Maximum integer multiflow and minimum multicut problems in two-sided uniform grid graphs

Cédric Bentz [a,*] and

[a] *LRI*
Université Paris-Sud and CNRS 91405 Orsay Cedex, France

Marie-Christine Costa [b]

[b] *CEDRIC-CNAM*
292 rue Saint-Martin 75141 Paris Cedex 03, France

Frédéric Roupin [c]

[c] *CEDRIC-CNAM-IIE*
18, Allée Jean Rostand, 91025 Evry Cedex, France

Abstract

In this paper, we deal with the maximum integer multiflow and the minimum multicut problems in rectilinear grid graphs with uniform capacities on the edges. The first problem is known to be $\mathcal{NP}$-hard when any vertex can be a terminal, and we show that the second one is also $\mathcal{NP}$-hard. Then, we study the case where the terminals are located in a two-sided way on the boundary of the outer face. We prove that, in this case, both problems are polynomial-time solvable. Furthermore, we give two efficient combinatorial algorithms using a primal-dual approach. Our work is based on previous results concerning related decision problems.

Key words: minimum multicut, maximum integer multiflow, grid graph.

* Corresponding author. Phone: +33 (0) 1 69 15 31 06.
Email addresses: `cedric.bentz@lri.fr` (Cédric Bentz), `costa@cnam.fr` (Marie-Christine Costa), `roupin@iie.cnam.fr` (Frédéric Roupin).

1 Introduction

1.1 Definitions

The maximum integer multiflow problem (MaxIMF) and the minimum multicut problem (MinMC) are difficult problems that arise, in particular, in the field of telecommunications. For both problems, we are given an edge-capacitated graph and a list of K pairs (source s_k, sink t_k) of *terminal* vertices. Each pair (s_k, t_k) defines a *net* (or a *commodity*), s_k and t_k being *mates*. MaxIMF consists in maximizing the sum of the integral flows of each commodity (from s_k to t_k), subject to capacity and flow conservation requirements. MinMC is to find a minimum weight set of edges whose removal separates s_k from t_k for each one of the K nets (then, each net (s_k, t_k) is said to be *cut* or *disconnected*). These problems can be formulated as two integer linear programs whose continuous relaxations are dual [4; 11]. The *maximum fractional multiflow problem* corresponds to the relaxation of MaxIMF where the flows are allowed to be fractional. Moreover, when all the edges have unit capacities, MaxIMF turns into the *maximum edge disjoint paths problem* (MaxEDP). A decision problem related to MaxEDP is the *edge disjoint paths problem* (EDP): given K nets in a graph, decide whether all can be routed along edge-disjoint paths.

In this paper, we study these problems in particular graphs, the rectilinear grids with uniform capacities. We shall denote by MaxIMFUG and MinMCUG, respectively, the problems MaxIMF and MinMC defined in these graphs. For both problems, an instance is then given by a triple $(G, \mathcal{N}, \mathrm{c})$ where $G = (V, E)$ is an undirected rectilinear grid with vertex set V and edge set E, whose edges are valued by a unique integer c, and $\mathcal{N}$ is a list of K nets. Moreover, as in [6; 18], we shall assume in Sections 3, 4, 5 and 6 of this paper, that the uniform grids we study are *augmented*, i.e., that each terminal is linked to the grid by a unique edge valued by c, unless we explicitly mention it. This can be assumed without loss of generality, since a uniform grid that does not satisfy this property can easily be transformed into an equivalent one that does. For each net (s_k, t_k), let s_k lie on the vertex v_k, which is adjacent to $deg(v_k) \in \{2, 3, 4\}$ vertices. Let v'_k and $deg(v'_k)$ be defined symmetrically for t_k. Then, replace s_k and t_k by $\min(deg(v_k), deg(v'_k))$ terminals linked by a unique edge valued by c to v_k and v'_k respectively, and replace (s_k, t_k) by $\min(deg(v_k), deg(v'_k))$ new nets defined on these $2 \cdot \min(deg(v_k), deg(v'_k))$ new terminals.

1.2 Related work

MaxIMF and MinMC have been studied in unrestricted graphs and in several types of planar graphs where they mostly remain $\mathcal{NP}$-hard [11; 5], and Călinescu et al. show that MinMC remains $\mathcal{NP}$-hard even in bounded-degree planar graphs [2]. More references and results concerning these problems can be found in [4].

Now, we turn to the problem MaxEDP. In [3], Chan and Chin give algorithms to find the maximum number of disjoint paths in grids when any vertex from a given source set can be paired with any vertex from a given sink set. Kleinberg and Tardos give in [12] a constant-factor approximation algorithm for MaxEDP in graphs they call *densely embedded and nearly eulerian*, and which generalize the rectilinear grids. Further results concerning MaxEDP can be found in [13].

Eventually, let us look at the decision problem EDP: this problem has been widely studied in grid graphs because of its interest in the design of VLSI circuits. An extensive survey on EDP can be found in [9], but we only detail some results here. Formann et al. study a special case in which short paths are required and give a polynomial algorithm to solve it [6], but the general problem is known to be $\mathcal{NP}$-complete in grids [14]. Moreover, Marx shows that it remains $\mathcal{NP}$-complete even in eulerian grids [15].

Nevertheless, Okamura and Seymour provide a good characterization for this problem when the graph is planar and eulerian, and the terminals all lie on the outer face [16]. In [7; 9], Frank gives necessary and sufficient conditions for the existence of K edge disjoint paths in grids where the terminals are distinct and lie on the uppermost and lowermost lines. We detail these conditions in Section 3 and use them as a starting point for solving the associated optimization problem.

1.3 Two-sided grids

In Sections 3, 4, 5 and 6, we focus on optimization problems associated with EDP, the decision problem studied by Frank in [7; 9]. He assumes that the grids are *two-sided*, i.e., that all the terminals lie on the uppermost and lowermost lines and are distinct. As mentioned in Section 1.1, EDP consists in deciding whether it is possible to route all the nets using edge disjoint paths. The corresponding optimization problem consists in *maximizing the number of nets linked by edge disjoint paths.* Hence, for each net (s_k, t_k), at most one path having s_k and t_k as endpoints is allowed. So, this maximization problem is equivalent to the problem MaxEDP defined on a grid where each source s_k

(resp. sink t_k) is linked to the rest of the grid by a unique edge e_k (resp. e'_k). The assumptions made by Frank are then equivalent to saying that each terminal is linked to a vertex of the uppermost or lowermost line and that at most one terminal can be linked to a given vertex. We also refer to this case as the two-sided one. In Section 5, we generalize our results concerning MaxEDP, and so we assume that all the edges, including e_k and e'_k for $k \in \{1, \ldots, K\}$, are valued by c ≥ 2.

1.4 Results and organization of the paper

First, we prove that MinMCUG is $\mathcal{NP}$-hard when several terminals can be on the same vertex, even if all the terminals lie on the uppermost and lowermost lines. Moreover, the result extends to augmented grids.

The main contribution of the rest of the paper is to use Frank's results to solve MaxIMFUG and MinMCUG in the two-sided case. The basis of our approach is the fact that solving MaxEDP in the two-sided case is equivalent to removing the minimum number of nets in order to fulfill Frank's conditions.

We show how to find efficiently an optimal solution for MaxEDP by selecting the nets to be removed via linear programming. We also prove that MinMCUG is polynomial-time solvable in the two-sided case, since a feasible solution whose value is proved to be equal to the maximum fractional multiflow can be obtained by solving a continuous linear program. Then, we use the results for MaxEDP and MinMCUG to solve MaxIMFUG in polynomial time in the two-sided case: we study different cases and settle each one of them by showing how the theorem of Okamura and Seymour [16] can be applied. As a by-product, the gap between the optimal values of MaxIMFUG and MinMCUG is shown to be at most one. Eventually, for the two-sided case, we describe two specific combinatorial algorithms solving MinMCUG and MaxIMFUG in polynomial-time, thus enabling us to avoid solving linear programs.

