THE SHORTEST MULTIPATHS PROBLEM IN A CAPACITATED DENSE CHANNEL

CÉDRIC BENTZ[†], MARIE-CHRISTINE COSTA[*], CHRISTOPHE PICOULEAU[*§], AND MARIA ZRIKEM[*]

Abstract. In this paper, we present a simple polynomial-time algorithm solving the shortest multipaths problem in particular grid graphs called *dense channels*. Our work extends the results of Formann et al. [5], by considering arbitrary horizontal and vertical capacities.

Keywords: combinatorial optimization, capacitated multipaths problem, grid graph, polynomial-time algorithm.

1. Introduction

Routing problems frequently arise in industrial settings. In particular, there have been a lot of papers concerning routing problems in planar graphs, and especially in grids, because of their application in the design of grid-like VLSI circuits (for instance, see [9, 10], [12, pp. 625–712], and the papers cited in the *Grid graphs* section in [13, pp. 1323–1325]). Moreover, papers often deal with the unit capacity case, i.e., the edge-disjoint paths problem (EDP): given an undirected graph and K vertex pairs (s_k, t_k), decide whether all the s_k's can be linked to the corresponding t_k's by K edge-disjoint paths. For the case of a grid where the s_k's and t_k's are on its boundary, Frank gives a polynomial-time algorithm for this problem [6], but if they are allowed to lie anywhere in the grid, Marx shows that it becomes NP-complete even in eulerian grids [11].

Two kinds of optimization problems can be associated with EDP.

First, one can ask what is the maximum number of pairs (s_k, t_k) that can be routed along edge-disjoint paths (problem MEDP). In grids, one can only expect, in polynomial time, to solve special cases [1, 3] or to design approximation algorithms [8].

Second, one can ask for a feasible routing with minimum total paths length. More formally, given an undirected graph $G = (V, E)$ with integral

[†]LRI, UNIVERSITÉ PARIS-SUD AND CNRS, 91405 ORSAY CEDEX, FRANCE.
[*]CEDRIC-CNAM, 292, RUE SAINT MARTIN 75141 PARIS CEDEX 03, FRANCE.
[§]CORRESPONDING AUTHOR: CHRISTOPHE PICOULEAU, CNAM, CHAIRE DE RECHERCHE OPÉRATIONNELLE, 292, RUE SAINT MARTIN 75141 PARIS CEDEX 03, FRANCE. E-MAIL: CHP@CNAM.FR

capacities on the edges and K pairs (source s_k, sink t_k) of *terminal* vertices, the *shortest multipaths problem* (SMPP) consists of linking each pair (s_k, t_k) by a path with respect to the capacity constraints (i.e., the number of paths routed through each edge must be at most the capacity of this edge), such that the sum of the lengths of the K paths is minimized [4], where the length of a path is the *number* of edges it contains. An instance of SMPP is called *feasible* if there exists a way of routing the K paths without violating any capacity constraint, i.e., if there exists at least one feasible solution. A variant of this problem is to find a routing minimizing the length of the longest path. Both this variant and SMPP are $\mathcal{NP}$-hard in planar graphs, even if all the terminals lie on the outer face, the maximum degree is four and all the vertices not on the outer face have even degrees [2], although EDP is polynomial-time solvable in this case, even with unbounded degrees [7]. Thus, it seems that even interesting polynomial-time solvable cases need to have a very special structure.

A *dense channel* is a rectilinear grid with K terminal pairs and K vertical lines, in which all the terminals are distinct, all the sources lie on the uppermost horizontal line and all the sinks on the lowermost one (see Figure 1). Assume that each horizontal (resp. vertical) edge of the grid is valued by the same positive integral capacity C_h (resp. C_v). Formann et al. study the case of a dense channel where $C_h = C_v = 1$ [5]: in this case, it follows from a result of Frank in [6] that the problem has no solution (i.e., no instance is feasible), unless either all the terminal pairs can be routed vertically (trivial case) or there is an additional vertical line on the right or on the left of the grid. Thus, for the non trivial case, the authors add a vertical line without terminal on one side of the grid and prove that SMPP is then polynomial-time solvable.

