On Approximating Node-Disjoint Paths in Grids

Julia Chuzhoy[*1] **and David H. K. Kim**[†2]

1 **Toyota Technological Institute at Chicago**
6045 S. Kenwood Ave., Chicago, Illinois 60637, USA
`cjulia@ttic.edu`

2 **Department of Computer Science, University of Chicago**
1100 East 58th Street, Chicago, Illinois 60615, USA
`hongk@cs.uchicago.edu`

Abstract

In the Node-Disjoint Paths (NDP) problem, the input is an undirected n-vertex graph G, and a collection $\{(s_1,t_1),\ldots,(s_k,t_k)\}$ of pairs of vertices called *demand pairs*. The goal is to route the largest possible number of the demand pairs (s_i,t_i), by selecting a path connecting each such pair, so that the resulting paths are node-disjoint. NDP is one of the most basic and extensively studied routing problems. Unfortunately, its approximability is far from being well-understood: the best current upper bound of $O(\sqrt{n})$ is achieved via a simple greedy algorithm, while the best current lower bound on its approximability is $\Omega(\log^{1/2-\delta} n)$ for any constant δ. Even for seemingly simpler special cases, such as planar graphs, and even grid graphs, no better approximation algorithms are currently known. A major reason for this impasse is that the standard technique for designing approximation algorithms for routing problems is LP-rounding of the standard multicommodity flow relaxation of the problem, whose integrality gap for NDP is $\Omega(\sqrt{n})$ even on grid graphs.

Our main result is an $O(n^{1/4}\cdot\log n)$-approximation algorithm for NDP on grids. We distinguish between demand pairs with both vertices close to the grid boundary, and pairs where at least one of the two vertices is far from the grid boundary. Our algorithm shows that when all demand pairs are of the latter type, the integrality gap of the multicommodity flow LP-relaxation is at most $O(n^{1/4}\cdot\log n)$, and we deal with demand pairs of the former type by other methods. We complement our upper bounds by proving that NDP is APX-hard on grid graphs.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Node-disjoint paths, approximation algorithms, routing and layout

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2015.187

1 Introduction

In the classical Node-Disjoint Paths (NDP) problem, the input is an undirected n-vertex graph $G=(V,E)$, and a collection $\{(s_1,t_1),\ldots,(s_k,t_k)\}$ of pairs of vertices, called *source-destination*, or *demand*, pairs, that we would like to route. In order to route a pair (s_i,t_i), we need to select some path P connecting s_i to t_i. The goal is to route the largest possible number of the demand pairs on node-disjoint paths: that is, every vertex of G may participate in at most one path in the solution.

NDP is one of the most basic and extensively studied routing problems. When the number of the demand pairs k is bounded by a constant, Robertson and Seymour [27, 29] have

[*] Supported in part by NSF grant CCF-1318242.
[†] Supported in part by NSF grant CCF-1318242.

18th Int'l Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'15) / 19th Int'l Workshop on Randomization and Computation (RANDOM'15).
Editors: Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim; pp. 187–211
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

shown an efficient algorithm for the problem, as part of their seminal Graph Minors project. However, when k is a part of the input, the problem is known to be NP-hard [17]. Even though the NDP problem, together with its many variants, has been extensively studied, its approximability is still poorly understood. The best currently known upper bound on the approximation factor is $O(\sqrt{n})$ [22], achieved by the following simple greedy algorithm: start with graph G and an empty solution. While G contains any path connecting any demand pair, choose the shortest such path P, add P to the solution, and delete all vertices of P from G. Surprisingly, this elementary algorithm is the best currently known approximation algorithm for NDP, even for restricted special cases of the problem, where the input graph G is a planar graph, or even just a grid. On the negative side, it is known that there is no $O(\log^{1/2-\delta} n)$-approximation algorithm for NDP for any constant δ, unless $\mathsf{NP} \subseteq \mathsf{ZPTIME}(n^{\mathrm{poly}\log n})$ [5, 4]. Perhaps the biggest obstacle in breaking the $O(\sqrt{n})$-approximation barrier for the problem is the fact that the integrality gap of the standard multicommodity flow LP-relaxation for NDP is $\Omega(\sqrt{n})$, even in grid graphs. In the LP-relaxation, instead of connecting the demand pairs by paths, we try to send as much flow as possible between the demand pairs, subject to the constraint that each vertex carries at most one flow unit. The $O(\sqrt{n})$-approximation greedy algorithm described above can be cast as an LP-rounding algorithm for the multicommodity flow LP, and therefore, the integrality gap of the LP is $\Theta(\sqrt{n})$. So far, rounding this LP relaxation has been the main method used in designing approximation algorithms for a variety of routing problems, and it appears that new techniques are needed in order to improve the $O(\sqrt{n})$-approximation factor for NDP.

In this paper we break the $O(\sqrt{n})$-barrier on the approximation factor for NDP on grid graphs [1], by providing an $O(n^{1/4} \cdot \log n)$-approximation algorithm. Our algorithm distinguishes between two types of demand pairs: an (s_i, t_i) pair is *bad* if both s_i and t_i are close to the grid boundary, and it is good otherwise. Interestingly, the standard integrality gap examples for the multicommodity flow LP relaxation usually involve a grid graph, and bad demand pairs. Our algorithm deals with bad and good demand pairs separately, and in particular it shows that if all demand pairs are good, then the integrality gap of the LP relaxation becomes $O(n^{1/4} \cdot \log n)$ (but unfortunately it still remains polynomial in n - see Section 6). We complement these results by showing that NDP is APX-hard even on grid graphs. We believe that understanding the approximability of NDP on grid graphs is an important first step towards understanding the approximability of the NDP problem in general, as grids seem to be the simplest graphs, for which the approximability of the NDP problem is widely open, and the integrality gap of the multicommodity flow LP is $\Omega(\sqrt{n})$. We hope that some of the techniques introduced in this paper will be helpful in breaking the $O(\sqrt{n})$-approximation barrier in general planar graphs.

NDP in grid graphs has been studied in the past, often in the context of VLSI layout. Aggarwal, Kleinberg and Williamson [1] consider a special case, where the set of the demand pairs is a permutation — that is, every vertex of the grid participates in exactly one demand pair. They show that for any permutation, one can route $\Omega(\sqrt{n}/\log n)$ demand pairs. They also show that with spacing d, every permutation contains a set of $\Omega(\sqrt{nd}/\log n)$ pairs that can be routed on node-disjoint paths. Our algorithm for routing on grids is inspired by their work.

Cutler and Shiloach [16] studied NDP in grids in the following three settings. They assume that all source vertices appear on the top row R_1 of the grid, and all destination

[1] Since n denotes, by convention, the number of vertices in the input graph, the size of the grid is $(\sqrt{n} \times \sqrt{n})$.

vertices appear on some other row R_ℓ of the grid, sufficiently far from the top and the bottom rows (for example, $\ell = \lceil n/2 \rceil$). In the packed-packed setting, the sources are a set of k consecutive vertices of R_1, and the destinations are a set of k consecutive vertices of R_ℓ. They show a necessary and a sufficient condition for when all demand pairs can be routed in the packed-packed instance. The second setting is the packed-spaced setting. Here, the sources are again a set of k consecutive vertices of R_1, but the distance between every consecutive pair of the destination vertices on R_ℓ is at least d. For this setting, the authors show that if $d \geq k$, then all demand pairs can be routed. We extend their algorithm to a more general setting, where the destination vertices may appear anywhere in the grid, as long as the distance between any pair of the destination vertices, and any destination vertex and the boundary of the grid, is at least $\Omega(k)$. This extension of the algorithm of [16] is used as a basic building block in both our algorithm, and the APX-hardness proof. We note that Robertson and Seymour [28] provided sufficient conditions for the existence of node-disjoint routing of a given set of demand pairs in the more general setting of graphs drawn on surfaces, and they provide an algorithm whose running time is $\text{poly}(n) \cdot f(k)$ for finding the routing, where $f(k)$ is at least exponential in k. Their result implies the existence of the routing on grids, when the destination vertices are sufficiently spaced from each other and from the grid boundaries. However, we are not aware of an algorithm for finding the routing, whose running time is polynomial in n and k, and we provide such an algorithm here. The third setting studied by Cutler and Shiloach is the spaced-spaced setting, where the distance between any pair of source vertices, and any pair of destination vertices is at least d. The authors note that they could not come up with a better algorithm for this setting, than the one provided for the packed-spaced case.

Other Related Work

A problem closely related to NPD is the Edge-Disjoint Paths (EDP) problem. It is defined similarly, except that now the paths chosen to the solution are allowed to share vertices, and are only required to be edge-disjoint. It is easy to show, by using a line graph of the EDP instance, that NDP is more general than EDP. The approximability status of EDP is very similar to that of NDP: there is an $O(\sqrt{n})$-approximation algorithm [13], and it is known that there is no $O(\log^{1/2-\delta} n)$-approximation algorithm for any constant δ, unless $\mathsf{NP} \subseteq \mathsf{ZPTIME}(n^{\text{poly}\log n})$ [5, 4]. As in the NDP problem, we can use the standard multicommodity flow LP-relaxation of the problem, in order to obtain the $O(\sqrt{n})$-approximation algorithm, and the integrality gap of the LP-relaxation is $\Omega(\sqrt{n})$ even on planar graphs. However, for even-degree planar graphs, Kleinberg [19], building on the work of Chekuri, Khanna and Shepherd [12, 11], has shown an $O(\log^2 n)$-approximation LP-rounding algorithm. Aumann and Rabani [8] showed an $O(\log^2 n)$-approximation algorithm for EDP on grid graphs, and Kleinberg and Tardos [21, 20] showed $O(\log n)$-approximation algorithms for wider classes of nearly-Eulerian uniformly high-diameter planar graphs, and nearly-Eulerian densely embedded graphs. Recently, Kawarabayashi and Kobayashi [18] gave an $O(\log n)$-approximation algorithm for EDP when the input graph is either 4-edge-connected planar or Eulerian planar. It appears that the restriction of the graph G to be Eulerian, or near-Eulerian, makes the EDP problem significantly simpler, and in particular improves the integrality gap of the LP-relaxation. The analogue of the grid graph for the EDP problem is the wall graph (see Figure 1): the integrality gap of the standard LP relaxation for EDP on wall graphs is $\Omega(\sqrt{n})$, and to the best of our knowledge, no better than $O(\sqrt{n})$-approximation algorithm for EDP on walls is known. Our $O(n^{1/4} \cdot \log n)$-approximation algorithm for NDP on grids can be extended to the EDP problem on wall graphs (see Section 7).

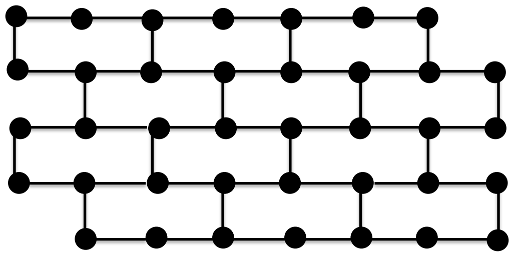

Figure 1 A wall graph.

A variation of the NPD and EDP problems, where small congestion is allowed, has been a subject of extensive study. In the NDP with congestion (NDPwC) problem, the input is the same as in the NDP problem, and we are additionally given a non-negative integer c. The goal is to route as many of the demand pairs as possible with congestion at most c: that is, every vertex may participate in at most c paths in the solution. EDP with Congestion (EDPwC) is defined similarly, except that now the congestion bound is imposed on edges and not vertices. The classical randomized rounding technique of Raghavan and Thompson [25] gives a constant-factor approximation for both problems, if the congestion c is allowed to be as high as $\Theta(\log n/\log\log n)$. A recent line of work [12, 24, 3, 26, 14, 15, 10, 9] has lead to an $O(\text{poly}\log k)$-approximation for both NDPwC and EDPwC problems, with congestion $c = 2$. For planar graphs, a constant-factor approximation with congestion 2 is known [30]. All these algorithms perform LP-rounding of the standard multicommodity flow LP-relaxation of the problem.

Organization

We start with Preliminaries in Section 2, and show a generalization of the algorithm of Cutler and Shiloah [16] for routing with well-separated destinations in Section 3. In Section 4 we provide an $O(n^{1/4} \cdot \log n)$-approximation algorithm for NDP on grids, and we provide the APX-hardness proof in Section 5. We discuss the integrality gap of the multicommodity flow LP-relaxation when all terminals are far from the grid boundary in Section 6, and we sketch the extension of our $O(n^{1/4} \log n)$-approximation algorithm to EDP on wall graphs in Section 7.

2 Preliminaries

We consider the NDP problem in two-dimensional grids: The input is an $(N \times N)$-grid graph $G = (V, E)$, and a collection $\mathcal{M} = \{(s_1, t_1), \dots, (s_k, t_k)\}$ of pairs of vertices, called demand, or source-destination, pairs. The goal is to find a largest cardinality collection $\mathcal{P}$ of paths, where each path in $\mathcal{P}$ connects some demand pair (s_i, t_i), and every vertex of G participates in at most one path in $\mathcal{P}$. The vertices in the set $\{s_1, t_1, \dots, s_k, t_k\}$ are called *terminals*. By convention, we denote $n = |V|$, so $n = N^2$.

We assume that the grid rows are indexed $R_1, \dots, R_N$ in the top-to-bottom order, and the columns are indexed $C_1, \dots, C_N$ in the left-to-right order. We denote by $v(i, j)$ the unique vertex in $R_i \cap C_j$. Given a vertex $v \in V$, let $\text{col}(v)$ denote the column, and $\text{row}(v)$ denote the row in which v lies. The boundary of the grid is $\Gamma(G) = R_1 \cup R_N \cup C_1 \cup C_N$. We call R_1, R_N, C_1, C_N the *boundary edges* of the grid. Given any integers $1 \le i \le i' \le N$, $1 \le j \le j' \le N$, we denote by $G[i : i', j : j']$ the sub-graph of G, induced by the set

$\{v(i'',j'') \mid i \le i'' \le i', j \le j'' \le j'\}$ of vertices. We sometimes say that $G[i:i', j:j']$ is the sub-grid of G, spanned by rows $R_i, \ldots, R_{i'}$ and columns $C_j, \ldots, C_{j'}$.

