integral flow. We can repeat applying the lemma as long as the value B satisfies: $B \geq e^{a/\widehat{\gamma}} = e^{2aB_0}$. This takes $O(\log^* N)$ iterations.

At this point, for each rectangle Q with boundary capacity B_0, for each color k, there are at most e^{2aB_0} fractional paths of color k that leave Q (see below the analysis that bounds the increase in $\widehat{\gamma}$). By duplicating each color a (large) constant number of times, we can partition the paths among the duplicates so that at most a single path leaves each such rectangle Q.

We now proceed applying Lemma 6 with $\ell = B_0$. After $O(1)$ iterations we end up with a $(B_0)^{-1}$-integral solution. (B_0 is chosen large enough so that this is possible). Our initial flow $\{\gamma, f\}$ has $\overline{\gamma}(f)$ and $\widehat{\gamma}_{B_0}(f)$ at most $(2B_0)^{-1}$. Through all the $O(\log^* N)$ iterations these values increase by a factor of at most

$$\prod_{i=1}^{t} \left(1 + \frac{1}{\sqrt{\ln B_i}}\right)^2,$$

where the B_is are given by the recurrence $B_i = e^{\sqrt[3]{B_{i-1}}}$, and $t = O(\log^* N)$. For a sufficiently large choice of B_0, this product is at most 2. Therefore, in the resulting $(B_0)^{-1}$-integral solution $\{\gamma', f'\}$, both $\overline{\gamma}(f')$, $\widehat{\gamma}_{B_0}(f')$ are at most $(B_0)^{-1}$ (and the conditions regarding $\overline{\gamma}$, $\widehat{\gamma}$ in Lemma 6 are satisfied for all iterations). We now pick for each commodity a single path among the B_0 possible paths and give it weight 1. Let's denote the resulting solution by $\{\gamma'', f''\}$. Obviously, $\overline{\gamma}(f'') \leq 1$ and $\widehat{\gamma}_{B_0}(f'') \leq 1$. Since we are assuming that there is at most one terminal in each rectangle with boundary capacity B_0 or less, it is also true that $\widehat{\gamma}_1(f'') \leq 1$. By Corollary 8 there is an integral solution $\{\gamma_{int}, f_{int}\}$ such that $\gamma_{int} \leq 1$. ■

4 Constructing the Paths

There are two non-algorithmic arguments in the above proof. Firstly, we may begin with a collection of paths of exponential size (since the path with least flow may carry an exponentially small amount of flow). This is easily amended by using, instead of a linear programming algorithm, an approximation scheme for multicommodity flow based on Lagrangian relaxations (e.g. [13], which can be used since we have small integral capacities). We are guaranteed to get a near optimal solution (sufficient for our purposes) with a polynomial number of flow paths carrying the same amount of flow each.

Secondly, the use of the Local Lemma in the proof of Lemma 6 is non-algorithmic. Beck [10] proposed an algorithmic version of the Local Lemma that works in certain cases. His ideas do not seem to work here due to the complicated structure of the dependency graph, which has events of varying degrees and probabilities. Nevertheless, we are able to apply Beck's results together with some additional ideas to obtain a weaker algorithmic result (for simplicity, we deal with the case where the number of connections n is bounded by a polynomial in N):

Theorem 9. There is a poly$(\log\log N)$ approximation algorithm for MPCP on the mesh.

Proof Sketch. The algorithm is similar to the procedure described in the proof of Theorem 5. We show here how to implement a single iteration of the rounding procedure, and point out the places where this procedure has to be modified significantly.

In what follows, $c_1, c_2, c_3, c_4, c_6, c_8, c_9, c_{10}, c_{11}$, and c_{12} are constants.

Notice that there is nothing magical about the choice of $\ln^2 N$ as the λ that defines G_N. In fact, even a smaller λ can be used in the previous argument. For the algorithm we need to use larger $\ln^{c_1} N$-tiles, for c_1 sufficiently large. Further notice that we may assume that initially $B \le c_2 \ln N$, for c_2 sufficiently large. If B is larger, then randomized rounding to a $(c_2 \ln N)^{-1}$-integral solution succeeds with high probability, so we do not need to use the Local Lemma.

The algorithm we suggest for a single rounding iteration proceeds in several phases. The purpose of the first phase is to eliminate large degree nodes from the dependency graph. Then come phases which partition the dependency graph into smaller connected components. Finally, the last phase solves the remaining problem in each (small) connected component.

A path is either *undecided*, or *chosen*, or *removed*, or *passed*. Initially, all paths are undecided. During a phase we examine the undecided paths one at a time. We change the status of each undecided path to one of the other three categories. At the end of a phase, we examine each commodity. If we chose too few paths of commodity i, then we change the status of all the paths of commodity i to *passed*. To start a new phase, all passed paths become *undecided*.