In this paper, we give a greedy polynomial-time algorithm to solve SMPP in dense channels when C_v and C_h are arbitrary (except for the case $C_h = C_v = 1$, settled in [5]). The main feature of our algorithm is that it routes each pair (s_k, t_k) along a shortest path (i.e., unlike the case $C_h = C_v = 1$, no additional vertical line is needed). Obviously, the total length is thus minimum.

The paper is organized as follows. Definitions and properties used throughout the paper are given in Section 2. In Section 3, we solve SMPP in dense channels with $C_v = 1$ and $C_h = 2$. Then, in Section 4, we show how to generalize the results of Section 3 to the case where $C_v = 1$ and C_h is even. Section 5 deals with the case where $C_v = 1$ and C_h is odd. Eventually, using results from Sections 3 and 4, Section 6 settles the case where $C_v \geq 2$.

2. Preliminaries

Let m be the number of horizontal lines, or simply *lines*, of the grid (see [6] for a formal definition of a grid). The first line is the uppermost one. Each one of the K pairs (s_k, t_k) is called a *net*. The grid has $2K$ terminals and K vertical lines (or *columns*), the first and the K^{th} columns being respectively the leftmost and rightmost ones.

We assume without loss of generality that the nets are numbered such that the column of t_k is the k^{th} one, for each $k \in \{1, \ldots, K\}$. Given a net $l_k = (s_k, t_k)$, we denote by $source(l_k)$ the column of s_k, and say that l_k is *left* if $source(l_k) > k$, *right* if $source(l_k) < k$, and *straight* otherwise. For example, in Figure 1, the net (s_3, t_3) is left and the net (s_5, t_5) is right.

We call vertical (resp. horizontal) *strip* the region between two consecutive vertical (resp. horizontal) lines: let v_j denote the j^{th} vertical strip, lying between the j^{th} and the $j + 1^{st}$ columns. Since we assume that each horizontal edge is valued by C_h, the *capacity of* v_j, i.e., the sum of the capacities of the horizontal edges in v_j, is mC_h for each j. The *density* [5] (or *congestion* [6]) d_j of a vertical strip v_j is the number of nets "crossing" it

$$d_j = |\{\text{net } l_k \text{ s. t. } k \leq j < source(l_k) \text{ or } source(l_k) \leq j < k\}|$$

An example is given in Figure 1, where the densities are indicated by numbers lying in the corresponding vertical strips. The *density of the grid* is $d = \max_{j \in \{1,\ldots,K-1\}} \{d_j\}$ and is at most K. Lemma 1 details the three different cases for the densities of two consecutive vertical strips.

Lemma 1. *Given a vertical strip v_j of density d_j, d_{j+1} is equal to*

- *$d_j + 2$ if and only if the two nets having their source and sink on the $j + 1^{st}$ column are respectively a right and a left net;*
- *$d_j - 2$ iff the two nets having their source and sink on the $j + 1^{st}$ column are respectively a left and a right net;*
- *d_j iff there is a straight net on the $j + 1^{st}$ column or the two nets having a terminal on this column are both left or both right nets.*

Proof. Easily follows from the definition of the density. □

Lemma 1 and the fact that $d_1 \in \{0, 2\}$ imply that the density is always even. An obvious necessary condition for the feasibility of a given instance is that the capacity of each vertical strip v_j is at least the number of nets that have to cross it, i.e., its density

$$\text{for all } j,\ mC_h \geq d_j, \text{ i.e., } m \geq \lceil \tfrac{d}{C_h} \rceil \tag{1}$$

We are going to show that, whenever C_h is even or $C_v \geq 2$, the condition $m = \lceil \frac{d}{C_h} \rceil$ is sufficient for routing the K nets along shortest paths.