Given a path P in G, and a set S of vertices of G, we say that P is *internally disjoint* from S, if no vertex of S serves as an inner vertex of P. We will use the following simple observation.

▶ **Observation 1.** *Let G be a $(h \times w)$-grid, with $w, h > 2$, and let $k \le \min\{w-2, h-2\}$ be an integer. Then for any pair L, L' of opposing boundary edges of G, for any pair $S \subseteq V(L)$, $T \subseteq V(L')$ of vertex subsets on these boundary edges, with $|S| = |T| = k$, there is a set $\mathcal{P}$ of k node-disjoint paths, connecting the vertices of S to the vertices of T in G, such that all paths in $\mathcal{P}$ are internally disjoint from $V(L \cup L')$. Moreover, the path set $\mathcal{P}$ can be found efficiently.*

Proof. Let G' be the sub-graph of G, obtained by deleting all vertices of $(L \cup L') \setminus (S \cup T)$ from G. It is enough to show that there is a set $\mathcal{P}$ of k disjoint paths connecting the vertices of S to the vertices of T in G'.

Assume without loss of generality that L is the top and L' is the bottom boundary edge of G. Assume for contradiction that such a set $\mathcal{P}$ of paths does not exist. Then from Menger's theorem, there is a set Z of at most $k-1$ vertices, such that in $G' \setminus Z$, there is no path from a vertex of $S \setminus Z$ to a vertex of $T \setminus Z$. However, the vertices of S lie on k distinct columns of G, so at least one such column, say C, does not contain a vertex of Z. Similarly, there is some column C' of G that contains a vertex of T, and $V(C') \cap Z = \emptyset$. Finally, since there are at least $k+2$ rows in G, there is some row $R \ne R_1, R_h$, that contains no vertex of Z. Altogether, $(C \cup R \cup C') \cap G'$ lie in the same connected component of $G' \setminus Z$, and this connected component contains a vertex of S and a vertex of T, a contradiction. The set $\mathcal{P}$ of paths can be found efficiently by computing the maximum single-commodity flow between the vertices of S and the vertices of T in G', and using the integrality of flow. ◀

Consider the input grid graph G. The L_∞-distance between two vertices $v(i,j)$ and $v(i',j')$ is defined as $d_\infty(v(i,j), v(i',j')) = \max(|i-i'|, |j-j'|)$. The distance between a set $S \subseteq V(G)$ of vertices and a vertex $v \in V(G)$ is $d_\infty(v,S) = \min_{u \in S}\{d_\infty(v,u)\}$.

Multicommodity Flow LP Relaxation

For each demand pair $(s_i, t_i) \in \mathcal{M}$, let $\mathcal{P}_i$ be the set of all paths connecting s_i to t_i in G, and let $\mathcal{P} = \bigcup_{i=1}^k \mathcal{P}_i$. In order to define the multicommodity flow LP-relaxation of NDP, every path $P \in \mathcal{P}$ is assigned a variable $f(P)$ representing the amount of flow that is sent on P, and for each demand pair (s_i, t_i), we introduce variable x_i, whose value is the total amount of flow sent from s_i to t_i. The LP-relaxation is then defined as follows.

$$\begin{array}{lll}
\text{(LP-flow)} & \max & \sum_{i=1}^k x_i \\
& \text{s.t.} & \sum_{P \in \mathcal{P}_i} f(P) = x_i \quad \forall 1 \le i \le k \\
& & \sum_{P: v \in P} f(P) \le 1 \quad \forall v \in V \\
& & f(P) \ge 0 \qquad\quad \forall 1 \le i \le k, \forall P \in \mathcal{P}_i
\end{array}$$

Even though this LP-relaxation has exponentially many variables, it can be efficiently solved by standard techniques, e.g. by using an equivalent polynomial-size edge-based formulation.

It is well known that the integrality gap of (LP-flow) is $\Omega(\sqrt{n})$ even in grid graphs. Indeed, let G be an $(N \times N)$-grid, and let $k = N-2$. We let the sources $s_1, \ldots, s_k$ appear

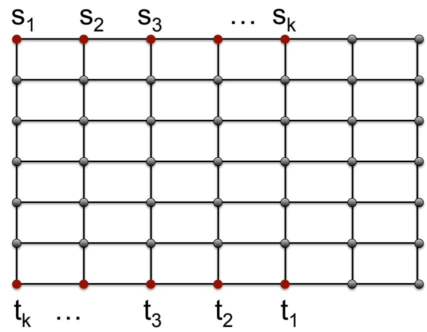

Figure 2 Integrality gap example.

consecutively on row R_1, starting from $v(1,1)$ in this order, and the destinations appear consecutively on row R_N starting from $v(N,1)$, in the opposite order: $t_k, \ldots, t_1$ (see Figure 2). It is easy to see that there is a solution to (LP-flow) of value $k/3 = \Omega(N)$: for each pair (s_i, t_i), we send $1/3$ flow unit on the path P_i, where P_i is an s_i–t_i path lying in the union of columns C_i, C_{N-i-1} and row $R_i + 1$. On the other hand, it is easy to see that the value of any integral solution is 1, since any pair of paths connecting the demand pairs have to cross. Since the number of vertices in G is $n = N^2$, this gives a lower bound of $\Omega(\sqrt{n})$ on the integrality gap of (LP-flow).

3 Routing with Well-Separated Destinations

In this section we generalize the results of Cutler and Shiloach [16], by proving the following theorem.

▶ **Theorem 2.** *Let H be the $(N \times N)$-grid, and let $\mathcal{M} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ be a set of $k \geq 4$ demand pairs in H, such that: (i) $s_1, \ldots, s_k$ are all distinct, and they appear on the first row of H; (ii) for all $1 \leq i \neq j \leq k$, $d_\infty(t_i, t_j) > 4k + 4$; and (iii) for all $1 \leq i \leq k$, $d_\infty(t_i, V(\Gamma(H))) > 4k + 4$. Then there is an efficient algorithm that routes all demand pairs in $\mathcal{M}$ in graph H.*

The rest of this section is devoted to proving Theorem 2. For each destination vertex t_j, we define a sub-grid B_j of H of size $((2k+3) \times (2k+3))$, centered at t_j, that is, if $t_j = v(i, i')$, then B_j is a sub-grid of G spanned by rows $R_{i-(k+1)}, \ldots, R_{i+(k+1)}$ and columns $C_{i'-(k+1)}, \ldots, C_{i'+(k+1)}$ of H.

We call the resulting sub-grids $B_1, \ldots, B_k$ *boxes*. Notice that all boxes are disjoint from each other, due to the spacing of the destination terminals. We start with a high-level intuitive description of our algorithm. For each box B_j, we can associate an interval $I(B_j) \subseteq (1, N)$ with B_j, as follows: If C_{i_1}, C_{i_2} are the columns of H containing the first and the last columns of B_j, respectively, then $I(B_j) = (i_1, i_2)$. We say that the resulting set $\mathcal{I} = \{I(B_j)\}_{j=1}^k$ of intervals is aligned, if for all $i \neq j$, either $I(B_i) = I(B_j)$, or $I(B_i) \cap I(B_j) = \emptyset$. For simplicity, assume first that all intervals in $\mathcal{I}$ are aligned, and let $\{I_1, I_2, \ldots, I_r\}$ be the set of all distinct intervals in $\mathcal{I}$, ordered in their natural left-to-right order. For each $1 \leq h \leq r$, let $\mathcal{B}_h$ be the set of all boxes B_j with $I(B_j) = I_h$, and let $\mathcal{B} = \{B_j \mid 1 \leq j \leq k\}$. We define a "snake-like" ordering of the boxes in $\mathcal{B}$ as follows. For all $1 \leq h < h' \leq r$, the boxes of $\mathcal{B}_h$ appear before all boxes of $\mathcal{B}_{h'}$ in this ordering. Within each set $\mathcal{B}_h$, if h is odd, then the boxes of $\mathcal{B}_h$ are ordered in the order of their position in H from top to bottom, and otherwise they are ordered in the order of their position in H from bottom to top. We then define a set $\mathcal{P}$ of

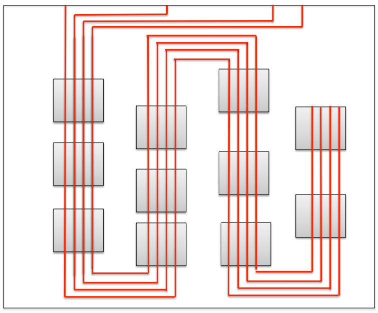

Figure 3 Traversing the boxes.

k paths, that start from the sources $s_1, \dots, s_k$, and visit all boxes in $\mathcal{B}$ in this order (see Figure 3). We will make sure that when the paths of $\mathcal{P}$ traverse any box B_j, the path $P_j \in \mathcal{P}$ that originates at s_j visits the vertex t_j. In order to accomplish this, we need the following lemma.

▶ **Lemma 3.** *Let B be the $((2k+3) \times (2k+3))$ grid, $t = v(k+2, k+2)$ the vertex in the center of the grid, and $1 \le j \le k$ any integer. Let $X = \{x_1, \dots, x_k\}$ be any set of k vertices on the top boundary edge L of B and $Y = \{y_1, \dots, y_k\}$ any set of k vertices on the bottom boundary edge L' of B, both sets ordered from left to right. Then we can efficiently find k disjoint paths $P'_1, \dots, P'_k$ in B, such that for $1 \le i \le k$, path P'_i connects x_i to y_i; all paths are internally disjoint from $V(L \cup L')$; and path P'_j contains t.*

Proof. Let $U = \{u_1, \dots, u_k\}$ be any set of k vertices on row R_{k+2} of B, ordered from left to right, such that $u_j = t$. Let $B' \subseteq B$ be the grid spanned by the top $k+2$ rows of B, and $B'' \subseteq B$ the grid spanned by the bottom $k+2$ rows of B. Note that $B' \cap B'' = R_{k+2}$.

From Observation 1, there is a set $\mathcal{P}_1$ of k node-disjoint paths in B', connecting the vertices of X to the vertices of U, and there is a set $\mathcal{P}_2$ of k node-disjoint paths in B'', connecting the vertices of U to the vertices of Y. Moreover, the paths in $\mathcal{P}_1 \cup \mathcal{P}_2$ are internally disjoint from $V(R_{k+2} \cup L \cup L')$. By concatenating the paths in $\mathcal{P}_1$ and $\mathcal{P}_2$, we obtain a set $\mathcal{P}'$ of k node-disjoint paths in B, connecting the vertices of X to the vertices of Y, such that the paths in $\mathcal{P}'$ are internally disjoint from $L \cup L'$. The intersection of each path in $\mathcal{P}'$ with the row R_{k+2} is exactly one vertex. Since graph B is planar, the paths in $\mathcal{P}'$ cross the row R_{k+2} in the same left-to-right order in which their endpoints appear on L and L'. Therefore, for $1 \le i \le k$, the ith path connects x_i to y_i, and the jth path contains the vertex t. ◀

Since in general the intervals in $\mathcal{I}$ may not be aligned, we need to define the ordering between the boxes, and the set of paths traversing them more carefully. We start by defining an ordering of the destination vertices $\{t_j\}_{j=1}^k$, which will define an ordering of their corresponding boxes.

We draw vertical lines in the grid at every column whose index is an integral multiple of $(3k+2)$, and let $\{V_1, V_2, \dots\}$ denote the sets of vertices of the resulting vertical strips of width $3k+2$, that is, for $1 \le m \le \lceil N/(3k+2) \rceil$,

$$V_m = \{v(j, j') \mid (m-1)(3k+2) < j' \le \min\{m(3k+2), N\}; 1 \le j \le N\}.$$

We assign every terminal t_j to the unique set V_m containing t_j. We then define a collection $\mathcal{S}$ of vertical strips of H as follows: For each set V_m, such that at least one terminal is assigned to V_m, we add $H[V_m]$ to $\mathcal{S}$. We assume that the set of strips $\mathcal{S} = \{S_1, \dots, S_p\}$ is indexed in their natural left-to-right order. Abusing the notation, we will denote $V(S_m)$ by V_m, for $1 \le m \le p$.

Consider some vertical strip S_m, and let $t_i, t_j \in V_m$, for $j \ne i$. Then the horizontal distance between t_i and t_j, $|\text{col}(t_i) - \text{col}(t_j)| \le 3k + 2$, and since $d_\infty(t_i, t_j) > 4k + 4$, t_i and t_j must be at a vertical distance at least $4k + 4$. Therefore, we can order the destination terminals assigned to the same vertical strip in the increasing or decreasing row index. We define the ordering of all destination terminals as follows: (1) for every $1 \le m < m' \le p$, every terminal $t_i \in V_m$ precedes every terminal $t_j \in V_{m'}$; and (2) for $t_i, t_j \in V_m$, with $\text{row}(t_j) > \text{row}(t_i)$, if m is odd then t_i precedes t_j, and if m is even, then t_j precedes t_i. Let $\mathcal{B} = \{B_j \mid 1 \le j \le k\}$ be the set of boxes corresponding to the destination vertices. The ordering of the destination vertices now imposes an ordering on $\mathcal{B}$. We re-index the boxes B_j according to this ordering, and we denote by $t(B_j)$ the unique destination terminal lying in B_j. We will say that a box B_j belongs to strip S_m iff the corresponding terminal $t(B_j) \in V_m$. (Note that B_j is not necessarily contained in S_m). The following observation is immediate.