The paper is organized as follows. In Section 2, we give the $\mathcal{NP}$-hardness proof for MinMCUG in general grids. In Sections 3 and 4, we solve MaxEDP and MinMCUG, respectively, by using linear programming. In Section 5, we solve MaxIMFUG. Finally, in Section 6, we give the two combinatorial algorithms solving MaxIMFUG and MinMCUG.

In the following, given a problem or a linear program P, we abuse notation and write "*Opt*(P)" for both "*the optimal value of a given instance of P*" and "*the optimal values of all the instances of P*". It will be clear from the context which one we mean, and, in the first case, which instance is considered.

Moreover, for a better understanding of the general frame of the paper, the longest and most technical proofs are given in appendix.

2 Complexity of MinMCUG

In this section, we show that MinMCUG is $\mathcal{NP}$-hard if several terminals can be located on the same vertex. Moreover, this holds even if all the terminals lie on the uppermost and lowermost lines. From Section 1.1, this also holds in augmented grids where the terminals are linked to vertices of the uppermost and lowermost lines, if we allow that several terminals can be linked to the same vertex.

MinMC is shown to be $\mathcal{NP}$-hard in unweighted stars in [11]: by replacing the center of the star by a grid of size p (p being the number of leaves), we obtain a reduction showing that MinMC is $\mathcal{NP}$-hard in unweighted grids, if every terminal is linked to the grid by an edge and several terminals can lie on the same vertex. However, this grid is not an augmented grid as defined in Section 1.1, and adapting the proof to this case does not seem quite straightforward. We give a new proof, which has been inspired by the proof in [11] but also works for the *non augmented grids*, by reducing from the $\mathcal{NP}$-complete problem Vertex Cover [10]:

Input: A graph $H = (V, E)$, an integer $S \leq |V|$.

Question: Does H admit a vertex cover of size at most S?

Let $\mathcal{V}$ be an instance of Vertex Cover. We define from $\mathcal{V}$ an instance $\mathcal{M}$ of MCUG, the decision problem associated with MinMCUG. Let $V = \{u_1, \ldots, u_n\}$ and c $= 1$. The grid $(G, \mathcal{N})$ has $2n-1$ vertical lines and $n+2$ horizontal lines (it is *not* an augmented grid). We denote by g_j^i the vertex being on the i^{th} horizontal line and the j^{th} vertical line of the grid. We define the nets $(g_j^1,\, g_{j+1}^1)$, $j \in \{1, \ldots, 2n-2\}$, and $(g_{2j}^1,\, g_{2j}^{n+2})$, $j \in \{1, \ldots, n-1\}$. Note that the cheapest way to cut these nets is to remove the *edges* $(g_j^1,\, g_{j+1}^1)$, $j \in \{1, \ldots, 2n-2\}$, and $(g_{2j}^1,\, g_{2j}^2)$, $j \in \{1, \ldots, n-1\}$; this leaves a grid where each terminal is linked to the rest of the grid by a single edge, with several terminals lying on the same vertex. Moreover, for each edge $(u_i,\, u_j) \in E$, we define the net $(g_{2i-1}^1,\, g_{2j-1}^1)$.

Lemma 1 *Given $S \leq n$, there exists a vertex cover $\hat{\mathrm{C}}$ of size at most S in $\mathcal{V}$ if and only if $\mathcal{M}$ admits a multicut $\bar{\mathrm{C}}$ of value at most $3(n-1) + S$.*

Proof. First, we show the part "only if" (i.e., necessity). Assume we are given a vertex cover $\hat{\mathrm{C}}$ of size $|\hat{\mathrm{C}}| \leq S$. We select in the cut $\bar{\mathrm{C}}$ all the edges $(g_j^1,\, g_{j+1}^1)$, $j \in \{1, \ldots, 2n-2\}$, and $(g_{2j}^1,\, g_{2j}^2)$, $j \in \{1, \ldots, n-1\}$. Moreover, for each vertex u_j in $\hat{\mathrm{C}}$, we select in $\bar{\mathrm{C}}$ the edge $(g_{2j-1}^1,\, g_{2j-1}^2)$. We obtain a set of edges of size $3(n-1) + |\hat{\mathrm{C}}| \leq 3(n-1) + S$.

To see that $\bar{\mathrm{C}}$ is a multicut, first note that all the nets $(g_j^1,\, g_{j+1}^1)$, $j \in \{1, \ldots, 2n-2\}$, and $(g_{2j}^1,\, g_{2j}^{n+2})$, $j \in \{1, \ldots, n-1\}$, are disconnected. It only remains to show that all the nets $(g_{2i-1}^1,\, g_{2j-1}^1)$, (i, j) such that $(u_i, u_j) \in E$, are cut. Assume there exists a non cut one, say $(g_{2a-1}^1,\, g_{2b-1}^1)$. Then, neither $(g_{2a-1}^1,\, g_{2a-1}^2)$ nor $(g_{2b-1}^1,\, g_{2b-1}^2)$ is in the cut. Thus, by the construction of $\bar{\mathrm{C}}$, neither u_a nor u_b is in $\hat{\mathrm{C}}$, although $(u_a,\, u_b) \in E$ since the net $(g_{2a-1}^1,\, g_{2b-1}^1)$ exists: a contradiction. Necessity follows.

Now, we show the part "if" (i.e., sufficiency). Assume we are given a multicut $\bar{\mathrm{C}}$ of size $3(n-1) + S'$, with $S' \leq S \leq n$ and $S' \in \mathbb{Z}$. Every edge $(g_j^1,\, g_{j+1}^1)$, $j \in \{1, \ldots, 2n-2\}$, is in $\bar{\mathrm{C}}$ since its two endpoints define a net. Moreover, since all the nets $(g_{2j}^1,\, g_{2j}^{n+2})$, $j \in \{1, \ldots, n-1\}$, are cut, $\bar{\mathrm{C}}$ contains a vertical edge of the $2j^{th}$ vertical line, for $j \in \{1, \ldots, n-1\}$. So, there exists a subset of $\bar{\mathrm{C}}$ containing $3(n-1)$ such edges: thus, in fact, $|\bar{\mathrm{C}}| \geq 3(n-1)$ and $S' \geq 0$. Let $\bar{\mathrm{C}}_{\mathrm{S}'}$ be the set of edges belonging to $\bar{\mathrm{C}}$ but not included in the subset described above: we have $|\bar{\mathrm{C}}_{\mathrm{S}'}| = S'$.

Let $(\bar{G}, \mathcal{N})$ be the graph obtained from $(G, \mathcal{N})$ by removing all the edges in $\bar{\mathrm{C}}$. We can partition the vertices of the type g_{2j-1}^1 into two sets: the first one, F, is the set of vertices of this type that can be linked by a path in $(\bar{G}, \mathcal{N})$ to g_j^{n+2} for some j (i.e., to a vertex of the $n+2^{nd}$ horizontal line of $(G, \mathcal{N})$); the second, $\bar{F}$, contains all the other vertices of this type.

Now, we show that all the vertices in F are in the same connected component of $(\bar{G}, \mathcal{N})$, i.e., that given $g_{2a-1}^1 \in F$ and $g_{2b-1}^1 \in F$, there exists a path from g_{2a-1}^1 to g_{2b-1}^1 in $(\bar{G}, \mathcal{N})$. Let p_a (resp. p_b) be a path in $(\bar{G}, \mathcal{N})$ from g_{2a-1}^1 (resp. g_{2b-1}^1) to a vertex of the $n+2^{nd}$ horizontal line of $(G, \mathcal{N})$. If p_a and p_b intersect at some vertex or if they can be linked by a path containing only horizontal edges, we are done. Otherwise, $\bar{\mathrm{C}}$ contains an edge from the j^{th} horizontal line of $(G, \mathcal{N})$, for $j \in \{2, \ldots, n+2\}$. More precisely, these $n+1$ horizontal edges belong to $\bar{\mathrm{C}}_{\mathrm{S}'}$, and thus $|\bar{\mathrm{C}}_{\mathrm{S}'}| \geq n+1$: this contradicts $S' \leq S \leq n$.

As a consequence, it does not exist any net $(g_{2a-1}^1, g_{2b-1}^1) \in \mathcal{N}$ with $g_{2a-1}^1 \in F$ and $g_{2b-1}^1 \in F$, since otherwise $\bar{\mathrm{C}}$ is not a multicut: hence, each net has at least one terminal in $\bar{F}$.