We define a *dense region* as a set of consecutive vertical strips of maximal density, this set being maximal in the sense of inclusion. A dense region will be denoted by $[a, b]$, where a (resp. b) is the leftmost (resp. rightmost) column adjacent to this region: $d_j = d$ for $j \in \{a, \ldots, b-1\}$ and

$$d_{a-1} = d_b = d - 2 = d_a - 2 = d_{b-1} - 2 \tag{2}$$

In Figure 1, we have $d = 4$ and a dense region $[a, b] = [2, 4]$.

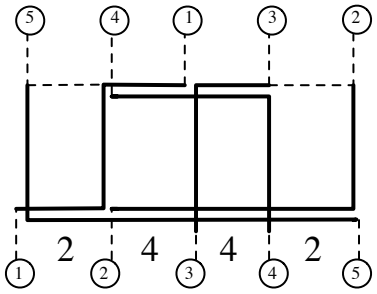

FIGURE 1. A dense channel with 2 lines and 5 nets

In order to avoid the trivial case, we assume throughout the rest of the paper that there exists at least one non straight net.

3. SMPP WITH A CAPACITY 2 ON EACH LINE

For $C_h = 2$, (1) becomes $m \geq \frac{d}{2}$, so we assume in this section that $m = \frac{d}{2}$.

3.1. **Description of the algorithm.** In this part, we give the greedy algorithm solving SMPP in dense channels with $C_v = 1$ and $C_h = 2$. The main idea of this algorithm is to route each net along a shortest path. It works in m iterations, where the i^{th} iteration corresponds to the construction of the paths on the i^{th} horizontal line. *Pulling a net* to the right (resp. to the left) means routing its path horizontally to the right (resp. to the left) on the current line. The algorithm can be informally described as follows:

- We construct the paths line by line, from the uppermost to the lowermost one: at each line, we obtain a new instance of SMPP;
- For each line, we compute the residual densities (i.e., the densities of the new instance of SMPP) and we update the dense regions;
- For each dense region $[a, b]$
  (1) We begin with a *left-pulling* phase, in which, starting from b, we sequentially choose one left net and pull it to the left;
  (2) Then, we proceed to a *right-pulling* phase in a symmetric way.

After the i^{th} iteration ends, for each j, there exists a net whose partial path is stopped at column j on the i^{th} line: let $N_{i+1,j}$ denote this net. Then, j can be viewed as the current source of the net $N_{i+1,j}$ in the new instance of SMPP obtained at iteration $i+1$. We call a column j *unused* at an iteration i if no net is pulled to or from it during this iteration. We now give formally the procedure 1-2-ShortestMultiPaths that solves SMPP.

PROCEDURE 1-2-ShortestMultiPaths
For k **from** 1 **to** K **do**
    $N_{1,source(l_k)} := k$; *// initially, the column of the source of the net l_k is source(l_k)*
EndFor
For i **from** 1 **to** m **do**
    For each dense region $[a, b]$ **do**
        Left-pulling_phase($[a, b], i$);
        Right-pulling_phase($[a, b], i$);
    EndFor
    Update the residual densities;
    Route vertically all the nets to the next horizontal line; *// pull down the nets*
    For each unused column j **do**
        $N_{i+1,j} := N_{i,j}$; *// the net having its source on the j^{th} column is unchanged*
    EndFor
EndFor

The left-pulling and right-pulling phases are quite symmetric, so we give only the first one. One can obtain the right-pulling phase by replacing "a" by "b", "left" by "right" and "$>$" by "$<$".

PROCEDURE Left-pulling_phase($[a, b], i$)
$k := N_{i,b}$; *// let l_k be the net whose source lies on b*
a_is_reached := **false**;
While l_k is a left net **and** $\neg$ a_is_reached **do**
    If $k > a$ **then** *// k is the column of t_k*
        Pull l_k to the left until k is reached;
        $N_{i+1,k} := k$; *// l_k becomes a straight net*
        $k := N_{i,k}$; *// the current net is now the net whose source lies on k*
    Otherwise
        Pull l_k to the left until a is reached;
        $N_{i+1,a} := k$; *// the source of l_k now lies on a*
        a_is_reached := **true**;
    EndIf
EndWhile

An example of a routing given by our algorithm is drawn in Figure 1.