▶ **Observation 4.** *If box B_j belongs to strip S_m, then at least $k + 2$ vertices from the top boundary of B_j, and at least $k + 2$ vertices from the bottom boundary of B_j belong to V_m.*

In order to complete the construction of the set $\mathcal{P}$ of paths routing all demand pairs, we define, for $1 \le i \le k$, a set $\mathcal{P}_i$ of k disjoint paths, with the following properties:

P1. Paths in $\mathcal{P}_1$ connect $\{s_i\}_{i=1}^k$ to some set of k vertices on the top boundary of B_1;

P2. For $i > 1$:
- if B_{i-1} and B_i belong to the same strip S_m, and m is odd, then paths in $\mathcal{P}_i$ connect k vertices on the bottom row of B_{i-1} to k vertices on the top row of B_i;
- if B_{i-1} and B_i belong to the same strip S_m, and m is even, then paths in $\mathcal{P}_i$ connect k vertices on the top row of B_{i-1} to k vertices on the bottom row of B_i;
- if B_{i-1} belongs to strip S_m and B_i to strip S_{m+1}, and m is odd, then paths in $\mathcal{P}_i$ connect k vertices on the bottom row of B_{i-1} to k vertices on the bottom row of B_i;
- if B_{i-1} belongs to strip S_m and B_i to strip S_{m+1}, and m is even, then paths in $\mathcal{P}_i$ connect k vertices on the top row of B_{i-1} to k vertices on the top row of B_i; and

P3. All paths in $\bigcup_{i=1}^k \mathcal{P}_i$ are disjoint from each other, and each path is internally disjoint from $\bigcup_{B \in \mathcal{B}} V(B)$.

▶ **Theorem 5.** *There is an efficient algorithm to find the collections $\mathcal{P}_1, \ldots, \mathcal{P}_k$ of paths with properties (3)–(3).*

We prove Theorem 5 below, and we first complete the proof of Theorem 2 here. Assume that we are given the path sets $\mathcal{P}_1, \ldots, \mathcal{P}_k$ with properties (3)–(3). For each box B_j, let $X_j \subseteq V(B_j)$ be the set of k vertices that serve as endpoints of the paths of $\mathcal{P}_j$, and let $Y_j \subseteq V(B_j)$ be the set of k vertices that serve as endpoints of the paths in $\mathcal{P}_{j+1}$. (For $j = k$, we choose the set Y_k of k vertices on the top or the bottom boundary of B_k (opposing the boundary edge where the vertices of X_k lie) arbitrarily). We construct the set $\mathcal{P}$ of paths gradually, by starting with $\mathcal{P} = \mathcal{P}_1$, and performing k iteration. We assume that at the beginning of iteration i, set $\mathcal{P}$ contains k disjoint paths, connecting the k source vertices to the vertices of X_i. This is clearly true at the beginning of the first iteration. The ith iteration is executed as follows. Assume that $t(B_i) = t_r$, and let $u \in X_i$ be the vertex where the path of $\mathcal{P}$ originating at s_r terminates. From Lemma 3, we can find a set $\mathcal{Q}_i$ of paths inside B_i, connecting the vertices of X_i to the vertices of Y_i, that are internally disjoint from the top and the bottom boundary edges of B_i, such that the path originating at u contains

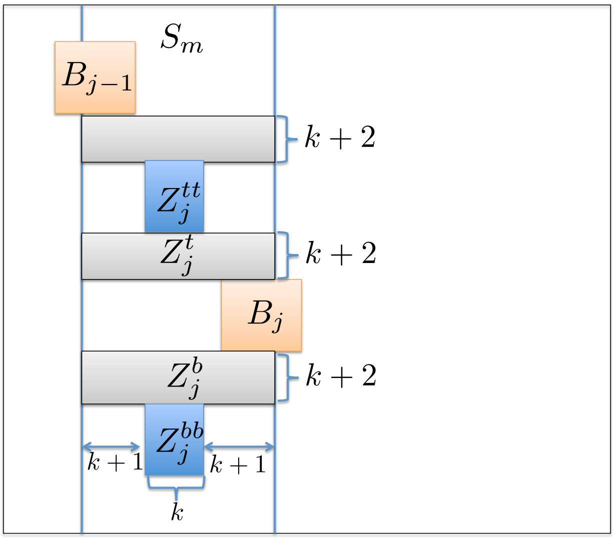

Figure 4 Graphs Z_j^b, Z_j^t, Z_j^{bb} and Z_j^{tt}.

the vertex t_r. We then concatenate the paths in $\mathcal{P}$ with the paths in $\mathcal{Q}_i$, and, if $i < k$, with the paths in $\mathcal{P}_{i+1}$, to obtain the new set $\mathcal{P}$ of paths, and continue to the next iteration. After k iterations, we obtain a collection of k node-disjoint paths that traverse all boxes B_j, such that for each $1 \le i \le k$, the path originating from s_i contains the vertex t_i. It now remains to prove Theorem 5.

Proof of Theorem 5. For each box B_j, for $1 \le j \le k$, we define four sub-graphs of H, Z_j^t, Z_j^b, Z_j^{tt}, Z_j^{bb}, that will be used in order to route the sets $\mathcal{P}_j$, $\mathcal{P}_{j+1}$ of paths.

Consider some box B_j, and assume that it belongs to strip S_m. Let C_ℓ, C_r be the columns of H that serve as the left and the right boundaries of S_m, respectively. Let R_t, R_b be the rows of H containing the top and the bottom row of B_j, respectively. Intuitively, Z_j^t is the sub-grid of strip S_m, containing the $k+1$ rows immediately above row R_t, in addition to the row R_t, and Z_j^b is defined similarly below B_j. Formally, Z_j^t is the sub-grid of H spanned by columns $C_\ell, \ldots, C_r$, and rows $R_{t-k-1}, \ldots, R_t$, so Z_j^t contains $k+2$ rows and $3k+2$ columns. Similarly, Z_j^b is the sub-grid of H spanned by columns $C_\ell, \ldots, C_r$, and rows $R_b, \ldots, R_{b+k+1}$, so Z_j^b contains $k+2$ rows and $3k+2$ columns (see Figure 4).

We now turn to define the grids Z_j^{tt} and Z_j^{bb}. Graph Z_j^{tt} is defined as follows. Assume w.l.o.g. that m is odd (recall that S_m is the strip containing $t(B_j)$). If B_j is not the topmost box that belongs to S_m, then let R_a be the row of H containing the bottom row of Z_{j-1}^b; otherwise let $R_a = R_{2k+1}$ if $j > 1$ and $R_a = R_{k+1}$ if $j = 1$. Let $R_{a'}$ be the row of H containing the top row of Z_j^t. We would like Z_j^{tt} to be the grid containing the segments of the middle k columns of S_m, between rows R_a and $R_{a'}$. Formally, we let Z_j^{tt} be the sub-grid of H spanned by rows $R_a, \ldots, R_{a'}$, and columns $C_{\ell+k+2}, \ldots, C_{\ell+2k+1}$.

We define the graph Z_j^{bb} similarly. If B_j is not the bottommost box of S_m, then let R_c be the row of H containing the top row of Z_{j+1}^t, and otherwise let $R_c = R_{N-k-1}$. Let $R_{c'}$ be the row of H containing the bottom row of Z_j^b. Graph Z_j^{bb} is the sub-grid of H spanned by rows $R_{c'}, \ldots, R_c$, and columns $C_{\ell+k+2}, \ldots, C_{\ell+2k+1}$.

Notice that if B_j is not the topmost box of S_m, then $Z_j^{tt} = Z_{j-1}^{bb}$, and if B_j is not the bottommost box of B_m, then $Z_j^{bb} = Z_{j+1}^{tt}$. We need the following observation.

▶ **Observation 6.** *For all* $1 \leq q \leq k$, $B_q \cap Z_j^{tt}, B_q \cap Z_j^{bb} = \emptyset$. *Moreover, if* $q \neq j$, *then additionally* $B_q \cap Z_j^b, B_q \cap Z_j^t = \emptyset$.

Proof. We prove for Z_j^t and Z_j^{tt}. The proofs for Z_j^b and Z_j^{bb} are symmetric.

Consider some box B_q with $q \neq j$, and assume for contradiction that $B_q \cap Z_j^t \neq \emptyset$. Then the vertical distance between $t(B_q)$ and $t(B_j)$ is less than $4k+4$, and so the horizontal distance between them must be greater than $4k+4$. However, $t(B_j)$ lies in the strip S_m, and, since B_q intersects Z_j^t, the horizontal distance between $t(B_q)$ and the left or the right column of S_m is at most $k+1$, and so the total horizontal distance between $t(B_q)$ and $t(B_j)$ is at most $4k+4$, a contradiction.

Consider now some box B_q, for $1 \leq q \leq k$, and assume for contradiction that $B_q \cap Z_j^{tt} \neq \emptyset$. If B_j is the topmost box in S_m, then B_q cannot belong to S_m. If B_j is not the topmost box of S_m, then B_q cannot belong to S_m due to the definition of Z_j^{tt}. Therefore, $t(B_q)$ lies in either S_{m+1} or S_{m-1}. But since B_q is a box of width $2k+3$, with $t(B_q)$ lying in $(k+2)$th column of B_q, it is impossible for B_q to intersect Z_j^{tt}. ◀

We are now ready to define the sets $\mathcal{P}_i$ of paths. In order to do so, we define a collection $\{H_1, \ldots, H_k\}$ of disjoint sub-graphs of H, and each such sub-graph H_i will be used to route the set $\mathcal{P}_i$ of paths. We start by letting H_1 be the union of three graphs, Z_1^t, Z_1^{tt}, and the sub-grid of H spanned by the top $k+1$ rows of H. We denote this latter graph by H_1'. Recall that the terminal $t(B_1)$ lies in strip S_1. Let A_1 be the set of k vertices on the top boundary of Z_1^{tt}, A_2 the set of k vertices on the bottom row of Z_1^{tt}, and let A_3 be any set of k vertices on the top row of B_1, that lie in S_1 (from Observation 4, such a set exists). From Observation 1, we can construct three sets of paths: set $\mathcal{P}_1'$ in H_1', connecting each source vertex to some vertex of A_1; set $\mathcal{P}_1''$ in Z_1^{tt} connecting the vertices of A_1 to the vertices of A_2 (the paths in $\mathcal{P}_1''$ are just the columns of Z_1^{tt}), and set $\mathcal{P}_1'''$ in Z_1^t, connecting the vertices of A_2 to the vertices of A_3. We let $\mathcal{P}_1$ be obtained by concatenating the paths in $\mathcal{P}_1', \mathcal{P}_1''$, and $\mathcal{P}_1'''$.

Consider now some index $1 < j \leq k$, and assume that B_{j-1} belongs to some strip S_m. We assume w.l.o.g. that m is odd (the case where m is even is dealt with similarly), and we show how to construct the set $\mathcal{P}_j$ of paths. We consider two cases. The first case is when B_j also lies in S_m. We then let H_j be the union of Z_{j-1}^b, Z_{j-1}^{bb} and Z_j^t. The set $\mathcal{P}_j$ of paths will be contained in H_j, and it is defined as follows. Let A_1 be any set of k vertices on the bottom row of B_{j-1}, that lie in V_m (this set exists due to Observation 4); let A_2 and A_3 be the vertices of the top and the bottom rows of Z_{j-1}^{bb}, respectively, and let A_4 be any set of k vertices on the top row of B_j that lie in V_m. As before, using Observation 1, we can construct three sets of paths: set $\mathcal{P}_j'$ in Z_{j-1}^b, connecting each vertex of A_1 to some vertex of A_2; set $\mathcal{P}_j''$ in Z_{j-1}^{bb} connecting the vertices of A_2 to the vertices of A_3 (the paths in $\mathcal{P}_j''$ are just the columns of Z_{j-1}^{bb}), and set $\mathcal{P}_j'''$ in Z_j^t, connecting the vertices of A_3 to the vertices of A_4. We let $\mathcal{P}_j$ be obtained by concatenating the paths in $\mathcal{P}_j', \mathcal{P}_j'$, and $\mathcal{P}_j'''$.

Finally, assume that B_j belongs to S_{m+1}. Let C_ℓ and C_r be the columns of H that serve as the left boundary of S_m and the right boundary of S_{m+1}, respectively. Let H_j' be the sub-grid of H, spanned by columns $C_\ell, \ldots, C_r$, and rows $R_{N-k-1}, \ldots, R_N$. We let H_j be the union of $Z_{j-1}^b, Z_{j-1}^{bb}, H_j', Z_j^b$ and Z_j^{bb}. Using methods similar to those described above, it is easy to find a set $\mathcal{P}_j$ of k disjoint paths in H_j, connecting k vertices on the bottom row of B_{j-1} to k vertices on the bottom row of B_j.

The case where m is even is dealt with similarly. The only difference is that in the case where B_j belongs to S_{m+1}, we use rows $R_{k+2}, \ldots, R_{2k+1}$ to define H_j', instead of rows $R_{N-k+1}, \ldots, R_N$, to avoid collision with the graph H_1'.

From the construction of the graphs H_i, it is easy to see that all such graphs are mutually disjoint, and therefore we obtain the desired sets $\mathcal{P}_1, \ldots, \mathcal{P}_k$ of paths with properties (3)–(3). ◀

4 An $\tilde{O}(n^{1/4})$-Approximation Algorithm

We assume that we are given the $(N \times N)$ grid graph $G = (V, E)$, so $n = |V| = N^2$, and a collection $\mathcal{M} = \{(s_i, t_i)\}_{i=1}^k$ of demand pairs. We say that a demand pair (s_i, t_i) is *bad* if both $d_\infty(s_i, \Gamma(G)), d_\infty(t_i, \Gamma(G)) \leq 4\sqrt{N} + 4$, and we say that it is good otherwise. Let $\mathcal{M}', \mathcal{M}'' \subseteq \mathcal{M}$ denote the sets of the good and the bad demand pairs in $\mathcal{M}$, respectively. We find an approximate solution to each of the two sub-problems, defined by $\mathcal{M}'$ and $\mathcal{M}''$, separately, and take the better of the two solutions. The following two subsections describe these two algorithms.