We first discuss the case that an *undecided* path is *passed* during phase I. Let m be the number of paths of color k using the edge e that have been considered so far. The pair (e, k) is *critical* iff at least $\frac{m}{B}(1+\epsilon)\ln^3 B + \epsilon\overline{\gamma}\ln^{c_1} N \ln^3 B$ of these paths have been chosen. Notice that the first term is simply the expectation. Let $A_{e,k}$ denote the event that (e, k) becomes critical at some point.

Let m be the number of paths of color k that have a terminal in rectangle Q that have been considered so far. The pair (Q, k) is *critical* iff at least $\frac{m}{B}(1+\epsilon)\ln^3 B + \epsilon\widehat{\gamma}\nabla Q \ln^3 B$ of these paths have been chosen. Again, the first term is the expectation. Let $E_{Q,k}$ denote the event that (Q, k) becomes critical at some point.

A path is passed if when we consider it, the path participates in a critical event. If a path is not passed, it is *chosen* with probability $B^{-1}(1+\epsilon)\ln^3 B$, and otherwise it is removed. Using Chernoff bounds and some elementary estimates, we get $\Pr[A_{e,k}] \le e^{-\frac{1}{4}\overline{\gamma}\ln^{c_1} N \ln^3 B}$, and $\Pr[E_{Q,k}] \le e^{-\frac{1}{4}\widehat{\gamma}\nabla Q \ln^2 B}$.

To complete phase I, we check all commodities. If less than $\ln^3 B$ paths have been chosen of commodity i, then all paths of that commodity become *passed*. Otherwise, all the paths of commodity i that have not been chosen become *removed*. Let F_i denote the event that none of the paths of commodity i were passed initially, but that fewer than $\ln^3 B$ of them were chosen (and therefore they were all eventually passed). Clearly, $\Pr[F_i] < e^{-\ln^2 B/2(1+\epsilon)}$.

The purpose of phase I is to eliminate the pairs (e, k) from further consideration. We show that w.h.p. no $\ln^{c_1} N$ capacity edge carries more than $\overline{\gamma}\ln^{c_1} N$ passed paths. Even if all these paths are turned into $\ln^{-3} B$-integral paths, the added congestion will be negligible. Therefore, in the following phases we may ignore the edges (e, k).

We say that a path *failed* if one of the paths of its commodity i participated in a critical pair, or if F_i occured. Let $f_{e,k}$ denote the number of failed paths that use (e, k). Clearly, the number of passed paths that use (e, k) is bounded by $f_{e,k}$. We estimate $\Pr[f_{e,k} > \overline{\gamma}\ln^{c_1} N]$ as follows. Summing over all events that may cause a path to fail, we get that its failure probability is at most B^{-c_3}, where c_3 can be made arbitrarily large by taking B_0 arbitrarily large. The events that paths fail are not independent. However, w.h.p. only the pairs (Q, k), $\nabla Q \leq c_4 \ln N$ have a chance of becoming critical. The probability that other pairs become critical can be made smaller than any polynomial in N by taking a large enough c_4. Therefore, each path is independent of all but at most $\ll \ln^{c_6} N$ other paths (recall that $B \leq c_2 \ln N$.) We use Chernoff-like bounds for events with limited dependency to get $\Pr[f_{e,k} > \overline{\gamma} \ln^{c_1} N] \ll N^{-c_8}$. ($c_1$ must be larger than c_6.)

Phase I succeeds if none of the edges (e, k) carries more than $\overline{\gamma} \ln^{c_1} N$ passed paths. The above analysis shows that phase I succeeds w.h.p. (If it fails, we redo it.) In phase II, we handle paths that were passed from phase I. The phase is identical to phase I, except that we do not check the pairs (e, k). The probability that a pair becomes critical is estimated as above. Let $f_{Q,k}$ denote the number of failed paths with an endpoint in (Q, k). We want to estimate $\Pr[f_{Q,k} > \widehat{\gamma} \nabla Q]$. We consider two cases:

Case 1: $\nabla Q \leq B^{c_9}$, for a constant $c_9 < c_3 - 1$. By summing up the failure probabilities of each of the paths leaving Q, we get that the desired probability is bounded by $B^{-c_{10}}$, where c_{10} depends on $c_3 - c_9$.

Case 2: $\nabla Q > B^{c_9}$. We analyze the contribution of each rectangle size separately, as well as that of the events F_i. Summing up these estimates gives $\Pr[f_{Q,k} > \widehat{\gamma} \nabla Q] < exp(-(\nabla Q)^{\delta})$, where $\delta = 1/c_9$.