3.2. **Correctness.** In this part, we prove by induction that, when $C_h = 2$, $C_v = 1$ and $m = \frac{d}{2}$, the algorithm given in Section 3.1 provides a feasible solution for SMPP where each net is routed along a shortest path. We consider an arbitrary iteration i, and assume that, at the beginning of this iteration, we are given an *initial instance*, i.e., an instance of SMPP in a dense channel of density $d - 2(i-1)$ and with $m - i + 1$ lines (obviously, this is true for $i = 1$). Then, we prove that, at the end of the iteration, we obtain a *reduced instance*, i.e., a valid instance of SMPP in a dense channel of density $d - 2i$ and with $m - i$ lines.

Fact 1. *In a left-pulling phase, no net whose current source is in a dense region $[a, b]$ is routed such that its new source is no longer in $[a, b]$, since we stop its path either when a is reached or before a is reached.*

Fact 2. *In a left-pulling phase, no left net (s_k, t_k) is routed such that it becomes a right net in the next iteration, since we stop its path either when k is reached or before k is reached.*

Fact 3. *For each net selected in a left-pulling phase, its path crosses only vertical strips that have to be crossed. This is because only left nets are pulled to the left, and, from Fact 2, no left net becomes right after being pulled.*

Fact 4. *Facts 1, 2 and 3 hold symmetrically for the right-pulling phase.*

Lemma 2. *The routing constructed at iteration i is valid.*

Proof. Let us consider the left-pulling phase (the two phases are independent and symmetric). From (2) and Lemma 1, given a dense region $[a, b]$, the net l_c whose current source lies on b is a left net, so we can pull it to the left. Its path stops either on a (if $c \leq a$) or on c (if $a < c < b$), where c is the column of its sink. In the latter case, $d_{c-1} = d_c$, thus we know from Lemma 1 that we can continue the left-pulling phase with a left net having its source on c, and so on until a is reached. □

Lemma 3. *At iteration i, the density of the reduced instance is $d - 2i$.*

Proof. In the left-pulling phase, given the current dense region $[a, b]$, we start from b, then we pull nets to the left until we reach a. From the proof of Lemma 2, each net selected in this phase starts where the previous selected one ended. Thus, from Facts 3 and 4, the density of each vertical strip in a dense region is decreased by 2 when the i^{th} iteration ends. From Facts 1 and 4, the densities of the other vertical strips remain unchanged, so no residual density exceeds $d - 2i$. □

Lemma 3 implies that, at the end of the m^{th} iteration, the residual density is $d-2m=0$: hence, each path has reached the column of its corresponding sink, and we are done. Thus, our algorithm provides a feasible solution.

Lemma 4. *The paths produced by the algorithm are shortest paths.*

Proof. Follows immediately from Facts 3 and 4, no net being pulled up. □

Theorem 1. *If $m = \frac{d}{2}$, the algorithm* 1-2-ShortestMultiPaths *outputs a feasible solution for SMPP in which all the nets are routed along shortest paths. Moreover, it can be implemented to run in $O(mK)$.*

Proof. The first part directly follows from Lemmas 2, 3 and 4.

Now, let us prove the second part. At each iteration, we must compute the residual densities. In fact, using Lemma 1, we compute the densities once (before the first iteration) in time $O(K)$, and then, at each iteration, Lemma 3 implies that we just have to decrease d_j by two for each v_j in a dense region. Updating the dense regions can be done in $O(K)$ by going through the vertical strips once, and the left-pulling and right-pulling phases run in $O(b-a)$ for each dense region $[a,b]$. There are m iterations, so our algorithm runs in $O(mK)$, which is linear in the size of the grid. □

4. SMPP with an even capacity on each line

In this section, we show how to generalize the results of Section 3 to the case where C_h is even and $C_v = 1$. From Section 2, a necessary condition for the feasibility of an instance is (1), i.e., $m \geq \lceil \frac{d}{C_h} \rceil$. Therefore, in the following of Section 4, we assume that $m = \lceil \frac{d}{C_h} \rceil$.