4.1 Routing the Good Pairs

Our first algorithm provides an $O(n^{1/4} \log n)$-approximation for the special case when all demand pairs are good. We start with a high-level overview of the algorithm. The algorithm is based on LP-rounding of (LP-flow), and so it proves that the integrality gap of (LP-flow) for this special case is $O(n^{1/4} \log n)$. The first step of the algorithm is to reduce the problem to the following special case: We are given a grid A of size $(\Theta(m) \times \Theta(m))$, where $m \leq N/8$ is some integer, and two disjoint sub-grids Q, Q' of A, of size $(m \times m)$ each, such that the minimum L_∞-distance between a vertex in Q and a vertex in Q' is $\Omega(m)$. We are also given a set $\mathcal{M}(Q, Q')$ of demand pairs, where for each pair $(s, t) \in \mathcal{M}(Q, Q')$, $s \in Q$, $t \in Q'$, and $d_\infty(s, \Gamma(Q)) > 4\sqrt{N} + 4$ (where N is the size of the side of our original grid G). We refer to the resulting routing problem as *2-square routing*. We show that an α-approximation algorithm to the 2-square routing problem immediately implies an $O(\alpha \log n)$-approximation to the original problem. We note that a similar reduction to the 2-square routing problem has been used in the past, e.g. in [1]. It is now enough to design an $O(\sqrt{m}) = O(\sqrt{N}) = O(n^{1/4})$-approximation algorithm for the 2-square routing problem. Let OPT' be the optimal solution to this problem, and let $\mathcal{M}^* \subseteq \mathcal{M}(Q, Q')$ be the subset of the demand pairs routed in OPT'. Notice that $|\mathsf{OPT}'| \leq 4m$, since each path in the optimal solution must contain at least one vertex of $\Gamma(Q)$. We define a partition $\mathcal{X}$ of Q into sub-squares of size $(\Theta(\sqrt{m}) \times \Theta(\sqrt{m}))$, and show an efficient algorithm to find a subset $\tilde{\mathcal{M}} \subseteq \mathcal{M}(Q, Q')$ of $\Omega(|\mathsf{OPT}'|/\sqrt{m})$ demand pairs, with $|\tilde{\mathcal{M}}| \leq \sqrt{m}$, so that the following holds. Let S' and T' denote the sets of the source and the destination vertices, participating in the pairs in $\tilde{\mathcal{M}}$, respectively. Then (i) for each square $X \in \mathcal{X}$, $|V(X) \cap S'| \leq 1$; (ii) all vertices in T' can be simultaneously routed to $\Gamma(Q') \setminus \Gamma(G)$ on node-disjoint paths; and (iii) every vertex of A participates in at most one demand pair. Set $\tilde{\mathcal{M}}$ is found by setting up an appropriate instance of the maximum flow problem. It is then easy to route all vertices in T' to $\Gamma(Q)$ on paths that are node-disjoint and internally disjoint from Q. We then use Theorem 2 to complete the routing inside Q. We now turn to describe the algorithm more formally.

Let (f, x) be the optimal solution to the linear program (LP-flow) on instance $(G, \mathcal{M}')$, and let $\mathsf{OPT}_{\mathsf{LP}}$ be its value. We show an algorithm that routes $\Omega(\mathsf{OPT}_{\mathsf{LP}}/(n^{1/4} \cdot \log n))$ demand pairs. The algorithm consists of two steps. In the first step, we reduce the problem to routing between two square sub-grids of G. We note that a similar reduction has been used in prior work, e. g. by Aggarwal et al. [1]. In the second step, we show an approximation algorithm for the resulting sub-problem.

Reduction to the 2-Square Problem

In this step, we reduce the problem of routing on G with a general set $\mathcal{M}'$ of good demand pairs, to a problem where we are given two disjoint sub-grids (or squares) Q_1, Q_2 of G, and every demand pair (s_j, t_j) has $s_j \in Q_1$ and $t_j \in Q_2$, or vice versa.

We start by partitioning the set $\mathcal{M}'$ of the demand pairs into $\lceil \log N \rceil$ subsets, $\mathcal{M}_1, \ldots, \mathcal{M}_{\lceil \log N \rceil}$, where

$$\mathcal{M}_h = \left\{ (s_j, t_j) \in \mathcal{M}' \mid 2^{h-1} \leq d_\infty(s_j, t_j) < 2^h \right\}.$$

For each $1 \leq h \leq \lceil \log N \rceil$, let $F_h = \sum_{(s_j,t_j) \in \mathcal{M}_h} x_j$, where x_j is the amount of flow sent from s_j to t_j in the solution to (LP-flow). We let h^* be the index maximizing F_{h^*}, so $F_{h^*} \geq \mathsf{OPT}_{\mathsf{LP}} / \lceil \log N \rceil$. From now on, we focus on routing the pairs in $\mathcal{M}_{h^*}$, and we will route $\Omega(F_{h^*}/n^{1/4})$ such pairs.

Assume first that $h^* \leq 6$. In this case, we partition the grid into sub-grids of size at most (256×256) with a random offset, as follows. Select an integer $0 \leq z < 256$ uniformly at random, and use the set $\mathcal{C} = \{C_{z+256i}\}_{i=0}^{\lfloor (N-z)/256 \rfloor}$ of columns and the set $\mathcal{R} = \{R_{z+256i}\}_{i=0}^{\lfloor (N-z)/256 \rfloor}$ of rows to partition the grid into sub-grids. Let $\mathcal{Q}$ be the resulting collection of sub-grids. We define a new LP-solution as follows: start with the original LP-solution; for every demand pair $(s_j, t_j) \notin \mathcal{M}_{h^*}$, set $x_j = 0$, and $f(P) = 0$ for all paths $P \in \mathcal{P}_j$. For every demand pair $(s_j, t_j) \in \mathcal{M}_{h^*}$, if s_j or t_j lie on a row of $\mathcal{R}$ or a column of $\mathcal{C}$, or if they belong to different sub-grids in $\mathcal{Q}$, set $x_j = 0$ and $f(P) = 0$ for all paths $P \in \mathcal{P}_j$. Since for each pair $(s_j, t_j) \in \mathcal{M}_{h^*}$, $d_\infty(s_j, t_j) < 64$, it is easy to see that the expected value of the resulting LP-solution is $W = \Omega(F_{h^*}) = \Omega(\mathsf{OPT}_{\mathsf{LP}} / \log N) = \Omega(\mathsf{OPT}_{\mathsf{LP}} / \log n)$. By trying all possible values $0 \leq z < 256$, we can find a partition $\mathcal{Q}$ of G, and a corresponding LP-solution, whose value is at least W. Notice that for each sub-grid $Q \in \mathcal{Q}$, the number of vertices of Q is bounded by 256^2, and so the total amount of flow routed between the demand pairs contained in Q is bounded by 256^2. For each sub-grid $Q \in \mathcal{Q}$, if there is any demand pair $(s_j, t_j) \in \mathcal{M}_{h^*}$ with $s_j, t_j \in Q$, and a non-zero value x_j in the current LP-solution, we select any such pair and route it via any path P contained in Q, which is disjoint from the boundary of Q. It is easy to see that the total number of the demand pairs routed is $\Omega(W) = \Omega(\mathsf{OPT}_{\mathsf{LP}} / \log n)$. From now on, we assume that $h^* > 6$.

For convenience, we denote h^* by h from now on. Let $m = 2^h/16$. We partition the grid into a collection $\mathcal{Q} = \{Q_{p,q} \mid 1 \leq p \leq \lfloor N/m \rfloor, 1 \leq q \leq \lfloor N/m \rfloor\}$ of disjoint sub-grids, or squares, as follows. First, partition G into $\lfloor N/m \rfloor$ disjoint vertical strips $V_1, \ldots, V_{\lfloor N/m \rfloor}$, each containing m consecutive columns of G, except for the last strip, that may contain between m and $2m-1$ columns. Next, partition each vertical strip V_p into $\lfloor N/m \rfloor$ disjoint sub-grids, where each sub-grid contains m consecutive rows of V_p, except possibly for the last sub-grid, that may contain between m and $2m-1$ rows. The width and the hight of each such sub-grid is then between m and $2m-1$, where $m \leq N/16$. Notice that for each such grid $Q_{p,q} \in \mathcal{Q}$, if L is the left boundary edge of $Q_{p,q}$, and L' is the left boundary edge of G, then either $L \subseteq L'$, or L and L' are separated by at least $m-1$ columns. The same holds for the other three boundary edges. We need the following observation.

▶ **Observation 7.** *Let $(s_j, t_j) \in \mathcal{M}_h$ be a demand pair, and assume that $s_j \in Q_{p,q}$ and $t_j \in Q_{p',q'}$. Then:*

$$5 \leq |p - p'| + |q - q'| \leq 34.$$

Proof. We first show that $|p - p'| + |q - q'| \geq 5$. Indeed, assume otherwise. Then both the horizontal and the vertical distances between s_j and t_j are less than $8m = 8 \cdot 2^h/16 = 2^{h-1}$, while $d_\infty(s_j, t_j) \geq 2^{h-1}$, a contradiction.

Assume now for contradiction that $|p-p'|+|q-q'|>34$. Then $d_\infty(s_j,t_j)>16m=2^h$, contradicting the fact that $d_\infty(s_j,t_j)<2^h$. ◀

We say that a pair $(Q_{p,q},Q_{p',q'})$ of squares in $\mathcal{Q}$ is *interesting* iff $5\le|p-p'|+|q-q'|\le 34$. Let $\mathcal{Z}$ be the set of all interesting pairs of squares in $\mathcal{Q}$. We associate an NDP instance with each such pair $Z=(Q_{p,q},Q_{p',q'})$, as follows. Let $\mathcal{M}(Z)\subseteq\mathcal{M}_h$ be the set of all demand pairs $(s_j,t_j)\in\mathcal{M}_h$ where $s_j\in Q_{p,q}$ and $t_j\in Q_{p',q'}$, or vice versa. We also define a box $A(Z)$, that contains $Q_{p,q}\cup Q_{p',q'}$, and adds a margin of m around them, if possible. More precisely, let ℓ be the smallest integer, such that $R_\ell\cap(Q_{p,q}\cup Q_{p',q'})\ne\emptyset$, and let ℓ' be the largest integer, such that $R_{\ell'}\cap(Q_{p,q}\cup Q_{p',q'})\ne\emptyset$. Similarly, let b and b' be the smallest and the largest integers, respectively, such that $C_b\cap(Q_{p,q}\cup Q_{p',q'}),C_{b'}\cap(Q_{p,q}\cup Q_{p',q'})\ne\emptyset$. We then let $A(Z)$ be the sub-grid of G spanned by rows $R_{\max\{1,\ell-m\}},\ldots,R_{\min\{\ell'+m,N\}}$, and by columns $C_{\max\{1,b-m\}},\ldots,C_{\min\{b'+m,N\}}$. For every interesting pair of squares $Z\in\mathcal{Z}$, we now define an instance of the NDP problem on graph $A(Z)$, with the set $\mathcal{M}(Z)$ of demand pairs. Let $F(Z)$ be the total amount of flow routed between the demand pairs in $\mathcal{M}(Z)$ in the current LP-solution F_h to our original problem (notice that in our LP-solution, the fractional routing of the demand pairs in $\mathcal{M}(Z)$ is not necessarily contained in $A(Z)$). From the above discussion, $\sum_{Z\in\mathcal{Z}}F(Z)=\Omega(\mathsf{OPT}_{\mathsf{LP}}/\log N)$. We will show an algorithm that routes, for each $Z\in\mathcal{Z}$, $\Omega(F(Z)/n^{1/4})$ demand pairs in $\mathcal{M}(Z)$ integrally, in graph $A(Z)$. However, it is possible that for two pairs $Z,Z'\in\mathcal{Z}$, $A(Z)\cap A(Z')\ne\emptyset$, and the two routings may interfere with each other. We resolve this problem in the following step.

From Observation 7, it is easy to see that for each interesting pair of squares $Z\in\mathcal{Z}$, the number of pairs $Z'\in\mathcal{Z}$ with $A(Z)\cap A(Z')\ne\emptyset$ is bounded by some constant c. We construct a graph H, whose vertex set is $V(H)=\{v_Z\mid Z\in\mathcal{Z}\}$, and there is an edge $(v_Z,v_{Z'})$ iff $A(Z)\cap A(Z')\ne\emptyset$. As observed above, the maximum vertex degree in this graph is bounded by some constant c, and so we can color H with $c+1$ colors. Let $U_i\subseteq V(H)$ be the set of vertices of color i. We select a color class i^*, maximizing the value $F^{i^*}=\sum_{v_Z\in U_{i^*}}F(Z)$. Clearly, $F^{i^*}=\Omega(\mathsf{OPT}_{LP}/\log N)$. For every pair $v_Z,v_{Z'}$ of vertices in U_{i^*}, we now have $A(Z)\cap A(Z')=\emptyset$. In order to obtain an $O(n^{1/4}\log n)$-approximation algorithm for the special case where all demand pairs are good, it is now enough to prove the following theorem.

▶ **Theorem 8.** *There is an efficient algorithm, that, for every interesting pair $Z\in\mathcal{Z}$ of squares, routes $\Omega(F(Z)/n^{1/4})$ demand pairs of $\mathcal{M}(Z)$ inside the grid $A(Z)$.*

The Rounding Algorithm

From now on we focus on proving Theorem 8. We assume that we are given an interesting pair $Z=(Q,Q')$ of squares, where the width and the height of each square is bounded by $2m-1$. We are also given a collection $\mathcal{M}(Z)$ of demand pairs, that, for convenience, we denote by $\mathcal{M}$ from now on. For each demand pair $(s_j,t_j)\in\mathcal{M}$, we can assume without loss of generality that $s_j\in Q$ and $t_j\in Q'$. Recall that we have a fractional solution (f,x) that routes $F^*=F(Z)$ flow units between the demand pairs in $\mathcal{M}$, in the grid G. Additionally, we are given a square $A=A(Z)$, containing Q and Q', as defined above. Recall that for any pair $v\in Q$, $v'\in Q'$ of vertices, $d_\infty(v,v')\ge 5m$.

From our definition of good demand pairs, it is possible that for a pair $(s_j,t_j)\in\mathcal{M}$, $d_\infty(s_j,\Gamma(G))\le 4\sqrt{N}+4$, or $d_\infty(t_j,\Gamma(G))\le 4\sqrt{N}+4$, but not both. We say that (s_j,t_j) is a type-1 pair if $d_\infty(s_j,\Gamma(G))\le 4\sqrt{N}+4$, and we say that it is a type-2 demand pair otherwise. Let F_1 be the total flow in the LP-solution between the type-1 demand pairs, and F_2 the total flow between type-2 demand pairs. We assume without loss of generality that

$F_1 \leq F_2$, so $F_2 \geq F^*/2$. From now on we focus on routing type-2 demand pairs. Abusing the notation, we use $\mathcal{M}$ to denote the set of all type-2 demand pairs.