Moreover, for every j such that $g_{2j-1}^1 \in \bar{F}$ (i.e., g_{2j-1}^1 is a terminal vertex separated from all the vertices of the $n+2^{nd}$ horizontal line of $(G, \mathcal{N})$), there obviously exists a vertical edge of the $2j-1^{st}$ vertical line that is in $\bar{\mathrm{C}}$ (and, more precisely, in $\bar{\mathrm{C}}_{S'}$). This implies that $|\bar{F}| \leq |\bar{\mathrm{C}}_{\mathrm{S}'}|$ $(= S')$.

So, we can construct a vertex cover $\hat{\mathrm{C}}$ by selecting every vertex $u_j \in V$ such that $g^1_{2j-1} \in \bar{F}$. $\hat{\mathrm{C}}$ is actually a vertex cover, since, as we showed previously, each net (i.e., each edge in V) has at least one endpoint in $\bar{F}$. $|\hat{\mathrm{C}}| = |\bar{F}| \leq S' \leq S$, so Lemma 1 follows. $\square$

Since obviously MCUG is in $\mathcal{NP}$ and the above reduction is made in polynomial time, Lemma 1 implies:

Theorem 2 MCUG *is* $\mathcal{NP}$-complete.

Corollary 3 MinMCUG *is* $\mathcal{NP}$-hard.

On the one hand, Corollary 3 shows that MinMCUG is $\mathcal{NP}$-hard in augmented grids where the terminals are linked to vertices of the uppermost and lowermost lines, if several terminals can be linked to the same vertex (in fact, even if only 5 terminals can be linked to the same vertex, since Vertex Cover remains $\mathcal{NP}$-hard in graphs where no vertex has more than 3 neighbors). On the other hand, we show in Sections 3, 4 and 5 that both MinMCUG and MaxIMFUG are polynomial-time solvable in two-sided augmented grids (i.e., if at most one terminal can be linked to each vertex).

3 Solving MaxEDP

In the following of the paper, we consider a two-sided uniform grid $(G, \mathcal{N}, \mathrm{c})$ ($(G, \mathcal{N})$ for short). We begin this section with some definitions. To simplify the notations, we assume that we add two "border" lines to the grid (one on the top, the other on the bottom), that contain terminals (as shown in Figure 2 in Section 4.2). We call these two lines the *uppermost* and *lowermost* lines respectively. We denote by m the number of horizontal lines (or simply *lines*), including neither the uppermost nor the lowermost lines, and by n the number of vertical lines (or *columns*) of the grid (recall that there are at most two terminals for each column, and so $n \geq K$). A *full grid* is a grid in which all the vertices of the uppermost and lowermost lines are terminal vertices: in this case, $K = n$ (examples are given in Figures 1 and 2). A vertex which is not a terminal is called *free*. Given a terminal z, we denote respectively by $lin(z)$ and $col(z)$ the border line of z (i.e., uppermost or lowermost) and the column of the vertex linked to z (or simply *column of* z), the first and the n^{th} columns being respectively the leftmost and rightmost ones. A net (s_k, t_k) is called *straight* if $col(s_k) = col(t_k)$. Given a non straight net (s_k, t_k), s_k (resp. t_k) is the *left terminal* if $col(s_k) < col(t_k)$ (resp. $col(s_k) > col(t_k)$), the *right* *terminal* otherwise. We say that a net l is *strictly R-longer* than a net l' if and only if the right terminal of l is on the right of the right terminal of l' (we will say only *R-longer* if we allow the right terminals to lie on the same column). We define a (*strictly*) *L-longer* net in a similar way, by replacing *right* by *left*: for instance, in Figure 1, the net (s_{10}, t_{10}) is R-longer and strictly L-longer than the net (s_9, t_9). A vertical (resp. horizontal) *strip* is the region (and the edges) between two consecutive vertical (resp. horizontal) lines: v_j (resp. h_j) will denote the j^{th} vertical (resp. horizontal) strip, the first being the leftmost (resp. uppermost) one. Note that v_j is between the j^{th} and the $j+1^{th}$ columns. The *density* [6] (or *congestion* [7]) d_j of a vertical strip v_j is the number of nets "crossing" it:

$$d_j = |\{(s_k, t_k) \text{ s. t. } col(s_k) \leq j < col(t_k) \text{ or } col(t_k) \leq j < col(s_k)\}|$$

A vertical strip v_j is *saturated* if $d_j = m$. The *density of the grid* is $d = \max\limits_{j \in \{1,\dots,n-1\}} \{d_j\}$ (see Figure 1) and we define $d_0 = 0$. Let n_j^L (resp. n_j^R) be the number of nets whose left (resp. right) terminal is on the j^{th} column. Then, we have:

$$\forall j \in \{0, \dots, n-2\},\ d_{j+1} = d_j + n_{j+1}^L - n_{j+1}^R \tag{1}$$

Furthermore

$$\forall j \in \{1, \dots, n-1\},\ n_j^L + n_j^R \leq 2 \tag{2}$$

Let $n_j^S \in \{0, 1\}$ be the number of straight nets on the j^{th} column. *A full grid has the property that d_j is even for all j*, since for each $j \in \{1, \dots, n-1\}$, (2) becomes $n_j^L + n_j^R + 2n_j^S = 2$ and thus (1) implies:

$$\forall j \in \{0, \dots, n-2\},\ d_{j+1} \in \{d_j - 2, d_j, d_j + 2\} \tag{3}$$

Recall that, in two-sided grids, MaxEDP consists in linking by edge disjoint paths as many pairs $(s_k,\, t_k)$ as possible.

If all the nets are straight, then they all can be routed vertically. The corresponding multicut is trivially obtained by cutting each net *on its source*, i.e., by selecting every e_k, $k \in \{1, \dots, K\}$, in the cut: in that case, we have $Opt(\text{MaxEDP}) = Opt(\text{MinMCUG}) = K$ (in the case where $c \geq 2$, we have $Opt(\text{MaxIMFUG}) = Opt(\text{MinMCUG}) = Kc$).

Otherwise, Frank proves the following [7; 9]:

Theorem 4 (Frank) *Let $(G, \mathcal{N})$ be a two-sided rectilinear grid with m lines and density d. Assume there is at least one non straight net. Then, all the nets* *can be linked by edge disjoint paths if and only if $(G,\mathcal{N})$ satisfies:*

either	$m > d$ **and**		*there is a free vertex on a border line*	
or	$m \geq d$			(4)
	and	**either**	*there exists a one-sided net*	(5)
		or	*there exists an extremal vertex*	(6)
		or	*there are two non separated vertices*	(7)

where a *one-sided net* is a net whose terminals are both on the same border line, an *extremal vertex* is a free vertex of the uppermost or lowermost line which is either on the left of the leftmost saturated vertical strip or on the right of the rightmost saturated vertical strip, and two *non separated vertices* are two free vertices both located either on the uppermost or on the lowermost line, and which are not separated by a saturated vertical strip.

Proof (Sketch) In fact, Frank proves in [9] that, in the two-sided case, a sufficient condition is that the grid satisfies (4) and (5). (4) is necessary, since if $m < d$ there are m edge disjoint paths (one for each horizontal edge) that can cross the vertical strip having the greatest density, while d paths need to cross it. Moreover, it is shown in [7] that, in *bipartite* grids (i.e., in two-sided grids where (5) does not hold), the second part of Theorem 4 becomes *if and only if the grid satisfies* (4) *and either* (6) *or* (7). □

In the following of the paper, we assume that there is at least one non straight net. Note that either (6) holds or it can be fulfilled simply by removing a single net (a net whose source or sink is linked to a corner of the grid, for instance). Using Theorem 4 we get:

Proposition 5 *If $(G,\mathcal{N})$ satisfies $m \geq d$ then*

- *Opt*(MaxEDP) $= K$ *if $m > d$ and there exists a non terminal vertex on the uppermost or lowermost line;*
- *Opt*(MaxEDP) $= K$ *if* (5), (6) *or* (7) *is satisfied;*
- *Opt*(MaxEDP) $= K - 1$ *otherwise.*

Proposition 5 settles the case where (4) holds. Also note that (5), (6) and (7) can be checked in $O(n)$.