For $B = c_2 \ln N$, the above analysis shows that the probability that a pair (Q, k) needs to be considered after phase II is at most $\ln^{-c_{10}} N$, where c_{10} is determined by c_3. The maximum degree of a pair in the dependency graph is $\ln^{c_{11}} N$, where c_{11} is determined by c_1. (Notice that the survivals of two pairs are independent if their distance in the original dependency graph of Lemma 6 is greater than two.) Therefore, by taking c_3 sufficiently large compared with c_1, Beck's analysis can be applied to show that phase II leaves a dependency graph with connected components of size at most $\ln^{c_{12}} N$ each, where $c_{12} > c_{11}$. Now, randomized rounding to a $\ln^{-3} B$-integral solution succeeds in each connected component w.h.p.

The above algorithm cannot be iterated for smaller values of B. Therefore, after getting the poly$(\ln \ln N)$-integral flow, we adjust the fractional solution. Arguing as in the proof of Lemma 2, we can partition each color class into poly$(\ln \ln N)$ color classes, leaving at most a single path flowing out of each pair (Q, k) with $\nabla Q \leq (c_{10} \ln \ln N)^{c_9}$. As a result, we can ignore these pairs in future iterations.

The next iterations proceed as follows. Phase I and phase II are done as above. Notice that since we are dealing with pairs (Q, k) with $\nabla Q > (c_{10} \ln \ln N)^{c_9}$, the probability that such a pair survives phase II is at most $\ln^{-c_{10}} N$. Therefore, phase II leaves connected components of size at most $\ln^{c_{12}} N$. In each connected component, we execute phase III, similar to the previous ones. By the above analysis, any pair with boundary capacity greater than $(c_{10} \ln \ln N)^{c_9}$ has a chance of at most $\ln^{-c_{10}} N$ of surviving. By taking $c_{10} > c_{12}$, we get that w.h.p. no such pair survives. Finally, the paths passed from phase III can be chosen arbitrarily. The congestion is increased through all phases by a factor of $1 + O(\epsilon)$ at worst. ■

References

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

[2] N. Alon. A parallel algorithmic version of the local lemma. *Random Structures and Algorithms*, 2(4):367–378, 1991.

[3] N. Alon and J.H. Spencer. *The Probabilistic Method*. Wiley, 1992.

[4] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts. On-line load balancing with applications to machine scheduling and virtual circuit routing. In *Proc. 23rd ACM STOC*, 1993.

[5] Y. Aumann and Y. Rabani. Improved bounds for all optical routing. In *Proc. 6th ACM-SIAM SODA*, 1995.

[6] B. Awerbuch, Y. Azar, and S. Plotkin. Throughput competitive on-line routing. In *Proc. 34th IEEE FOCS*, 1993.

[7] B. Awerbuch, Y. Bartal, A. Fiat, and A. Rosén. Competitive non-preemptive call control. In *Proc. 5th ACM-SIAM SODA*, 1994.

[8] B. Awerbuch, R. Gawlick, F.T. Leighton, and Y. Rabani. On-line admission control and circuit routing for high performance computing and communication. In *Proc. 35th IEEE FOCS*, 1994.

[9] Y. Bartal and S. Leonardi. Private communication, March 1996.

[10] J. Beck. An algorithmic approach to the Lovász local lemma. *Random Structures and Algorithms*, 2(4):343–365, 1991.

[11] P. Erdős and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. In A. Hajnal et al., eds., *Infinite and Finite Sets*. North Holland, 1975, pages 609–628.

[12] N. Garg, V.V. Vazirani, and M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees, with applications to matching and set cover. In *Proc. ICALP*, 1993.

[13] P. Klein, S. Plotkin, C. Stein, and É. Tardos. Faster approximation algorithms for the unit capacity concurrent flow problem with applications to routing and finding sparse cuts. *SIAM J. Comput.*, 23(3):466–487, 1994.

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

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

[16] B. Korte, L. Lovász, H.J. Promel, and A. Schrijver, editors. *Paths, flows, and VLSI-layout*. Springer-Verlag. 1990.

[17] F.T. Leighton and B.M. Maggs. Fast algorithms for finding O(congestion + dilation) packet routing schedules. In *Proc. 28th Hawaii International Conference on System Sciences*, January 1995, vol 2, pages 555–563.

[18] F.T. Leighton, B.M. Maggs, and S.B. Rao. Packet routing and job-shop scheduling in O(congestion + dilation) steps. *Combinatorica*, 14(2):167–186, 1994.

[19] M. Middendorf and F. Pfeiffer. On the complexity of the disjoint paths problem. *Combinatorica*, 13(1):97–107, 1993.

[20] P. Raghavan. Probabilistic construction of deterministic algorithms: approximating packing integer programs. *Journal of Computer and Systems Sciences*, 37:130–143, 1988.

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

[22] P. Raghavan and E. Upfal. Efficient routing in all optical networks. In *Proc. 26th ACM STOC*, 1994.

[23] A. Srinivasan. Improved approximations of packing and covering problems. In *Proc. 27th ACM STOC*, 1995.

[24] A. Srinivasan. An extension of the Lovász local lemma, and its applications to integer programming. In *Proc. 7th ACM-SIAM SODA*, 1996.