We next define a sub-grid Q^+ of A, obtained by adding a margin of m around the grid Q, if possible. Specifically, let $R_\ell, R_{\ell'}$ be the rows of G, containing the top and the bottom rows of Q, respectively. Similarly, let $C_b, C_{b'}$ be the columns of G, containing the left and the right columns of Q, respectively. We let Q^+ be the sub-grid of G, spanned by rows $R_{\max\{1,\ell-m\}}, \ldots, R_{\min\{N,\ell'+m\}}$ and columns $C_{\max\{1,b-m\}}, \ldots, C_{\min\{N,b'+m\}}$. From our definition of A, $Q^+ \subseteq A$. Moreover, since $m \leq N$, and since we have assumed that all demand pairs are type-2 good pairs, all source vertices corresponding to the demand pairs in $\mathcal{M}$ are within L_∞ distance at least $4\sqrt{m}+5$ from the boundary of Q^+. We start with the following simple observation.

▶ **Observation 9.** *Let L' be a boundary edge of Q', such that $L' \not\subseteq \Gamma(G)$, and let $Y \subseteq V(L')$ be any set of its vertices. Then there is a boundary edge L of Q^+, and a set $\mathcal{P}$ of $|Y|$ disjoint paths in graph A, connecting every vertex of Y to a distinct vertex of L, such that the paths in $\mathcal{P}$ are internally disjoint from $Q^+ \cup Q'$.*

Proof. If the top boundary edge $\tilde{L}$ of Q^+ is separated by at least m rows from the top boundary edge of G, then set $L = \tilde{L}$; otherwise, let L be the bottom boundary edge of Q^+ - notice that it must be separated by at least m rows from the bottom boundary edge of G. Let $X \subseteq V(L)$ be any set of $|Y|$ vertices, and let A' be the graph obtained from A, by deleting all vertices in $Q^+ \setminus X$ and $Q' \setminus Y$ from it. It is enough to show that there is a set $\mathcal{P}$ of $|X| = |Y|$ disjoint paths in A', connecting the vertices of X to the vertices of Y. Let $z = |X|$. From Menger's theorem, if such a set of paths does not exist, then there is a set J of at most $z-1$ vertices, such that in $A' \setminus J$ there is no path from a vertex of $X \setminus J$ to a vertex of $Y \setminus J$. But from our definition of Q^+, Q', and A, it is clear that no such set of vertices exists. ◀

Let r be the smallest integral power of 2 greater than $4\sqrt{m}+4$, so $r = \Theta(\sqrt{m})$. Our next step is to partition Q into a collection $\mathcal{X}$ of disjoint sub-grids of size $(r \times r)$ each. For $1 \leq p, q \leq m/r$, we let $X_{p,q}$ be the sub-grid of Q, spanned by rows $R_{(p-1)r+1}, \ldots, R_{pr}$ and columns $C_{(q-1)r+1}, \ldots, C_{qr}$ of Q. We then let $\mathcal{X} = \{X_{p,q} \mid 1 \leq p, q \leq m/r\}$. The next theorem is key to finding the final routing.

▶ **Theorem 10.** *There is a subset $\mathcal{M}_1 \subseteq \mathcal{M}$ of $\Omega(F^*/n^{1/4})$ demand pairs, such that every vertex of $Q \cup Q'$ participates in at most one demand pair. Moreover, if S_1 and T_1 denote the sets of all source and all destination vertices of the pairs in $\mathcal{M}_1$, respectively, then:*

- *for every square $X_{p,q} \in \mathcal{X}$, at most one vertex of $X_{p,q}$ belongs to S_1; and*
- *there is a boundary edge L' of Q', with $L' \not\subseteq \Gamma(G)$, and a set $\mathcal{P}_1$ of node-disjoint paths in graph Q', connecting every vertex of T_1 to a distinct vertex of L'.*

Proof. Let U be the union of the boundary edges L' of Q', with $L' \not\subseteq \Gamma(G)$. We build a flow network $\mathcal{N}$, starting with the graph Q'. We add a source vertex a, that connects to every vertex in U with a directed edge. Let $S \subseteq Q$ be the set of all vertices participating in the demand pairs in $\mathcal{M}$ as sources. Observe that each vertex $s \in S$ may participate in several demand pairs in $\mathcal{M}$. We add every vertex $s \in S$ to graph $\mathcal{N}$, and for each demand pair $(s,t) \in \mathcal{M}$, we connect t to s with a directed edge. Next, for each square $X_{p,q} \in X$, we add a vertex $u_{p,q}$, and we connect every vertex $s \in S \cap X_{p,q}$ to $u_{p,q}$ with a directed edge. Finally, we add a destination vertex b, and connect every vertex $u_{p,q}$ for $1 \leq p, q \leq m/r$ to b with a directed edge. We set all vertex-capacities (except for those of a and b) to 1.

We claim that there is a valid flow of value $\Omega(F^*/\sqrt{m})$ from a to b in $\mathcal{N}$. Indeed, consider the multicommodity flow between the demand pairs in $\mathcal{M}$, given by our current LP-solution. For each (s_j, t_j)-pair in $\mathcal{M}$, we send $x_j/4r$ flow units on the edge (t_j, s_j) in $\mathcal{N}$. For each flow-path $P \in \mathcal{P}_j$, notice that P must contain some vertex of U. Let v be the last such vertex on P (where we view P as directed from s_j to t_j), and let P' be the sub-path of P from v to t_j. We send $f(P)/4r$ flow units on every edge in P'. For every vertex $v \in U$, we set the flow on the edge (a, v) to be the total flow leaving the vertex v; for each vertex $s \in S$, with $s \in X_{p,q}$, we set the flow on the edge $(s, u_{p,q})$ to be the total amount of flow entering s. The flow on edge $(u_{p,q}, b)$ is then set to the total amount of flow entering $u_{p,q}$. Notice that for each square $X_{p,q}$, every flow-path originating at a vertex of $S \cap X_{p,q}$ must cross the boundary $\Gamma(X_{p,q})$ of $X_{p,q}$, that contains at most $4r$ vertices. Therefore, the total amount of flow in the original LP-solution leaving the vertices in $S \cap X_{p,q}$ is at most $4r$. It is now easy to see that we have defined a valid a-b flow of value $\tilde{F} = \Omega(F^*/\sqrt{m})$.

From the integrality of flow, there is an integral flow of the same value in $\mathcal{N}$. Let $\mathcal{P}$ be the set of paths carrying one flow unit in the resulting flow. Then there is a boundary edge L' of Q', such that $L' \not\subseteq \Gamma(G)$, with at least $\tilde{F}/4$ of the paths in $\mathcal{P}$ containing a vertex of L'. Let $\mathcal{P}' \subseteq \mathcal{P}$ be this set of paths. We are now ready to define the final set $\mathcal{M}_1$ of the demand pairs, and the corresponding set $\mathcal{P}_1$ of paths. Consider some path $P \in \mathcal{P}'$, and let (t, s) be the unique edge with $(s, t) \in \mathcal{M}$ on this path. We then add (s, t) to $\mathcal{M}_1$. Let P' be the sub-path of P, starting from the last vertex on P that belongs to L', to vertex t. We add P' to $\mathcal{P}_1$. This finishes the definition of the subset $\mathcal{M}_1$ of demand pairs, and the corresponding set $\mathcal{P}_1$ of paths. ◀

If $|\mathcal{M}_1| > \sqrt{m}$, then we discard pairs from $\mathcal{M}_1$, until $|\mathcal{M}_1| \leq \sqrt{m}$ holds, and we update the sets S_1, T_1, and $\mathcal{P}_1$ accordingly.

For $w, w' \in \{0, 1\}$, let $S_{w,w'}$ be a subset containing all vertices $s \in S_1$ lying in the squares $X_{p,q}$, where $p = w \mod 2$ and $q = w' \mod 2$. Then there is some choice of $w, w' \in \{0, 1\}$, so that $|S_{w,w'}| \geq |S_1|/4$. We let $S_2 = S_{w,w'}$ for this choice of w, w', and we define $\mathcal{M}_2 = \{(s, t) \in \mathcal{M}_1 \mid s \in S_2\}$, and T_2 as the set of all destination vertices for the pairs in $\mathcal{M}_2$. Let $\mathcal{P}_2 \subseteq \mathcal{P}_1$ be the set of paths originating from the vertices of T_2. Let Y be the set of endpoints of the paths in $\mathcal{P}_2$ that lie on the boundary edge L' of Q'. Finally, from Observation 9, there is a boundary edge L of Q^+, a set Y' of $|Y|$ vertices of L, and a set $\mathcal{P}'_2$ of disjoint paths in A, connecting every vertex in Y to a distinct vertex of Y', so that the paths in $\mathcal{P}'_2$ are internally disjoint from $Q^+ \cup Q'$. By concatenating the paths in $\mathcal{P}_2$ and $\mathcal{P}'_2$, we obtain a new set $\mathcal{P}^*$ of paths, connecting every vertex of T_2 to a distinct vertex of Y'. Denote $\mathcal{M}_2 = \{(s_j, t_j)\}_{j=1}^{|\mathcal{M}_2|}$, and let $u_j \in Y'$ be the vertex where the path $P_j \in \mathcal{P}^*$, originating at vertex t_j, terminates. Notice that all vertices in S_2 are now at the L_∞-distance at least $r > 4\sqrt{m} + 4$ from each other, and at distance at least $4\sqrt{m} + 5$ from the boundaries of Q^+, and $|\mathcal{M}_1| \leq \sqrt{m}$. From Theorem 2, we can efficiently find a set $\mathcal{Y}$ of disjoint paths in graph Q^+, connecting every vertex $s_j \in S_2$ to the corresponding vertex $u_j \in Y'$. By concatenating the paths in $\mathcal{P}^*$ and $\mathcal{Y}$, we obtain a set of paths routing all pairs in $\mathcal{M}_2$.

Notice that from the above discussion, $|\mathcal{M}_2| = \min\{\Omega(\sqrt{m}), \Omega(F^*/\sqrt{m})\}$. It is easy to see that $F^* \leq 4m$, since every flow-path routing a pair in $\mathcal{M}$ must cross the boundary of Q'. Therefore, $|\mathcal{M}_2| = \Omega(F^*/\sqrt{m})$. Since $m \leq N = \sqrt{n}$, our algorithm routes $\Omega(F^*/n^{1/4})$ demand pairs.

4.2 Routing the Bad Pairs

The goal of this section is to prove the following theorem.

▶ **Theorem 11.** *Let $(G, \mathcal{M})$ be an instance of the NDP problem, where G is an $(N \times N)$ grid, and $\mathcal{M} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$. Assume further that for each demand pair (s_j, t_j), both $d_\infty(s_j, \Gamma(G)), d_\infty(t_j, \Gamma(G)) < d^*$, for some parameter $1 \leq d^* \leq N/4$. Then there is an efficient algorithm that finds an $O(d^*)$-approximate solution to the NDP instance $(G, \mathcal{M})$.*

Notice that by setting $d^* = 4\sqrt{N} + 5$, so that $d^* = \Theta(n^{1/4})$, we obtain an $O(n^{1/4})$-approximate solution for NDP instances on grid graphs, where all demand pairs are bad.

The rest of this section is dedicated to proving Theorem 11. Let T be the set of all vertices participating in the bad demand pairs. We call the vertices in T *terminals*. Let L_1, L_2, L_3, L_4 be the four boundary edges of the grid G. Notice that a terminal $t \in T$ may be within distance d^* from up to two boundary edges. For each terminal $t \in T$, we let $L(t)$ be any boundary edge of G, such that $d_\infty(t, V(L(t))) < d^*$. We now partition all bad demand pairs into 16 subsets: for $1 \leq p, q \leq 4$, set $\mathcal{M}_{p,q}$ contains all pairs (s_j, t_j), where $L(s_j) = L_p$ and $L(t_j) = L_q$. Let OPT be the optimal solution to the NDP instance. For every possible choice of $1 \leq p, q \leq 4$, let $\mathsf{OPT}_{p,q}$ be the optimal solution restricted to the pairs in $\mathcal{M}_{p,q}$. Clearly, there is a choice of p and q, such that at least $|\mathsf{OPT}|/16$ of the demand pairs routed in OPT belong to $\mathcal{M}_{p,q}$, and so $|\mathsf{OPT}_{p,q}| \geq \mathsf{OPT}/16$. For each choice of values $1 \leq p, q \leq 4$, we show an algorithm that routes $\Omega(|\mathsf{OPT}_{p,q}|/d^*)$ demand pairs in $\mathcal{M}_{p,q}$. We then take the best of these solutions, thus obtaining an $O(d^*)$-approximation algorithm.

Fix some $1 \leq p, q \leq 4$. We consider three cases.

The first case happens when L_p and L_q are two distinct opposing boundary edges of G. We assume without loss of generality that L_p is the top, and L_q is the bottom boundary of G. We say that a subset $\mathcal{M}' \subseteq \mathcal{M}_{p,q}$ of demand pairs is a *monotone matching*, if the following holds. Let S' be the set of all source vertices, and T' the set of all destination vertices, participating in the pairs in $\mathcal{M}'$. Then:

- All vertices of S' lie in distinct columns of G;
- All vertices of T' lie in distinct columns of G;
- Every vertex of $S' \cup T'$ participates in exactly one demand pair; and
- For any two distinct pairs $(s_i, t_i), (s_j, t_j) \in \mathcal{M}'$, $\mathrm{col}(s_i) < \mathrm{col}(s_j)$ iff $\mathrm{col}(t_i) < \mathrm{col}(t_j)$.

The following observation is immediate.