In the following of Section 3, we assume that $(G,\mathcal{N})$ satisfies $m < d$. Our approach is to remove enough nets in order to obtain a final grid that satisfies (4) (we do not consider other necessary conditions for the moment). An equivalent formulation of the problem is to select from $\mathcal{N}$ as many nets as possible without exceeding a density equal to m (see constraints (8) below). This problem can be modelled as follows:

$$
\textbf{\textit{(INP)}}\left|\begin{array}{lll}
\max & \sum_{k=1}^{K} x_k & \\
\text{s. t.} & \sum_{k \text{ s. t. } l_k \text{ crosses } v_j} x_k \leq m \; \forall j \in \{1,\ldots,n-1\} & (8) \\
& x_k \in \{0,1\} \qquad \forall k \in \{1,\ldots,K\} & (9)
\end{array}\right.
$$

where x_k is equal to 1 iff the net $l_k = (s_k, t_k)$ is selected. (CNP), the continuous relaxation of (INP), is obtained by replacing constraints (9) by

$$x_k \leq 1, \; \forall k \in \{1,\ldots,K\} \tag{10}$$

and $x_k \geq 0$, $\forall k \in \{1,\ldots,K\}$.

Lemma 6 M*, the matrix defined by the left part of constraints* (8)*, is totally unimodular.*

Proof. The vertical strips crossed by each net being consecutive, there is a single sequence of consecutive 1's on each column of M. Then, M is an interval matrix [17, p. 279]: hence, M is totally unimodular. □

As a consequence, we know that, m being integral, any basic solution for (CNP) is integer: thus, (INP) is polynomial-time solvable. In fact, we shall see in Section 6 that (INP) can be solved as a CallControl problem on a chain by an efficient combinatorial algorithm [1].

Given an arbitrary optimal solution x^* for (INP), let $\mathcal{N}^-$ be the set of nets selected in x^*, i.e., the nets l_k such that $x^*_k = 1$. We call a net in $\mathcal{N}$ *removed* if it does not belong to $\mathcal{N}^-$. $(G, \mathcal{N}^-)$ has a density equal to m, since in any optimal solution for (INP) at least one of the constraints (8) is saturated. Let K^* be the number of nets in $(G, \mathcal{N}^-)$: we have $K^* = Opt(INP)$. $(G, \mathcal{N}^-)$ satisfies (4), and we still have that either (6) holds or it is fulfilled by removing a single net, so

$$Opt(\text{MaxEDP}) \in \{K^* - 1, K^*\} \tag{11}$$

But then, can the K^* nets selected by (INP) be linked by edge disjoint paths? Equivalently, does (INP) always admit a solution that satisfies (5), (6) or (7)? Figure 1 shows an example where $K^* = 8$, but $Opt(\text{MaxEDP}) = K^* - 1 = 7$:

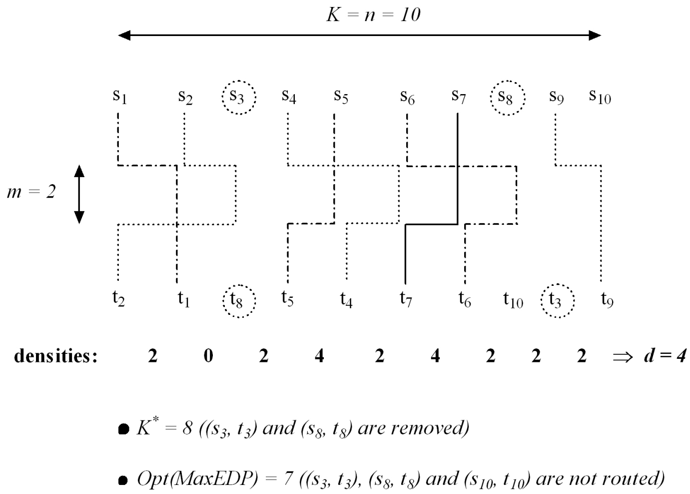


Figure 1. An instance with only $K^* - 1$ edge disjoint paths (in dashed lines).

the only feasible solution with 8 selected nets is obtained by removing (s_3, t_3) and (s_8, t_8), but it satisfies none of the three conditions (5), (6) and (7).

In general, it may exist several optimal solutions for (INP), and we have to determine whether one of them satisfies (5), (6) or (7). Let us first show that the situation is much simpler when m is odd.

Proposition 7 *If m is odd, then any grid satisfying $m = d$ satisfies* (6).

Proof. Assume (6) is not satisfied. Then, any vertex of the uppermost and lowermost lines located on the left (resp. the right) of the leftmost v_L (resp. the rightmost v_R) saturated vertical strip is a terminal. Since in a full grid all the densities are even (see (3)), the same is true for the two subgraphs respectively on the left of v_L and on the right of v_R. Hence, d_L and d_R, the respective densities of v_L and v_R, are even. But, since v_L and v_R are saturated, $d_L = d_R = m$. m being odd, we have a contradiction. □

Since $(G, \mathcal{N}^-)$ satisfies $m = d$, Proposition 7 immediately implies:

Corollary 8 *We have Opt(MaxEDP) $= K^*$ whenever m is odd and $(G, \mathcal{N})$ satisfies $m < d$.*

We still have to deal with the general case. We show the following:

Theorem 9 *When the grid $(G, \mathcal{N})$ satisfies $m < d$, an optimal solution for* MaxEDP *can be found by solving* $O(n^2)$ *continuous linear programs.*

Proof. In this proof, we do not make any distinction between an optimal solution x^* for (INP) and the grid obtained by removing all the nets not selected in this solution (i.e., all the nets l_k such that $x_k^* = 0$). If m is odd, then from Proposition 7 solving a single linear program suffices.

Otherwise, we first look for a solution satisfying (6). Let v_L (resp. v_R) be the leftmost (resp. rightmost) saturated vertical strip in $(G, \mathcal{N})$, and let l_{L^*} (resp. l_{R^*}) be the R-longest (resp. L-longest) net crossing v_L (resp. v_R). Then, there exists an optimal solution for (INP) that satisfies (6) if and only if, l_{L^*} *or* l_{R^*} *being removed* (i.e., $x_{L^*} = 0$ *or* $x_{R^*} = 0$), (INP) still has an optimal value of K^*. Indeed, sufficiency is easy, and necessity comes from the fact that, given a solution that satisfies (6), one can replace a removed net whose left terminal is on the left of v_L (resp. whose right terminal is on the right of v_R) by l_{L^*} (resp. by l_{R^*}), and obtain a feasible solution of the same value. So, we only need to solve two linear programs to know whether there exists an optimal solution for (INP) satisfying (6).

If such a solution does not exist, we then look for a solution satisfying (7). Given two vertices u_1 and u_2 both located on the same border line (i.e., $lin(u_1) = lin(u_2)$), if u_1 (resp. u_2) is a terminal, we denote by l_{u_1} (resp. l_{u_2}) the net it belongs to; otherwise we say that l_{u_1} (resp. l_{u_2}) is *undefined*. Assume without loss of generality that $col(u_1) < col(u_2)$. Let $(8)_{u_1,u_2}$ be the set of constraints (8) where m is replaced by $m-1$ for $j \in \{col(u_1), \ldots, col(u_2)-1\}$: i.e., we have $\sum_{k \text{ s. t. } l_k \text{ crosses } v_j} x_k \leq m-1, \forall j \in \{col(u_1), \ldots, col(u_2)-1\}$, and $\sum_{k \text{ s. t. } l_k \text{ crosses } v_j} x_k \leq m, \forall j \in \{1, \ldots, col(u_1)-1\} \cup \{col(u_2), \ldots, n-1\}$. Let $(INP(u_1, u_2))$ be the linear program obtained from (INP) by replacing (8) by $(8)_{u_1,u_2}$. Constraints $(8)_{u_1,u_2}$ guarantee that there is no saturated vertical strip between u_1 and u_2. Moreover, for all (u_1, u_2), the matrix associated with the left part of constraints $(8)_{u_1,u_2}$ is still M. Obviously, there exists an optimal solution for (INP) satisfying (7) if, l_{u_1} *and* l_{u_2} *being removed (or undefined)*, $Opt(INP(u_1, u_2)) = Opt(INP) = K^*$. The idea is then to solve $(INP(u_1, u_2))$ for each pair (u_1, u_2) such that u_1 and u_2 both lie on the same border line. Note that we have to do it for both the uppermost and the lowermost lines. So we need to solve at most $O(n^2)$ linear programs to decide whether (INP) admits an optimal solution that satisfies (7) or not.