Let $C_h = 2p$, $1 \leq p \leq \frac{d}{2}$. We begin by transforming the grid with m lines and a capacity equal to $2p$ on each horizontal edge into a new grid with mp lines and a capacity equal to 2 on each horizontal edge. The capacity of each vertical edge is 1 in both grids. It is easily seen that the new grid still satisfies (1), and thus we can apply the algorithm given in Section 3.1 and obtain a feasible set of shortest paths. We use this set of paths to construct a solution for the initial grid. For each i, the path of each net in the initial grid is routed through the horizontal edge of the i^{th} line lying in the j^{th} vertical strip if and only if, in the new grid, it is routed through an horizontal edge of this vertical strip lying on a line between the $(i-1)p+1^{st}$ and the ip^{th} one. In other words, for each net, its path in the initial grid enters (resp. exits) the i^{th} line where it enters (resp. exits) the $(i-1)p+1^{st}$ line (resp. ip^{th} line) in the new grid.

The following two facts hold in the initial grid, since otherwise they would not hold in the new grid anymore: no two paths share a vertical edge, and

the paths we obtain are shortest paths. Moreover, by the way we pack the lines of the new grid together, at most $2p$ paths are routed through each horizontal edge of the initial grid (i.e., at most 2 paths for each one of the p corresponding horizontal edges in the new grid).

5. SMPP with an odd capacity on each line

In this section, we show that the case where $C_v = 1$ and C_h is an odd number larger than 2 can be reduced to the case of Section 4. Consider a feasible routing: given any horizontal edge, an even number of paths is routed through it. Indeed, for each path routed through it "from left to right", there is necessarily a path routed through it "from right to left", since $C_v = 1$ and there are as many paths as columns. So, we do not modify the feasibility of the instance if we replace C_h (odd) by $C_h - 1$ (even). Hence, either $m \geq \lceil \frac{d}{C_h - 1} \rceil$ and we can apply the results of Section 4, or there is no solution.

6. SMPP with a capacity at least two on each column

Sections 3, 4 and 5 settle the case where $C_v = 1$ and $C_h \geq 2$. In this section, we deal with the case where $C_v \geq 2$. In fact, we start dealing with the case $C_v = 2$ and then we show how to settle the case $C_v > 2$.

6.1. SMPP with a capacity two on each column and one on each line.

$C_h = 1$, so (1) becomes $m \geq d$. In the following of Section 6.1, we assume that $m = d$. From Lemma 1, d is even, thus m is even. We successively proceed to the left-pulling and right-pulling phases on the $2i - 1^{st}$ and $2i^{th}$ lines respectively, for i from 1 to $\frac{m}{2}$. Then, similar arguments to those used in Section 3.2 show that we obtain a feasible set of shortest paths, and thus an optimal solution for SMPP. The only noticeable difference with the case studied in Section 3.2 occurs, for each i, after the left-pulling phase ends on the $2i - 1^{st}$ line: for each dense region $[a, b]$, there is a right net whose source lies on column a, so one right net *and* one left net (or a left net which has become straight on the $2i - 1^{st}$ line) will be routed through the vertical edge of column a lying between the $2i - 1^{st}$ and $2i^{th}$ lines.