▶ **Observation 12.** *Let $\mathcal{M}' \subseteq \mathcal{M}_{p,q}$ be any monotone matching with $|\mathcal{M}'| \leq N/2$. Then there is an efficient algorithm to route all pairs in $\mathcal{M}'$ in graph G.*

Our algorithm then simply computes the largest monotone matching $\mathcal{M}' \subseteq \mathcal{M}_{p,q}$, using standard dynamic programming: We maintain a dynamic programming table Π, that contains, for all $0 \leq x, y \leq N$, an entry $\Pi(x, y)$, whose value is the size of the largest monotone matching $\mathcal{M}(x, y) \subseteq \mathcal{M}_{p,q}$, such that every source vertex s participating in pairs in $\mathcal{M}(x, y)$ has $1 \leq \mathrm{col}(s) \leq x$, and every destination vertex t participating in pairs in $\mathcal{M}(x, y)$ has $1 \leq \mathrm{col}(t) \leq y$. We fill the entries of the table from smaller to larger values of $x + y$, initializing $\Pi(x, 0) = 0$ and $\Pi(0, y) = 0$ for all x and y. Entry $\Pi(x, y)$ is computed as follows. If there is a pair $(s, t) \in \mathcal{M}_{p,q}$, with $\mathrm{col}(s) = x$ and $\mathrm{col}(t) = y$, then we let $\Pi(x, y)$ be the maximum of $\Pi(x-1, y-1) + 1$, $\Pi(x-1, y)$, and $\Pi(x, y-1)$. Otherwise, $\Pi(x, y)$ is the maximum of $\Pi(x-1, y)$, and $\Pi(x, y-1)$. The size of the largest monotone matching $\mathcal{M}' \subseteq \mathcal{M}_{p,q}$ is then stored in $\Pi(N, N)$, and we can use standard techniques to compute the matching itself. Finally, we show that there is a large enough monotone matching $\mathcal{M}' \subseteq \mathcal{M}_{p,q}$.

▶ **Lemma 13.** *There is a monotone matching $\mathcal{M}' \subseteq \mathcal{M}_{p,q}$ of cardinality $\Omega(\mathsf{OPT}_{p,q}/d^*)$.*

Proof. For every source vertex s of a demand pair in $\mathcal{M}_{p,q}$, let $P(s)$ denote the segment of the column in which s lies, from the first row of G to s itself. Similarly, for each destination vertex t of a demand pair in $\mathcal{M}_{p,q}$, let $P(t)$ denote the segment of the column in which t lies, from t to the last row of G.

Consider the solution $\mathsf{OPT}_{p,q}$, and let $\mathcal{M}^* \subseteq \mathcal{M}_{p,q}$ be the set of the demand pairs routed in it. For each pair $(s_i, t_i) \in \mathcal{M}^*$, let $P_i \in \mathsf{OPT}_{p,q}$ be the path routing this demand pair in the solution. We say that two demand pairs (s_i, t_i) and (s_j, t_j) in $\mathcal{M}^*$ *have a conflict* iff either P_i contains a vertex of $P(s_j) \cup P(t_j)$, or P_j contains a vertex of $P(s_i) \cup P(t_i)$.

Let H be a directed graph, that contains a vertex v_i for every pair $(s_i, t_i) \in \mathcal{M}^*$, and a directed edge (v_i, v_j) iff path P_i intersects $P(s_j)$ or $P(t_j)$. Notice that the length of every path $P(s_j)$ or $P(t_j)$ is bounded by d^*, and so every vertex of H has in-degree bounded by $2d^*$. Therefore, any vertex-induced sub-graph H' of H with z vertices has at most $2d^*z$ edges, and contains at least one vertex whose degree (including the incoming and the outgoing edges) is at most $4d^*$.

We now construct the set $\mathcal{M}'$ of demand pairs as follows. Start with $\mathcal{M}' = \emptyset$. While H is non-empty, let v_i be any vertex of degree at most $4d^*$. Delete v_i and all its neighbors from H, and add the pair (s_i, t_i) to $\mathcal{M}'$. When this procedure terminates, it is easy to see that $\mathcal{M}'$ contains at least $|\mathsf{OPT}_{p,q}|/(4d^* + 1) = \Omega(|\mathsf{OPT}_{p,q}|/d^*)$ demand pairs. Moreover, if (s_i, t_i) and (s_j, t_j) are distinct pairs in $\mathcal{M}'$, then there is no conflict between (s_i, t_i) and (s_j, t_j). In particular, this means that $\text{col}(s_i) \neq \text{col}(s_j)$ and $\text{col}(t_i) \neq \text{col}(t_j)$. Moreover, if we assume that $\text{col}(s_i) < \text{col}(s_j)$, then $\text{col}(t_i) < \text{col}(t_j)$ must hold: this is since the union of $P_i, P(s_i)$ and $P(t_i)$ partitions the face defined by $\Gamma(G)$ into a number of sub-faces, and both s_j and t_j must be contained in a single sub-face, as the path P_j cannot intersect the paths $P_i, P(s_i)$ and $P(t_i)$. ◀

This concludes the analysis of the algorithm for the case where L_p and L_q are two distinct opposing boundary edges of G. The case where L_p and L_q are two adjacent boundary edges of G is dealt with very similarly. Finally, we consider the case where $L_p = L_q$. Assume without loss of generality that L_p is the bottom boundary edge of the grid. We say that a subset $\mathcal{M}' \subseteq \mathcal{M}_{p,q}$ is a *nested matching*, if the following holds. Let S' be the set of all source vertices, and T' the set of all destination vertices, participating in the pairs in $\mathcal{M}'$. Then:

- All vertices of S' lie in distinct columns of G;
- All vertices of T' lie in distinct columns of G;
- Every vertex of $S' \cup T'$ participates in exactly one demand pair; and
- For any two distinct pairs $(s_i, t_i), (s_j, t_j) \in \mathcal{M}'$, with $\text{col}(s_i)$ lying to the left of $\text{col}(s_j)$, either both $\text{col}(s_i), \text{col}(t_i)$ lie to the left of both $\text{col}(s_j), \text{col}(t_j)$, or both $\text{col}(s_j), \text{col}(t_j)$ lie between $\text{col}(s_i)$ and $\text{col}(t_i)$, or both $\text{col}(s_i), \text{col}(t_i)$ lie between $\text{col}(t_j)$ and $\text{col}(s_j)$.

It is immediate to see that any nested matching $\mathcal{M}' \subseteq \mathcal{M}_{p,q}$, with $|\mathcal{M}'| \leq N/2$ can be routed efficiently in G. As before, we can find a largest-cardinality nested matching $\mathcal{M}' \subseteq \mathcal{M}_{p,q}$ using standard dynamic programming techniques. The following lemma will then finish the proof.

▶ **Lemma 14.** *There is a nested matching $\mathcal{M}' \subseteq \mathcal{M}_{p,q}$ of cardinality $\Omega(\mathsf{OPT}_{p,q}/d^*)$.*

Proof. We construct the paths $P(s), P(t)$, the graph H', and the matching $\mathcal{M}'$ corresponding to an independent set in H' exactly as in the proof of Lemma 13. As before, $|\mathcal{M}'| = \Omega(\mathsf{OPT}_{p,q}/d^*)$. Moreover, if (s_i, t_i) and (s_j, t_j) are distinct pairs in $\mathcal{M}'$, then there is no conflict between (s_i, t_i) and (s_j, t_j). As before, this means that $\text{col}(s_i) \neq \text{col}(s_j)$ and $\text{col}(t_i) \neq \text{col}(t_j)$. Assume now that $\text{col}(s_i)$ lies to the left of $\text{col}(s_j)$. Then the union of

$P_i, P(s_i)$ and $P(t_i)$ partitions the face defined by $\Gamma(G)$ into a number of sub-faces, and both s_j and t_j must be contained in a single sub-face, as before. In this case, this means that either both $\text{col}(s_i), \text{col}(t_i)$ lie to the left of both $\text{col}(s_j), \text{col}(t_j)$, or both $\text{col}(s_j), \text{col}(t_j)$ lie between $\text{col}(s_i)$ and $\text{col}(t_i)$, or both $\text{col}(s_i), \text{col}(t_i)$ lie between $\text{col}(t_j)$ and $\text{col}(s_j)$. ◀

4.3 Putting Everything Together

Our algorithm for an input NDP instance $(G, \mathcal{M})$, where G is an $(N \times N)$ grid, applies the algorithm from Section 4.1 to the set $\mathcal{M}'$ of the good demand pairs, and the algorithm from Section 4.2 to the set $\mathcal{M}''$ of the bad demand pairs, and returns the better of the two solutions. Since each of the two algorithms achieves an $O(n^{1/4} \log n)$-approximation to the corresponding problem, and since at least half of the demand pairs routed in the optimal solution are either all good pairs, or all bad pairs, we obtain an $O(n^{1/4} \log n)$-approximation overall.

5 APX-Hardness Proof

In this section we prove that NDP does not have a $(1+\delta)$-approximation algorithm on grid graphs, for some fixed $\delta > 0$, unless $\mathsf{P} = \mathsf{NP}$. We perform a reduction from the 3SAT(5) problem. In this problem we are given a 3SAT formula φ on n variables and $5n/3$ clauses. Each clause contains exactly 3 distinct literals and each variable participates in exactly 5 different clauses. We say that φ is a Yes-Instance if it is satisfiable. We say that φ is a No-Instance with respect to some parameter ϵ, if no assignment satisfies more than an ϵ-fraction of clauses. The following well-known theorem follows from the PCP theorem [7, 6].

▶ **Theorem 15.** *There is a constant $\epsilon : 0 < \epsilon < 1$, such that it is NP-hard to distinguish between* Yes-Instances *and* No-Instances *(defined with respect to ϵ) of the 3SAT(5) problem.*

Let φ be the input 3SAT(5) formula, defined over the set $\{x_1, \ldots, x_n\}$ of variables, and a set $C_1, \ldots, C_m$ of clauses, where $m = 5n/3$. Our graph G is the $(N \times N)$ grid, where $N = (m+1)(4m+6)$. The set $\mathcal{M}$ of demand pairs consists of three subsets: set $\mathcal{M}_1$ representing the variables of φ, set $\mathcal{M}_2$ representing the clauses, and set $\mathcal{M}_3$ of additional auxiliary pairs. We now define each set of the demand pairs in turn.

Let $I_1, \ldots, I_n$ be any set of mutually disjoint sub-paths of the top row R_1 of the grid, each containing exactly 13 vertices of R_1. For $1 \le j \le n$, let s_j be the vertex lying exactly in the middle of I_j, so s_j is the 7th vertex of I_j from the left. Let t_j and t'_j be the first and the last vertices of I_j, respectively. We then define:

$$\mathcal{M}_1 = \left\{(s_j, t_j), (s_j, t'_j) \mid 1 \le j \le n\right\}.$$

Let $V(j,T)$ be the set of vertices lying on I_j between t_j and s_j (excluding t_j and s_j), and similarly, let $V(j,F)$ be the set of vertices lying on I_j between s_j and t'_j. The intuition is that, since the paths routing the demand pairs are required to be completely disjoint, for each $1 \le j \le n$, we can only route one of the two pairs: (s_j, t_j) or (s_j, t'_j). The routing of the former pair is interpreted as assigning the value ‘F’ to variable x_j, and the routing of the latter pair is interpreted as assigning the value ‘T’ to variable x_j. Intuitively, in the former case, all vertices of $V(j,T)$ will be “blocked” by the path routing (s_j, t_j), while in the latter case all vertices of $V(j,F)$ are “blocked”.

We now turn to define the second set, $\mathcal{M}_2$ of the demand pairs. Let $R = R_{N-4m-6}$ be the row lying within distance $4m+6$ from the bottom row of the grid. Let $y_1, \ldots, y_m$ be any set of m vertices on R, ordered from left to right, so that the distance between every consecutive pair is at least $4m+5$; the distance between y_1 and the left boundary of G is at least $4m+5$, and the distance between y_m and the right boundary of G is at least $4m+5$. Since the grid size is $N \times N$, and $N = (m+1)(4m+6)$, we can find such vertices $y_1, \ldots, y_m$. For each $1 \le h \le m$, vertex y_h will serve as a source vertex corresponding to the clause C_h. We will associate it with three destination vertices, z_h^1, z_h^2, z_h^3, as follows. Assume that $C_h = \ell_{h_1} \vee \ell_{h_2} \vee \ell_{h_3}$. For $1 \le i \le 3$, let x_{h_i} be the variable corresponding to the literal ℓ_{h_i}. If $\ell_{h_i} = x_{h_i}$, then we let z_h^i be some vertex in set $V(h_i, T)$, and otherwise we let z_h^i be some vertex in set $V(h_i, F)$. We select the vertices z_h^i in such a way, that all vertices in set $Z = \left\{z_h^i \mid 1 \le h \le m, 1 \le i \le 3\right\}$ are distinct. Since each variable participates in exactly 5 clauses, and each set $V(j,T), V(j,F)$ contains 5 vertices, we can ensure that all vertices in Z are distinct. We define:

$$\mathcal{M}_2 = \left\{(y_h, z_h^1), (y_h, z_h^2), (y_h, z_h^3) \mid 1 \le h \le m\right\}.$$

Before we define the third set of the demand pairs, we provide some intuition. As mentioned above, we associate each assignment in $\{T, F\}$ to each variable x_j with the routing of either (s_j, t_j) or (s_j, t'_j) along the corresponding segment of the first row. For each clause C_h, if at least one of its literals ℓ_{h_i} is satisfied, we will route the corresponding demand pair (y_h, z_h^i) (we discuss this in more detail later). However, in the No-Instance case, a solution can "cheat" by routing the pairs (s_j, t_j), or (s_j, t'_j) differently: for example, we can route them on a path that goes around some of the sources y_h. In order to avoid this, we create an artificial "bottleneck" by adding a new set of demand pairs. Recall that $v(i,j)$ is a vertex lying in the intersection of row R_i and column C_j of the grid. The last set $\mathcal{M}_3$ of demand pairs contains $8m$ demand pairs $\{a_i, b_i\}_{i=1}^{8m}$, where for $1 \le i \le 8m$, we define $a_i = v(m+4+i, m+1)$, and $b_i = v(m+4+i, N)$. In other words, the ith demand pair in set $\mathcal{M}_3$ consists of the $(m+1)$st and the last vertex of the row R_{m+4+i}. The final set of the demand pairs is $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2 \cup \mathcal{M}_3$. This completes the description of the NDP instance. We now analyze its properties.