Eventually, using the above ideas, we can check if there is a solution satisfying (5) by solving $O(n)$ linear programs (indeed, there are only $O(n)$ pairs (u_1, u_2) to be considered). Therefore, we need to solve $O(\max(2, n, n^2))$ linear programs to decide whether Opt(MaxEDP) is equal to K^* or not. □

We shall see in Section 6 that $n = O(K)$, and thus, solving $O(K^2)$ linear programs suffices. Eventually, once we have computed Opt(MaxEDP) and decided which nets have to be routed, we can use the algorithms given in [7] or [18] to actually route the edge disjoint paths.

It can be noticed that (CNP) is integral even for the weighted case, i.e., even if we assign a weight to each net and want to maximize the total weight of the routed nets (let us denote by $(WINP)$ the corresponding generalization of (INP)). It is then natural to try to generalize the results obtained for MaxEDP to the corresponding Weighted MaxEDP problem (WMEDP). In fact, whenever m is odd and $m < d$, we have $Opt(\text{WMEDP}) = Opt(WINP)$ (this generalizes Corollary 8), and, in all other cases (including $m \geq d$), WMEDP can be solved by using the algorithm given in Theorem 9. However, note that, when looking for a solution satisfying (6), we must try *all the nets crossing* v_L *or* v_R (and not only l_{L^*} and l_{R^*}), but this does not increase the running time in the worst case.

4 Solving MinMCUG

4.1 Preliminary results

In this section, we introduce several notions and results that will be useful in Sections 4.2 and 5. First, we recall the definition of the well-known *demand multiflow problem.* In this problem, we are given a supply graph $G = (V, E)$ and a demand graph $H = (T, \mathcal{N})$, whose vertices $T \subseteq V$ are the terminals and whose edges are the nets. Each edge e in E is valued by a capacity $U(e)$ and each edge (or net) l in $\mathcal{N}$ is valued by a demand $D(l)$. The problem is to decide whether it is possible to route all the demands. For notational convenience, we shall use $\mathcal{N}$ instead of $H = (T, \mathcal{N})$ (T being implicit), and the expression *demand set* instead of *demand graph.* Given a subset $X \subseteq V$, we define the *cut* $\delta_G(X)$ (resp. $\delta_{\mathcal{N}}(X)$) as the set of edges (u_i, u_j) in E (resp. in $\mathcal{N}$) such that $u_i \in X$ and $u_j \in V \setminus X$. For $L \subseteq E$ and $L' \subseteq \mathcal{N}$, we define $U(L) = \sum_{l \in L} U(l)$ and $D(L') = \sum_{l \in L'} D(l)$. Then, a necessary condition for the solvability is the well-known *cut condition*:

$$\forall X \subseteq V,\, U(\delta_G(X)) \geq D(\delta_{\mathcal{N}}(X))$$

A vertex v is *odd* if $U(\delta_G(\{v\})) + D(\delta_{\mathcal{N}}(\{v\}))$ is odd. Furthermore, we say that $(G, \mathcal{N})$ satisfies the *eulerian condition* iff it has no odd vertex, i.e.,

$$\forall v \in V,\, U(\delta_G(\{v\})) + D(\delta_{\mathcal{N}}(\{v\})) \text{ is even} \tag{12}$$

Okamura and Seymour prove the following [16]:

Theorem 10 (Okamura and Seymour) *Let $G = (V, E)$ be a planar supply graph with demand set $\mathcal{N}$. Let U and D be the capacity and demand functions respectively. Assume that the terminal vertices are on the boundary of the outer face of G. Then there exists a feasible fractional multiflow if and only if the cut condition holds. If moreover U and D are integer-valued and $(G, \mathcal{N})$ satisfies the eulerian condition, then there is an integer multiflow.*

In the following, we assume that an instance $\mathcal{I}$ of the demand multiflow problem is given by $\mathcal{I} = (G, \mathcal{N}, U, D)$ with G, $\mathcal{N}$, U and D defined as above.

Let $G = (V, E)$ be a two-sided grid and let $\mathcal{I} = (G, \mathcal{N}, U, D)$ be an instance of the demand multiflow problem. Recall that, for each net $l_k = (s_k, t_k)$, e_k and e'_k are the edges adjacent to s_k and t_k respectively. Obviously, a necessary condition for the solvability of $\mathcal{I}$ is that $\min(U(e_k), U(e'_k)) \geq D((s_k, t_k))$ for each $k \in \{1, \ldots, K\}$. So, assume it is satisfied: then, solving the demand multiflow problem on $(G, \mathcal{N})$ is equivalent to solving it on the grid obtained from $(G, \mathcal{N})$ by contracting every e_k and every e'_k into a single vertex (and whose set of edges is thus $E' = E \setminus \bigcup_{k \in \{1,\ldots,K\}} \{e_k, e'_k\}$). *For this problem*, we can then assume that $(G, \mathcal{N})$ is not an augmented grid. We will use Theorem 10 in Sections 4.2 and 5. First, we need the following result concerning the cut condition in two-sided non augmented grids.

Lemma 11 *Let $(G, \mathcal{N})$ be a two-sided non augmented grid where*
(a) *all the horizontal edges have the same capacity U_h,*
(b) *all the vertical edges have the same capacity $U_v \geq \max_{l \in \mathcal{N}} D(l)$, except the ones located on the leftmost and rightmost columns, whose capacities are at most U_v,*
(c) *each horizontal strip h_i satisfies $\sum_{e \text{ is an edge of } h_i} U(e) \geq \sum_{k=1}^{K} D(l_k)$,*
(d) *either $U_h = U_v$ or each vertical edge on the leftmost and rightmost columns is also valued by U_v.*

Then the cut condition is satisfied by any $X \subseteq V$ if and only if it is satisfied by any $X \subseteq V$ such that $\delta_G(X)$ is a vertical strip of the grid.

The proof of Lemma 11 is given in Appendix A. In the following, we shall see that the grids we need to consider always satisfy the assumptions of Lemma 11, and thus Theorem 10 will apply if and only if the cut condition holds for each vertical strip, i.e., for $U_h = 1$, if and only if $m \geq d$.

4.2 Solving MinMCUG by linear programming

First recall that, since we assume that all the edges are valued by the same integer, solving MinMCUG is equivalent to finding a *minimum set of edges* whose removal separates s_k from t_k for each net (so, in the following of Section

4.2, we will assume that $c = 1$, unless a different value is explicitly mentioned). We solve MinMCUG in the two-sided case by using an approach based on a duality relationship. We start by proposing a linear programming formulation, and we show how this provides a feasible solution for MinMCUG. Then, we prove that there exists a *fractional* multicommodity flow having the same value as this particular solution.

Let y_j, $j \in \{1, \ldots, n-1\}$, be the dual variables associated with constraints (8) and let w_k, $k \in \{1, \ldots, K\}$, be the ones associated with constraints (10). The dual linear program of (CNP) is given by:

$$
\boldsymbol{(CD)}\left|\begin{array}{lllr}
\min & \displaystyle\sum_{k=1}^{K} w_k + m \sum_{j=1}^{n-1} y_j & & \\
\text{s. t.} & w_k + \displaystyle\sum_{j \text{ s. t. } l_k \text{ crosses } v_j} y_j \geq 1 & \forall k \in \{1, \ldots, K\} & (13) \\
& y_j \geq 0 & \forall j \in \{1, \ldots, n-1\} & (14) \\
& w_k \geq 0 & \forall k \in \{1, \ldots, K\} & (15)
\end{array}\right.
$$

Lemma 12 (CD) *admits an optimal solution that defines a multicut.*

Proof. Since the objective function of (CD) is to be minimized and has only positive coefficients, then from constraints (13) any optimal solution satisfies $y_j \leq 1$ for all j, and $w_k \leq 1$ for all k. From Lemma 6, the constraint matrix of (CD) is totally unimodular. Thus, if we consider only basic solutions, we can replace the constraints (14) and (15) by $y_j \in \{0, 1\}$ for all j and $w_k \in \{0, 1\}$ for all k respectively, and get an integer program such that any of its solutions defines a particular multicut C whose value is $\sum_{k=1}^{K} w_k + m \sum_{j=1}^{n-1} y_j$, and whose edges are given by (see Figure 2):

- $w_k = 1 \Leftrightarrow e_k \in$ C;
- $y_j = 1 \Leftrightarrow v_j$ *is in* C, i.e., all the edges of v_j belong to C.