6.2. SMPP with a capacity two on each column and at least two on each line.

We borrow ideas from Section 4. We transform the initial grid (which satisfies the necessary condition (1)) into a new grid where the vertical edges are also valued by 2, the horizontal edges by 1, and having mC_h lines. If mC_h is odd then $mC_h > d$, since d is even and (1) holds (i.e., $mC_h \geq d$). Thus, in this case, the new grid could have d lines only. However, for the sake of simplicity, we assume that it has mC_h lines and

that, from the d^{th} line to the $mC_h{}^{th}$ line, the nets are routed vertically. We use the results of Section 6.1 to obtain an optimal solution for SMPP in this new grid: for each i, the path of each net in the initial grid enters (resp. exits) the i^{th} line where it enters (resp. exits) the $(i-1)C_h + 1^{st}$ line (resp. $iC_h{}^{th}$ line) in the new grid. An analysis quite similar to the one of Section 4 shows that this provides a solution for SMPP in the initial grid where all the nets are routed along shortest paths.

6.3. **SMPP with a capacity strictly greater than two on each column.** This case can be reduced to the case where $C_v = 2$, because, in the routings constructed in the two previous sections, at most two paths share any vertical edge. Thus, this case can also be solved in polynomial time.

7. Conclusion

We have solved in polynomial time SMPP in dense channels with arbitrary horizontal and vertical capacities, extending the results in [5], where only unit capacities are considered. For $C_h = 2$ and $C_v = 1$ and for $C_h = 1$ and $C_v = 2$, the running time of our greedy algorithm is linear in the size of the grid. Moreover, we have proved that, if $\max(C_h, C_v) > 1$, whenever there is a solution, there always exists one where all the nets are routed along shortest paths, although this is not true if $C_h = C_v = 1$. This also shows that our algorithm solves the variant where one wants to minimize the length of the longest path.

References

[1] C. Bentz, M.-C. Costa and F. Roupin. Maximum integer multiflow and minimum multicut problems in two-sided uniform grid graphs. To appear in Journal of Discrete Algorithms (2005).

[2] U. Brandes, G. Neyer and D. Wagner. Edge-disjoint paths in planar graphs with minimum total length. Technical report, Konstanzer Schriften in Mathematik und Informatik 19, Universität Konstanz (1996), 11 p.

[3] W.-T. Chan and F.Y.L. Chin. Efficient algorithms for finding the maximum number of disjoint paths in grids. Journal of Algorithms 34 (2000), pp. 337–369. Elsevier.

[4] M.-C. Costa, A. Hertz and M. Mittaz. Bounds and heuristics for the Shortest Capacited Paths Problem. Journal of Heuristics vol. 8(4) (2002), pp. 449–465. Kluwer.

[5] M. Formann, D. Wagner and F. Wagner. Routing through a dense channel with minimum total wire length. Journal of Algorithms 15 (1993), pp. 267–283. Elsevier.

[6] A. Frank. Disjoint paths in a rectilinear grid. Combinatorica 2 (1982), pp. 361–371. Springer-Verlag.

[7] A. Frank. Edge-disjoint paths in planar graphs. Journal of Combinatorial Theory, Series B 39 (1985), pp. 164–178. Elsevier.

[8] J. Kleinberg and E. Tardos. Disjoint paths in densely embedded graphs. Proceedings 36^{th} IEEE FOCS (1995), pp. 52–61. IEEE Computer Society.

[9] B. Korte, L. Lovász, H. J. Prömel and A. Schrijver (editors). *Paths, Flows and VLSI-Layout*. Algorithms and combinatorics 9 (1990), 383 p. Springer-Verlag.

[10] T. Lengauer. Combinatorial algorithms for integrated circuit layout. Teubner, Stuttgart (1990), 697 p. Wiley.

[11] D. Marx. Eulerian disjoint paths problem in grid graphs is NP-complete. Discrete Applied Mathematics 143(1–3), pp. 336–341 (2004). Elsevier.

[12] R. Möhring, D. Wagner and F. Wagner. VLSI network design: a survey. In M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser (editors), Handbooks in Operations Research/Management Science, Volume on Networks (1995). Elsevier.

[13] A. Schrijver. Combinatorial Optimization - Polyhedra and Efficiency. Algorithms and Combinatorics 24 (2003). Springer-Verlag.