Completeness

Assume that the 3SAT(5) formula φ is a Yes-Instance. We show that in this case we can route $9m + n = 16n$ demand pairs. Consider the assignment $f : \{x_1, \ldots, x_n\} \to \{T, F\}$ that satisfies φ.

For each $1 \le i \le n$, if x_i is assigned the value 'T', then we route the pair (s_i, t'_i) via the segment of the row R_1 between these two vertices; if x_i is assigned value F, then we route the pair (s_i, t_i) via the corresponding segment of R_1. For each pair $(a_i, b_i) \in \mathcal{M}_3$, we route (a_i, b_i) via the segment of row R_{m+4+i} connecting these two vertices. Finally, we define the routing of m demand pairs in $\mathcal{M}_2$. For each clause C_h, let ℓ_h^* be any of the literals of C_h that is satisfied by the assignment f, and let $z_h = z_h^i$ be the destination vertex corresponding to ℓ_h^*, so that $(y_h, z_h) \in \mathcal{M}_2$. We will route the pairs $\{(y_h, z_h)\}_{1 \le h \le m}$.

In order to do so, we define three sub-grids of G: B_1 is the sub-grid spanned by rows $R_2, \ldots, R_{m+5}$, and all columns of the grid; B_2 is the sub-grid spanned by rows $R_{m+5}, \ldots, R_{9m+4}$ and columns $C_1, \ldots, C_m$ of the grid; and B_3 spanned by rows $R_{9m+4}, \ldots, R_N$ and all columns of the grid.

For each $1 \le h \le m$, let e_h be the unique vertical edge of the grid incident on vertex z_h, and let z'_h be its other endpoint. Let $S_1 = \{z'_h \mid 1 \le h \le m\}$, so S_1 contains m distinct vertices on the top row of B_1, and let $E' = \{e_h \mid 1 \le h \le m\}$. Let S_2 be the set of m vertices on the top boundary of B_2. Then the vertices of S_2 also lie on the bottom boundary of B_1, and from Observation 1, there is a set $\mathcal{P}_1$ of disjoint paths in B_1, connecting all vertices of S_1 to the vertices of S_2, so that the paths in $\mathcal{P}_1$ are internally disjoint from $V(R_2 \cup R_{m+5})$. Let S_3 be the set of m vertices on the bottom boundary of B_2, and let $\mathcal{P}_2$ be the set of the columns of B_2, so $\mathcal{P}_2$ is a set of m paths, connecting all vertices of S_2 to the vertices of S_3, in graph B_2. Finally, consider the graph B_3, and observe that S_3 is a set of m distinct vertices lying on the top boundary of B_3, while $\{y_h \mid 1 \leq h \leq m\}$ is a set of m vertices lying at L_∞-distance at least $4m+5$ from each other, and from the boundary of B_3. From Theorem 2, we can route any matching between the vertices of S_3 and the vertices of $\{y_h \mid 1 \leq h \leq m\}$ in graph S_3. Let $\mathcal{P}'$ be the set of paths obtained by concatenating $E', \mathcal{P}_1, \mathcal{P}_2$. Then $\mathcal{P}'$ is a set of disjoint paths connecting the vertices of $\{z_h \mid 1 \leq h \leq m\}$ to the vertices of S_3. We denote the vertices of S_3 by $\{z''_1, \ldots, z''_m\}$, where z''_h is the vertex that serves as an endpoint of the path of $\mathcal{P}'$ originating at z_h. We can now construct a set $\mathcal{P}_3$ of disjoint paths in B_3, routing the pairs $\{(y_h, z''_h) \mid 1 \leq h \leq m\}$. By concatenating the paths in $\mathcal{P}'$ and $\mathcal{P}_3$, we obtain the final routing of the pairs in $\{(y_h, z_h) \mid 1 \leq h \leq m\}$. Altogether, we route n demand pairs in $\mathcal{M}_1$, all $8m$ demand pairs in $\mathcal{M}_3$, and m demand pairs in $\mathcal{M}_2$, routing $n + 9m = 16n$ pairs in total.

Soundness

Let $\delta = (1-\epsilon)/200$, where ϵ is the constant from Theorem 15. Assume that φ is a No-Instance, so no assignment can satisfy more than ϵm clauses of φ. We show that the value of the optimal solution of the corresponding NDP problem is at most $(1-\delta)\cdot 16n$. Assume otherwise, and let $\mathcal{P}$ be a set of paths, routing more than $(1-\delta)\cdot 16n$ demand pairs.

Our first observation is that at least $6m$ of the demand pairs in $\mathcal{M}_3$ must be routed by $\mathcal{P}$. Indeed, assume otherwise. Then $\mathcal{P}$ routes at most n pairs in $\mathcal{M}_1$, fewer than $6m$ pairs in $\mathcal{M}_3$, and at most m pairs in $\mathcal{M}_2$. In total, $\mathcal{P}$ routes at most $n + 7m = 38n/3 < (1-\delta)\cdot 16n$ pairs, since $\delta < 1/200$. Therefore, at least $6m$ of the demand pairs in $\mathcal{M}_3$ are routed. Let i be the smallest index, so that (a_i, b_i) is routed in $\mathcal{P}$, and let $P \in \mathcal{P}$ be the path routing (a_i, b_i). Let U be the set of vertices of column C_{m+1} (the column where the sources of the pairs in $\mathcal{M}_3$ lie), that belong to rows $R_1, \ldots, R_{9m+4}$. We use the following observation.

► **Observation 16.** *There is a contiguous sub-path P' of P, containing b_i and some vertex of U, such that P' is internally disjoint from U, and it does not contain any vertex of row $R = R_{N-4m-6}$.*

Proof. If P does not contain any vertex of R, then, since it must contain at least one vertex of U (the vertex a_i), such path P' clearly exist. Therefore, we assume that $P \cap R \neq \emptyset$. Let v be the last vertex of P lying on row R, where we view P as directed from a_i to b_i. Let P^* be the segment of P from v to b_i.

We claim that $P^* \cap U \neq \emptyset$. Indeed, assume otherwise. Let C_j be the column in which v lies and let Q be the segment of C_j from v to the bottom vertex of C_j. If C_j is the last column, then path P^* separates all vertices in $\{a_j\}_{j=1}^{8m}$ from all vertices in $\{t_j\}_{j=i+1}^{8m}$, contradicting the fact that at least $6m$ demand pairs in $\mathcal{M}_3$ are routed, and i is the smallest index for which pair (a_i, b_i) is routed. Therefore, C_j is not the last column. The union of Q and P^* partitions the face defined by $\Gamma(G)$ into a number of sub-faces. Let F_2 be the sub-face containing the top left boundary of the grid, and let F_1 be the union of the remaining sub-faces. Since $P^* \cup Q$ is disjoint from U, all vertices $\{a_j\}_{j=1}^{8m}$ belong to F_2, while the vertices $\{t_j\}_{j=i+1}^{8m}$ belong to F_1. Therefore, all paths of $\mathcal{P}$ routing the pairs in $\mathcal{M}_3$ must intersect Q, while Q contains only $4m + 7$ vertices, a contradiction. We conclude that

$P^* \cap U \neq \emptyset$. Let u be the last vertex on P^* that belongs to U. We can then let P' be the segment of P^* between u and b_i. ◀

Let v^* be the endpoint of P' lying in U, and let $R' = \text{row}(v^*)$. Let I be the sub-path of R' between v^* and the first vertex of row R' (excluding v^*). Since path P' is disjoint from row R, it is easy to see that every path in $\mathcal{P}$ that routes a demand pair in $\mathcal{M}_2$ has to contain at least one vertex of I.

We partition the set of variables of φ into three subsets. Set X_1 contains all variables x_j, such that none of the pairs (s_j, t_j), (s_j, t'_j) is routed by $\mathcal{P}$; X_2 contains all variables x_j, such that one of the pairs (s_j, t_j), (s_j, t'_j) is routed by some path $Q_j \in \mathcal{P}$, and $|Q_j \cap I| \geq 2$. Set X_3 contains all remaining variables. We need the following three observations.

▶ **Observation 17.** $|X_1| \leq 16\delta n$.

Proof. Assume otherwise. Then $\mathcal{P}$ routes fewer than $n(1-16\delta)$ pairs of $\mathcal{M}_1$, at most $8m$ pairs of $\mathcal{M}_2$ and at most m pairs of $\mathcal{M}_3$. In total, this is fewer than $n(1-16\delta)+9m = 16n(1-\delta)$ pairs, a contradiction. ◀

▶ **Observation 18.** $|X_2| \leq 8\delta n$.

Proof. Assume otherwise. As observed above, if $(y, z) \in \mathcal{M}_2$ is routed by $\mathcal{P}$ via some path Q, then $Q \cap I \neq \emptyset$. Since $|I| = m$, the number of pairs in $\mathcal{M}_2$ routed by $\mathcal{P}$ is less than $m - 16\delta n$, and the total number of pairs routed is smaller than $n + (m - 16\delta n) + 8m = 16n(1-\delta)$. ◀

▶ **Observation 19.** *Let $x_j \in X_3$ be some variable, and let $Q \in \mathcal{P}$ be the path originating at s_j. If Q terminates at t_j, then no path of $\mathcal{P}$, routing a demand pair in $\mathcal{M}_2$, may contain any vertex of $V(j, T)$, and if Q terminates at t'_j, then no path of $\mathcal{P}$, routing a demand pair in $\mathcal{M}_2$, may contain any vertex of $V(j, F)$.*

Proof. Assume that Q terminates at t_j: the proof for t'_j is symmetric. Since $|I \cap Q| < 2$, the path Q, together with the sub-path of R_1 between t_j and s_j, forms a closed curve L in the natural drawing of the grid, such that all sources of all pairs in $\mathcal{M}_2$ lie outside L. Therefore, the paths of $\mathcal{P}$ originating from the sources of the demand pairs in $\mathcal{M}_2$ cannot contain the vertices of $V(j, T)$. ◀

We now define an assignment to the variables of φ that satisfies more than ϵm clauses of φ, leading to a contradiction. The assignment is defined as follows. For each variable $x_j \in X_3$, let $Q_j \in \mathcal{P}$ be the path originating at s_j. If Q_j terminates at t_j, then we assign the value 'F' to x_j; otherwise we assign the value 'T' to it. All other variables are assigned arbitrary values.

Let $\mathcal{C}$ be the collection of clauses C_h, such that there is a path originating at vertex y_h in $\mathcal{P}$. It is easy to see that $|\mathcal{C}| \geq m - 16\delta n$, since otherwise $\mathcal{P}$ contains fewer than $n + 8m + (m - 16\delta n) = 16n(1-\delta)$ paths. Let $\mathcal{C}' \subseteq \mathcal{C}$ be the subset of clauses containing the variables of $X_1 \cup X_2$. Since each variable participates in at most 5 clauses, from Observations 17 and 18, $|\mathcal{C}'| \leq 5 \cdot 24\delta n = 120\delta n$. Let $\mathcal{C}^* = \mathcal{C} \setminus \mathcal{C}'$. Then $|\mathcal{C}^*| \geq m - 136\delta n \geq \epsilon m$. We claim that every clause $C_h \in \mathcal{C}^*$ is satisfied by our assignment. Indeed, let $P \in \mathcal{P}$ be the path originating at y_h, and let z_h^i be its other endpoint. Assume that the corresponding literal ℓ_{h_i} corresponds to variable x_j. From our definition of $\mathcal{C}^*$, $x_j \in X_3$. Let $P' \in \mathcal{P}$ be the path originating from s_j. If $z_h^i \in V(j, T)$, then $\ell_{h_i} = x_j$. From Observation 19, P' terminates at t'_j, and variable x_j is assigned the value 'T'. If $z_h^i \in V(j, F)$, then $\ell_{h_i} = \overline{x}_j$. From Observation 19, P' terminates at t_j, and variable x_j is assigned the value 'F'. In either case, the assignment to x_j satisfies the clause C_h.

To conclude, we have shown an efficient algorithm, that, given a 3SAT(5) formula φ, constructs an instance $(G,\mathcal{M})$ of the NDP problem, where G is a grid graph, whose size is polynomial in the size of φ. If φ is a Yes-Instance, then there is a solution of value $16n$ to the NDP instance, and if φ is a No-Instance, then no solution routes more than $16n(1-\delta)$ demand pairs in the NDP instance, for some constant δ. Since it is NP-hard to distinguish the Yes- and the No-instances of 3SAT(5), we conclude that no efficient algorithm can obtain a better than $(1-\delta)$-approximation for NDP on grids, unless $\mathsf{P}=\mathsf{NP}$.

6 Integrality Gap of (LP-flow) for Good Pairs

We prove that the integrality gap of (LP-flow) is $\Omega(n^{1/8})$ even when all of the terminals are far from the grid boundary. We note that the family of instances that we construct here was previously used by Cutler and Shiloah [16], to provide a lower bound on the size of permutation layouts. Our analysis also closely follows theirs.

Given any integer $p>10$, let $k=p^2$ and $N=6k$. We show that the integrality gap of (LP-flow) on the $(N\times N)$ grid G, where all terminals are within distance at least $N/6$ from $\Gamma(G)$ is $\Omega(k^{1/4})=\Omega(n^{1/8})$, where $n=N^2$ is the number of vertices in the grid.

In order to define the demand pairs, we let S be any set of k consecutive vertices on row R_{2k} of G, where all vertices are at distance at least $2k$ from both the left and the right boundary of G, and define a set T of k consecutive vertices on row R_{4k} similarly. We partition the set S into p subsets $S_1,\ldots,S_p$ of p consecutive vertices each, where for $1\leq i,j\leq p$, the jth vertex in set S_i is denoted by $s_{i,j}$. Similarly, we partition T into p subsets $T_1,\ldots,T_p$ of p consecutive vertices each, and for $1\leq i,j\leq p$, the jth vertex in set T_i is denoted by $t_{i,j}$. The set $\mathcal{M}$ of the demand pairs is then:

$$\mathcal{M}=\left\{(s_{i,j},t_{j,i})\mid 1\leq i,j\leq p\right\}.$$

It is easy to see that there is a solution to (LP-flow) of value $k/3$: for each pair $(s_{i,j},t_{j,i})$, we send $1/3$ flow unit on the path P, lying in the union $\mathrm{col}(s_{i,j}),\mathrm{col}(t_{j,i})$ and R_{ip+j}, that connects $s_{i,j}$ to $t_{j,i}$. We next show that the value of any integral solution is $O(k^{3/4})$, thus establishing the integrality gap of $\Omega(k^{1/4})$.