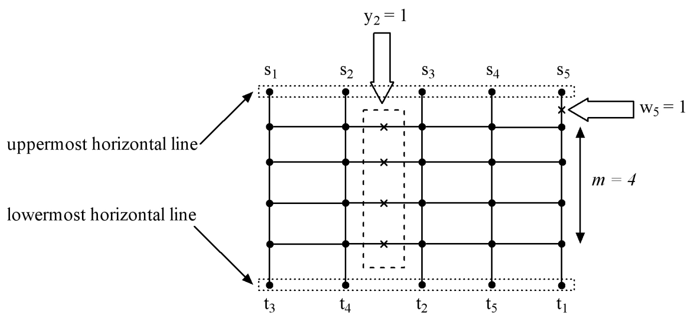


Figure 2. A cut given by (CD).

C is indeed a multicut since, from (13), for each net $l_k = (s_k, t_k)$, either $w_k = 1$ and so e_k is in C (thus, s_k is separated from the grid and so from t_k), or there exists a j such that l_k crosses v_j and $y_j = 1$ (so, there exists a vertical strip between s_k and t_k whose (horizontal) edges are in C). $\square$

If $(G, \mathcal{N})$ satisfies $m \geq d$, then for convenience we shall write K^* for K. Let (w^*, y^*) be an integer optimal solution for (CD). By the duality relationship between the linear programs (CNP) and (CD), we have $Opt(CNP) = Opt(CD) = K^* = \sum_{k=1}^{K} w_k^* + m \sum_{j=1}^{n-1} y_j^*$. Thus we can obtain, in polynomial time, a feasible solution for MinMCUG that contains K^* edges, i.e., which is optimal among all the multicuts associated with integer solutions of (CD). But there can exist other types of multicuts, and we have to prove that this solution is also optimal for MinMCUG, i.e., that there always exists an optimal multicut of this type: from Section 3, the optimal value of MaxEDP can be $K^* - 1$ (see Figure 1 in Section 3), so this leaves a gap of one unit between Opt(MaxEDP) and $Opt(CD)$.

Lemma 13 *In two-sided grids with unit capacities, the optimal value of the maximum fractional multiflow problem is* K^*.

Proof. We turn to the demand multiflow problem described in Section 4.1. We build an instance $\mathcal{I}$ of this problem to prove Lemma 13: the idea is to work on $(G, \mathcal{N}^-)$. $(G, \mathcal{N}^-)$ has K^* nets, and we have $U(e) = 1$ for every edge e. We define demands as $D(l) = 1$ for every net l in $\mathcal{N}^-$, and we consider $\mathcal{I} = (G, \mathcal{N}^-, U, D)$. $(G, \mathcal{N}^-)$ has a density equal to m, thus the cut condition holds for each vertical strip. Moreover, $\mathcal{I}$ satisfies (a), (b), (c) and (d), thus, from Lemma 11, $(G, \mathcal{N}^-)$ satisfies the cut condition, and, from the first part of Theorem 10, Lemma 13 follows. $\square$

If all the edges are valued by a unique integer c ≥ 2, then the optimal values of both (CD) and the maximum fractional multiflow problem are K^*c: we define $U(e) =$ c for every edge e and $D(l) =$ c for every net l in $(G, \mathcal{N}^-)$, and we apply Lemma 11. The value of any feasible multiflow being at most the value of any multicut [4], Lemma 13 implies that there is no integrality gap for the multicut problem, and hence:

Corollary 14 *In the two-sided case, an optimal solution for* MinMCUG *is obtained by solving* (CD).

In Section 6, we propose an efficient combinatorial algorithm that solves (CD) and thus, from Corollary 14, that computes an optimal multicut.

5 Solving MaxIMFUG

In this section, we solve MaxIMFUG in the two-sided case, *under the assumption that* $\mathrm{c} \geq 2$ (Section 3 deals with the case $\mathrm{c} = 1$). Recall that $K^* = Opt(CNP)$ if $m < d$, and $K^* = K$ otherwise. The main result is stated in the following theorem:

Theorem 15 *If* (5) *does not hold,* c *is odd,* $K = n$ *and* $d \leq m < \lceil \frac{d\mathrm{c}}{\mathrm{c}-1} \rceil$*, then* $Opt(\text{MaxIMFUG}) = K\mathrm{c} - 1$*; Otherwise* $Opt(\text{MaxIMFUG}) = K^*\mathrm{c}$.

The proof of Theorem 15 is given in Appendix B, where the different cases are settled by four lemmas (see Table B.1 in Appendix B). The details of the proof also show that solving MaxIMFUG when $\mathrm{c} \geq 2$ only requires finding $(G, \mathcal{N}^-)$, i.e., solving a single linear program. This is in contrast with MaxEDP, where Theorem 9 shows that, in the worst case, we need to solve $\mathrm{O}(K^2)$ linear programs.

Furthermore, it can be noticed that the optimum value of MaxIMFUG when $\mathrm{c} \geq 2$ is equal to $K^*\mathrm{c}$ whenever $m < d$, and whenever m is large enough (i.e., $m \geq \lceil \frac{d\mathrm{c}}{\mathrm{c}-1} \rceil$) as well; whereas the optimum value of MaxEDP is not always equal to K^* when $m < d$, and is never equal to K when $m \geq d$, (5) does not hold and $K = n$ (even if m is very large).

6 Algorithmic aspects

6.1 Solving (INP) and (CD)

In Sections 3 and 5, we show how (INP) can be used to solve MaxEDP and MaxIMFUG respectively. In Section 4, we show, by means of a duality relationship, that an integer optimal solution for (CD) is an optimal solution for MinMCUG.

In this section, we describe two combinatorial algorithms, CAN and CAC, solving (INP) and (CD) respectively. Furthermore, the proof given in Section 6.2 shows that CAN also provides an optimal solution for all $(INP(u_1, u_2))$ (as required in the proof of Theorem 9, in Section 3).

PROCEDURE CAN // *CAN solves (INP)*
Input: The grid $(G, \mathcal{N})$, with $\mathcal{N} = \{l_1, \ldots, l_K\}$
Output: $\mathcal{N}^- \subseteq \mathcal{N}$ such that $|\mathcal{N}^-| = K^*$ and the density of $(G, \mathcal{N}^-)$ is m

Sort the nets such that l_{k+1} is R-longer than l_k, for each $k \in \{1, \ldots, K-1\}$;
For each k **from** 1 **to** K **do**

Select l_k in $\mathcal{N}^-$ if it does not increase any density to more than m;

// If solving $(INP(u_1, u_2))$*, just replace* m *by* $m-1$ *for all* $v_{j, j \in \{col(u_1), \ldots, col(u_2)-1\}}$
EndFor

One can run CAC only after running CAN, since $\mathcal{N}^-$ must have been computed. To describe CAC, we need to introduce the *forbidden area* notion. We will see in Section 6.2 that a net selected in CAN must be cut only once. Therefore, at the moment a vertical strip v_j crossed by a selected net (say l_k) is added to the cut, one knows that no other vertical strip crossed by l_k will be added to the cut. In order to guarantee this, we define l_j^* as the L-longest selected net crossing v_j, and the forbidden area of v_j as the region between v_j and the left terminal of l_j^*. Any vertical strip on the left of v_j which is crossed by l_j^* is said to be in the forbidden area of v_j. At each step, the whole forbidden area is defined as the union of all the forbidden areas already defined. Hence, when v_j is the current vertical strip examined by the algorithm, updating the forbidden area means finding l_j^*. For instance, in Figure 2, v_2 is in the cut and its forbidden area is defined by $l_1 = (s_1, t_1)$ (or by (s_3, t_3)): then, no other vertical strip between v_2 and the leftmost vertical line (where s_1, the left terminal of l_1, lies) will be added to the cut. Thus, in this example, v_2 is the only vertical strip that is in the cut.