In our analysis we use the notions of graph drawing and graph crossing number. A drawing of a graph H in the plane is a mapping, in which every vertex of H is mapped into a point in the plane, and every edge into a continuous curve connecting the images of its endpoints, such that no three curves meet at the same point, and no curve contains an image of any vertex other than its endpoints. A *crossing* in such a drawing is a point where the images of two edges intersect, and the *crossing number* of a graph H, denoted by $\mathsf{cr}(H)$, is the smallest number of crossings achievable by any drawing of H in the plane. We use the following well-known theorem [2, 23].

▶ **Theorem 20.** *For any graph $H=(V,E)$ with $|E|>7|V|$, $\mathsf{cr}(H)\geq\frac{|E|^3}{29|V|^2}$.*

Let OPT denote the optimal integral solution for the instance $(G,\mathcal{M})$, let $\mathcal{M}^*\subseteq\mathcal{M}$ be the set of the demand pairs routed by OPT, and let $x=|\mathsf{OPT}|$. We define two bipartite graphs. The first bipartite graph, $H=(S,T,E^*)$ is defined over the sets S and T of the source and the destination vertices of $\mathcal{M}$, and it contains an edge $e=(s,t)$ for every pair $(s,t)\in\mathcal{M}^*$. The second graph is $H'=(A,B,E')$, where $A=\{v_1,\ldots,v_p\}$, $B=\{u_1,\ldots,u_p\}$, and E' contains all edges (v_i,u_j), where $(s_{i,j},t_{j,i})\in\mathcal{M}^*$. The following claim is central to our analysis.

▶ **Claim 21.** *There is a drawing of H' with at most $2px$ crossings.*

(a) Before (b) After

Figure 5 Altering the drawing around S_i.

We prove Claim 21 below, after we complete the analysis of the integrality gap here. If $|E'| \leq 14p$, then $|\mathsf{OPT}| = O(\sqrt{k})$ and we are done, so we assume that $|E'| > 14p$. Then from Theorem 20, $\mathsf{cr}(H') \geq \frac{x^3}{116p^2}$, while from Claim 21, $\mathsf{cr}(H') \leq 2px$. Therefore, $x = O(p^{3/2}) = O(k^{3/4})$. It now remains to prove Claim 21.

Proof of Claim 21. Notice that the natural drawing of the grid G, together with the solution OPT to the NDP instance gives a planar drawing φ of the graph H in the plane. For each $1 \leq i \leq p$, let $S'_i \subseteq S_i$ be the set of the sources that have an edge incident to them in E^*, and define $T'_i \subseteq T_i$ similarly. Let $x_i = |S'_i|$ and $y_i = |T'_i|$. For each $1 \leq i \leq p$, if $x_i = 0$, then the vertex v_i of H', corresponding to S_i is an isolated vertex, and we can draw it anywhere. Otherwise, let $s_{i,j} \in S'_i$ be any vertex. We draw v_i at $\varphi(s_{i,j})$. Let $I(i)$ be the segment of row R_{2k} containing the vertices of S_i, and no other vertices. Let L_i be a very thin strip (of height $1/10$) around the segment $I(i)$ (see Figure 5). We alter the drawings of all edges in E^*, originating at the vertices of S'_i, so that they now originate at $\varphi(s_{i,j})$, by re-routing them inside the strip L_i. Since the number of paths in OPT containing the vertices of S_i is bounded by p, it is easy to do so, by introducing at most px_i crossings. We perform the same transformation for the sets T_i of destination vertices, and obtain a drawing of the graph H' with at most $p\sum_{i=1}^{p}(x_i + y_i) \leq 2px$ crossings. ◀

7 Approximation Algorithm for EDP on Wall Graphs

In this section we show that the algorithm from Section 4 can be adapted to give an $O(n^{1/4} \cdot \log n)$-approximation for EDP on wall graphs of width and height $N = \Omega(\sqrt{n})$. In order to construct a wall W of height h and width r (or an $(h \times r)$- wall), we start from a grid of height h and width $2r$. Consider some column C_j of the grid, for $1 \leq j \leq r$, and let $e^j_1, e^j_2, \ldots, e^j_{h-1}$ be the edges of C_j, in the order of their appearance on C_j, where e^j_1 is incident on $v(1, j)$. If j is odd, then we delete from the graph all edges e^j_i where i is even. If j is even, then we delete from the graph all edges e^j_i where i is odd. We process each column C_j of the grid in this manner, and in the end delete all vertices of degree 1. The resulting graph is a wall of height h and width r, that we denote by W (See Figure 1).

Let E_1 be the set of edges of W that correspond to the horizontal edges of the original grid, and let E_2 be the set of the edges of W that correspond to the vertical edges of the original grid. The sub-graph of W induced by E_1 is a collection of h node-disjoint paths, that we refer to as the rows of W. We denote these rows by $R_1, \ldots, R_h$, where for $1 \leq i \leq h$, R_i is incident on $v(i, 1)$. Let V_1 denote the set of all vertices in the first row of W, and V_h the set of vertices in the last row of W. There is a unique set $\mathcal{C}$ of r node-disjoint paths, where each path $C \in \mathcal{C}$ starts at a vertex of V_1, terminates at a vertex of V_h, and is internally disjoint from $V_1 \cup V_h$. We refer to these paths as the columns of W. We order these columns from left to right, and denote by C_j the jth column in this ordering, for $1 \leq j \leq r$. The sub-graph $\Gamma(W) = R_1 \cup C_1 \cup R_h \cup C_r$ of W is a simple cycle, that we call the boundary of W.

For every vertex $v \in V(W)$, we let $\mathrm{col}(v)$ and $\mathrm{row}(v)$ denote the column and the row of W to which v belongs. As before, for a pair $u, v \in V(W)$ of vertices, we define:

$$d_\infty(u,v) = \max\left\{|\,\mathrm{col}(v) - \mathrm{col}(u)|, |\,\mathrm{row}(v) - \mathrm{row}(u)|\right\},$$

and for a vertex v and a subset $U \subseteq V(W)$ of vertices, we let $d_\infty(v,U) = \min_{u\in U}\{d_\infty(u,v)\}$.

Assume now that we are given an $(N \times N)$-wall graph $G = (V,E)$, so $n = |V| = \Theta(N^2)$, and a collection $\mathcal{M} = \{(s_i,t_i)\}_{i=1}^k$ of demand pairs. As before, we say that a demand pair (s_i,t_i) is bad if both $d_\infty(s_i,\Gamma(G)), d_\infty(t_i,\Gamma(G)) \le 4\sqrt{N}+4$, and we say that it is good otherwise. Let $\mathcal{M}', \mathcal{M}'' \subseteq \mathcal{M}$ denote the sets of the good and the bad demand pairs in $\mathcal{M}$, respectively. We find an approximate solution to each of the two sub-problems, defined by $\mathcal{M}'$ and $\mathcal{M}''$, separately, and take the better of the two solutions.

The algorithm for the bad pairs remains exactly the same as the algorithm from Section 4.2. We now focus on the problem defined by the set $\mathcal{M}'$ of the good pairs. Let G' be the $(N \times N)$-grid obtained from G, by contracting, for each $1 \le i,j \le N$, the unique edge $e \in R_i \cap C_j$, and consider the NDP problem instance $(G', \mathcal{M}')$. Any collection $\mathcal{P}'$ of node-disjoint paths in G', routing a subset $\tilde{\mathcal{M}} \subseteq \mathcal{M}'$ of the demand pairs immediately gives a collection $\mathcal{P}''$ of edge-disjoint paths in G, routing the same subset of the demand pairs. Moreover, it is easy to see that there is an LP-solution to (LP-flow) on instance $(G', \mathcal{M}')$ of value $\mathsf{OPT}'/2$, where OPT' is the optimal solution for the EDP instance $(G, \mathcal{M}')$. Indeed, for every path $P \in \mathsf{OPT}'$, we simply set $f(P') = 1/2$, where P' is the path of G' corresponding to the path P of G, and for every demand pair (s_j,t_j) routed by OPT', we set $x_j = 1/2$. It is immediate to verify that this is a feasible solution to (LP-flow) on NDP instance $(G', \mathcal{M}')$, of value $\mathsf{OPT}'/2$. We then use the algorithm from Section 4.1 to find an $O(n^{1/4} \cdot \log n)$-approximation solution to $(G', \mathcal{M}')$, which in turn gives an $O(n^{1/4} \cdot \log n)$-approximation solution to the EDP instance $(G, \mathcal{M}')$.

References

1 Alok Aggarwal, Jon Kleinberg, and David P. Williamson. Node-disjoint paths on the mesh and a new trade-off in VLSI layout. *SIAM J. Comput.*, 29(4):1321–1333, February 2000.
2 M. Ajtai, V. Chvátal, M.M. Newborn, and E. Szemerédi. Crossing-free subgraphs. In Gert Sabidussi Peter L. Hammer, Alexander Rosa and Jean Turgeon, editors, *Theory and Practice of Combinatorics A collection of articles honoring Anton Kotzig on the occasion of his sixtieth birthday*, volume 60 of *North-Holland Mathematics Studies*, pages 9–12. North-Holland, 1982.
3 Matthew Andrews. Approximation algorithms for the edge-disjoint paths problem via Raecke decompositions. In *Proceedings of IEEE FOCS*, pages 277–286, 2010.
4 Matthew Andrews, Julia Chuzhoy, Venkatesan Guruswami, Sanjeev Khanna, Kunal Talwar, and Lisa Zhang. Inapproximability of edge-disjoint paths and low congestion routing on undirected graphs. *Combinatorica*, 30(5):485–520, 2010.
5 Matthew Andrews and Lisa Zhang. Hardness of the undirected edge-disjoint paths problem. In *STOC*, pages 276–283. ACM, 2005.
6 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45:501–555, May 1998.
7 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: a new characterization of np. *J. ACM*, 45:70–122, January 1998.
8 Yonatan Aumann and Yuval Rabani. Improved bounds for all optical routing. In *Proceedings of the sixth annual ACM-SIAM symposium on Discrete algorithms*, SODA'95, pages 567–576, Philadelphia, PA, USA, 1995. Society for Industrial and Applied Mathematics.

9 Chandra Chekuri and Julia Chuzhoy. Half-integral all-or-nothing flow. Unpublished Manuscript.
10 Chandra Chekuri and Alina Ene. Poly-logarithmic approximation for maximum node disjoint paths with constant congestion. In *Proc. of ACM-SIAM SODA*, 2013.
11 Chandra Chekuri, Sanjeev Khanna, and F Bruce Shepherd. Edge-disjoint paths in planar graphs. In *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, pages 71–80. IEEE, 2004.
12 Chandra Chekuri, Sanjeev Khanna, and F. Bruce Shepherd. Multicommodity flow, well-linked terminals, and routing problems. In *Proc. of ACM STOC*, pages 183–192, 2005.
13 Chandra Chekuri, Sanjeev Khanna, and F. Bruce Shepherd. An $O(\sqrt{n})$ approximation and integrality gap for disjoint paths and unsplittable flow. *Theory of Computing*, 2(1):137–146, 2006.
14 Julia Chuzhoy. Routing in undirected graphs with constant congestion. In *Proc. of ACM STOC*, pages 855–874, 2012.
15 Julia Chuzhoy and Shi Li. A polylogarithimic approximation algorithm for edge-disjoint paths with congestion 2. In *Proc. of IEEE FOCS*, 2012.
16 M. Cutler and Y. Shiloach. Permutation layout. *Networks*, 8:253–278, 1978.
17 R. Karp. On the complexity of combinatorial problems. *Networks*, 5:45–68, 1975.
18 Ken-Ichi Kawarabayashi and Yusuke Kobayashi. An o(log n)-approximation algorithm for the edge-disjoint paths problem in eulerian planar graphs. *ACM Trans. Algorithms*, 9(2):16:1–16:13, March 2013.
19 Jon M. Kleinberg. An approximation algorithm for the disjoint paths problem in even-degree planar graphs. In *FOCS'05*, pages 627–636, 2005.
20 Jon M. Kleinberg and Éva Tardos. Disjoint paths in densely embedded graphs. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 52–61, 1995.
21 Jon M. Kleinberg and Éva Tardos. Approximations for the disjoint paths problem in high-diameter planar networks. *J. Comput. Syst. Sci.*, 57(1):61–73, 1998.
22 Stavros G. Kolliopoulos and Clifford Stein. Approximating disjoint-path problems using packing integer programs. *Mathematical Programming*, 99:63–87, 2004.
23 Frank Thomson Leighton. *Complexity Issues in VLSI: Optimal Layouts for the Shuffle-exchange Graph and Other Networks*. MIT Press, Cambridge, MA, USA, 1983.
24 Harald Räcke. Minimizing congestion in general networks. In *Proc. of IEEE FOCS*, pages 43–52, 2002.
25 Prabhakar Raghavan and Clark D. Tompson. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7:365–374, December 1987.
26 Satish Rao and Shuheng Zhou. Edge disjoint paths in moderately connected graphs. *SIAM J. Comput.*, 39(5):1856–1887, 2010.
27 N. Robertson and P. D. Seymour. Outline of a disjoint paths algorithm. In *Paths, Flows and VLSI-Layout*. Springer-Verlag, 1990.
28 Neil Robertson and Paul D. Seymour. Graph minors. VII. disjoint paths on a surface. *J. Comb. Theory, Ser. B*, 45(2):212–254, 1988.
29 Neil Robertson and Paul D Seymour. Graph minors. XIII. the disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110, 1995.
30 Loïc Seguin-Charbonneau and F. Bruce Shepherd. Maximum edge-disjoint paths in planar graphs with congestion 2. In *Proceedings of the 2011 IEEE 52Nd Annual Symposium on Foundations of Computer Science*, FOCS'11, pages 200–209, Washington, DC, USA, 2011. IEEE Computer Society.