PROCEDURE CAC // CAC *solves* (CD)
Input: The grid $(G, \mathcal{N}^-)$
Output: A multicut C $\subseteq E$ such that $|\text{C}| = K^*$

forbidden area $:= \emptyset$;
C $:= \emptyset$; // *initially, the cut is empty*
1. **For each** vertical strip v_j **from** right **to** left **do**

If v_j is saturated in $(G, \mathcal{N}^-)$ **then**

If v_j is not in the forbidden area **then**

C $:=$ C $\cup$ {edges in v_j}; // v_j *is added to the cut*

Update the forbidden area;

EndIf

EndIf
EndFor
2. **For each** net l_k selected in CAN which is not already cut **do**

C := C $\cup \{e_k\}$; // l_k *is cut on its source*
EndFor

6.2 Correctness

In this section, we prove the optimality of CAN and CAC by using the complementary slackness conditions associated with the dual linear programs (CNP) and (CD). Recall that M is totally unimodular and let x^* and (w^*, y^*) denote integer optimal solutions for (CNP) and (CD) respectively. Then, the complementary slackness conditions are given by:

- From (8): $y_j^*(m - \sum_{k/l_k \text{ crosses } v_j} x_k^*) = 0$. It means:

$$v_j \text{ is in the cut } (y_j^* = 1) \text{ only if it is saturated } (\sum_{k/l_k \text{ crosses } v_j} x_k^* = m) \quad (16)$$

- From (10): $w_k^*(1 - x_k^*) = 0$. It means:

$$\text{a removed net } (x_k^* = 0) \text{ cannot be cut on its source } (w_k^* = 0) \quad (17)$$

- From (13): $x_k^*((w_k^* + \sum_{j/l_k \text{ crosses } v_j} y_j^*) - 1) = 0$. It means:

$$\text{a selected net } (x_k^* = 1) \text{ is cut only once } (w_k^* + \sum_{j/l_k \text{ crosses } v_j} y_j^* = 1) \quad (18)$$

First, note that the solution given by CAN is feasible. Moreover, the solutions given by CAN and CAC satisfy the complementary slackness conditions. Indeed, on the one hand we select in the cut only vertical strips which are saturated (16), on the other hand a net not selected in CAN is never cut on its source (17), and eventually every net that has been selected in CAN is cut once in CAC (at least in step 2), but never twice or more (i.e., we never have $w_k^* + \sum_{j/l_k \text{ crosses } v_j} y_j^* \geq 2$) because of the forbidden area notion (18). We still have to prove that the solution given by CAC defines a multicut, i.e., that every removed net is actually cut.

Lemma 16 *Every net removed in CAN is cut in CAC.*

Proof. Assume there exists a removed net which is not cut, l. Let v be the leftmost vertical strip crossed by l which is saturated in $(G, \mathcal{N}^-)$ (v exists, since otherwise l would have been selected in CAN). Let $\hat{v}$ be the leftmost vertical strip on the right of v being in the cut ($\hat{v}$ exists, since otherwise v is the rightmost saturated vertical strip in $(G, \mathcal{N}^-)$, and so v is in the cut and l is cut), and let $\hat{l}$ be the net defining the forbidden area of $\hat{v}$. v is in this forbidden area, since otherwise v is in the cut and so l is cut.

l has been examined before $\hat{l}$, since $\hat{l}$ is strictly R-longer than l (because $\hat{l}$ crosses $\hat{v}$ and l does not). Moreover, by the choice of v, all the saturated vertical strips crossed by l are crossed by $\hat{l}$. So l should have been selected instead of $\hat{l}$ in CAN, a contradiction. Lemma 16 follows. $\square$

6.3 Efficient implementation

In this section, we show that CAN and CAC run in $\mathrm{O}(K)$, and thus that both have asymptotically optimal running times. From Theorem 4, we do not change the solvability of the instance if we assume that there are no more than two consecutive columns without terminals in the grid. Since there are $2K$ terminals, we can assume without loss of generality that $n \leq 6K$. Moreover, since $(G, \mathcal{N}, \mathrm{c})$ is a grid, giving c, m, n and the K pairs ($col(s_k)$, $col(t_k)$) and ($lin(s_k)$, $lin(t_k)$) is sufficient to fully describe the input. So, we assume that the input of CAN is a table $\mathcal{T}$ of length $n = \Theta(K)$, where the j^{th} element contains, for each one of the two (or less) terminals being on the j^{th} column of the grid, the column of its mate. If, for instance, a set of pairs ($col(s_k)$, $col(t_k)$) is given, one can easily compute $\mathcal{T}$ in $\mathrm{O}(K)$, by going through this set of pairs once.

Running CAN can be done by solving an instance of the CallControl problem on a chain (see [1] for details): each vertical strip of the grid v_j becomes an edge f_j of the chain, each net becomes a *call*, and the capacity of each f_j is m if solving (INP), and $m-1$ (if $j \in \{col(u_1), \ldots, col(u_2)-1\}$) or m (otherwise) if solving $(INP(u_1, u_2))$ for some (u_1, u_2). It is shown in [1] that this problem can be solved in $\mathrm{O}(p+q)$, where p is the number of calls and q the number of edges. In our case, we have $p = K$ and $q = n-1 = \mathrm{O}(K)$, so CAN can be solved in $\mathrm{O}(K)$.

Now, we show that CAC also runs in $\mathrm{O}(K)$. The main difficulty for CAC is to efficiently update the forbidden area. Let the output of CAN be the table $\mathcal{T}' = \mathcal{T} \setminus \{\text{removed nets}\}$. One can compute in $\mathrm{O}(K)$ the densities of this new grid by using (1), and store them in a new table called $\mathcal{D}$. At each step, $\mathcal{D}$ will be used to know whether the current vertical strip is saturated or not. If this strip is added to the cut, $\mathcal{T}'$ will then be used to find the net that defines its forbidden area.

Let $\bar{v}_j$ and $\bar{v}_{j+1}$ be two consecutive vertical strips of the cut. The net defining the forbidden area of $\bar{v}_j$ cannot have its right endpoint on the right of $\bar{v}_{j+1}$, since otherwise this net crosses both $\bar{v}_j$ and $\bar{v}_{j+1}$, and so is strictly L-longer than the net defining the forbidden area of $\bar{v}_{j+1}$: a contradiction. So, when a vertical strip is added to the cut, finding its forbidden area only requires examining the nets whose right terminals are between this vertical strip and the previous one being in the cut. Thus, during the whole execution of CAC, each column in $\mathcal{T}'$ is examined only once. Since the same holds for each vertical strip, the running time of step 1 is $\mathrm{O}(\max(|\mathcal{T}'|, |\mathcal{D}|)) = \mathrm{O}(K)$. Moreover, at the moment a net is examined, one can know whether it will cross a vertical strip that belongs to the cut, and remove it from $\mathcal{T}'$ if it does. Step 2 consists then in cutting on its source each net that remains in $\mathcal{T}'$: it takes $\mathrm{O}(K)$ time. Thus, CAC runs in $\mathrm{O}(K)$.

7 Conclusion

We have solved MaxIMFUG and MinMCUG in the two-sided case by using several results concerning decision problems related to our optimization problems [7; 8; 9; 16]. On the algorithmic side, we have given two combinatorial algorithms to solve them, and the one solving MinMCUG runs in linear time, while the one solving MaxIMFUG runs in linear time whenever $\mathrm{c} \geq 2$ or $m \geq d$ or m is odd. Furthermore, we have proved that the gap between the optimal values of MaxIMFUG and MinMCUG is at most one, and we have shown how to determine the very special cases where these two values are not equal. For MaxEDP, we are always able to compute in linear time a solution whose value is at least the optimal value of MaxEDP minus one, even when our algorithm would take more time (i.e., $\mathrm{O}(n^3)$ time) to find an optimal solution. Nevertheless, we do not know whether the part $\mathrm{O}(n^2)$ in the expression of Theorem 9 could be improved or not. Moreover, we would like to point out that, throughout the paper, the running times of the routing algorithms are omitted on purpose. Thus, the complexity results given for MaxIMFUG and MaxEDP only provide the time needed to decide how many units of flow are routed for each net. If one wants to explicitly construct the routing, the algorithms given in [7; 8; 16; 18] can be used.

It would be quite interesting to extend our results to more general graphs, when a good characterization is known for the decision problem. For instance, Frank proves in [8] that, when all the terminals are on the boundary of the outer face, EDP is polynomial-time solvable in planar graphs more general than grids, i.e., inner eulerian planar graphs, including outerplanar graphs (a very special class of planar graphs, having all their vertices lying on the outer face). However, Garg et al. prove in [11] that MaxEDP is $\mathcal{NP}$-hard in outerplanar graphs (by showing that MaxIMF is $\mathcal{NP}$-hard in trees with capacities 1 and 2). Furthermore, they show that MinMC is $\mathcal{NP}$-hard in stars with unit capacities. Thus, it seems that, even in classes of graphs where EDP can be solved efficiently, for MaxEDP and MinMCUG, only very special cases are likely to be polynomial-time solvable.

References

[1] U. Adamy, C. Ambuehl, R. Sai Anand and T. Erlebach. Call Control in Rings. Proceedings ICALP, Lecture Notes in Computer Science 2380 (2002), pp. 788-799.

[2] G. Călinescu, C.G. Fernandes and B. Reed. Multicuts in unweighted graphs and digraphs with bounded degree and bounded tree-width. Journal of Algorithms 48 (2003), pp. 333-359. Academic Press.

[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. Academic Press.

[4] M.-C. Costa, L. Létocart and F. Roupin. Minimal multicut and maximal integer multiflow: A survey. EJOR 162 (2005), pp. 55-69.

[5] E. Dahlhaus, D.S. Johnson, C.H. Papadimitriou, P.D. Seymour and M. Yannakakis. The complexity of multiterminal cuts. SIAM J. Comput. 23 (1994), pp. 864-894.

[6] 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.

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

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

[9] A. Frank. Packing Paths, Circuits and Cuts - A Survey. In B. Korte, L. Lovász, H. J. Prömel and A. Schrijver. *Paths, Flows and VLSI-Layout.* Algorithms and combinatorics 9 (1990), pp. 47-100. Springer-Verlag. Berlin.

[10] M.R. Garey and D.S. Johnson. Computers and Intractability: a Guide to the Theory of NP-completeness. W.H. Freeman and Co., San Fransisco (1979).

[11] N. Garg, V.V. Vazirani and M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. Algorithmica 18 (1997), pp. 3-20.

[12] J. Kleinberg and É. Tardos. Disjoint paths in densely embedded graphs. Proceedings 36^{th} IEEE FOCS (1995), pp. 52-61.

[13] J. Kleinberg. Approximation algorithms for disjoint paths problems. PhD thesis, MIT, Cambridge, MA (1996).

[14] M.E. Kramer and J. van Leeuwen. The complexity of wire routing and finding the minimum area layouts for arbitrary VLSI circuits. Advances

in computing research 2: VLSI theory, F.P. Preparata (ed.), JAI Press, London (1984), pp. 129-146.

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

[16] H. Okamura and P.D. Seymour. Multicommodity flows in planar graphs. Journal of Combinatorial Theory, Series B 31 (1981), pp. 75-81.

[17] A. Schrijver. Theory of linear and integer programming. Interscience series in discrete mathematics and optimization (1986). Wiley.

[18] K. Weihe. Edge-disjoint routing in plane switch graphs in linear time. Proceedings 40^{th} IEEE FOCS, New York City (1999), pp. 330-340.

A Proof of Lemma 11

Lemma 11 *Let* $(G, \mathcal{N})$ *be a two-sided non augmented grid where*
(a) *all the horizontal edges have the same capacity* U_h*,*
(b) *all the vertical edges have the same capacity* $U_v \geq \max_{l \in \mathcal{N}} D(l)$*, except the ones located on the leftmost and rightmost columns, whose capacities are at most* U_v*,*
(c) *each horizontal strip* h_i *satisfies* $\sum_{e \text{ is an edge of } h_i} U(e) \geq \sum_{k=1}^{K} D(l_k)$*,*
(d) *either* $U_h = U_v$ *or each vertical edge on the leftmost and rightmost columns is also valued by* U_v*.*

Then the cut condition is satisfied by any $X \subseteq V$ *if and only if it is satisfied by any* $X \subseteq V$ *such that* $\delta_G(X)$ *is a vertical strip of the grid.*

Proof. The necessity of the cut condition on each vertical strip being obvious, we show the sufficiency. Let $X \subseteq V$ be such that $U(\delta_G(X)) < D(\delta_{\mathcal{N}}(X))$. We can assume w.l.o.g. that X is connected. Let us show that we can find a vertical strip that violates the cut condition.

Let $D^* = \max_{l \in \mathcal{N}} D(l)$ and let n be the number of columns of the grid. X is "bounded" by two columns, c_λ on its left and c_ρ on its right: it means that for every vertex u in X, u is between the λ^{th} and the ρ^{th} column inclusive (take λ as large as possible and ρ as small as possible). The proof is organized as follows: in the first part, we assume that $\lambda > 1$ and $\rho < n$; in the second part, we consider the case where $\lambda = 1$ or $\rho = n$.

First assume that $\lambda > 1$ and $\rho < n$. If X does not contain any vertex from the uppermost and lowermost lines, $|\delta_{\mathcal{N}}(X)| = 0$, a contradiction. If X does not contain any vertex from the uppermost (resp. lowermost) horizontal line, then, X being connected, $\delta_G(X)$ contains *at least* $\rho - \lambda + 1$ vertical edges (one from each column between c_λ and c_ρ), whereas $\delta_{\mathcal{N}}(X)$ contains *at most*

$\rho - \lambda + 1$ edges, since there are at most $\rho - \lambda + 1$ terminals between c_λ and c_ρ on the lowermost (resp. uppermost) line. Thus, from (b)

$$U(\delta_G(X)) \geq (\rho - \lambda + 1)U_v \geq (\rho - \lambda + 1)D^* \geq D(\delta_{\mathcal{N}}(X)) \tag{A.1}$$

a contradiction. Hence, X contains at least one vertex from the uppermost line and one vertex from the lowermost line: X "reaches" both lines.

Given a subset $Y \subseteq V$, let $\delta^v_G(Y)$ (resp. $\delta^h_G(Y)$) denote the set of vertical (resp. horizontal) edges in $\delta_G(Y)$. We have $\delta_G(Y) = \delta^v_G(Y) \cup \delta^h_G(Y)$ and $U(\delta_G(Y)) = U(\delta^v_G(Y)) + U(\delta^h_G(Y))$. We transform X into the subset $X' \subseteq V$ containing all the vertices between c_λ and c_ρ (i.e., X' is the smallest rectangle containing X, see case I in Figure A.1). Since X is connected and reaches both the uppermost and the lowermost lines, $|\delta^h_G(X)| \geq 2m$, and thus we have

$$U(\delta^h_G(X)) \geq 2mU_h = U(\delta^h_G(X')) \tag{A.2}$$

For notational convenience, we shall let $\delta_{\mathcal{N}}(X - X')$ denote the set of nets *being in* $\delta_{\mathcal{N}}(X)$ *and not in* $\delta_{\mathcal{N}}(X')$. Obviously, one has $D(\delta_{\mathcal{N}}(X - X')) \geq D(\delta_{\mathcal{N}}(X)) - D(\delta_{\mathcal{N}}(X'))$. For each vertical edge e that was removed from $\delta^v_G(X)$ when transforming X into X', at most one net has been removed from $\delta_{\mathcal{N}}(X)$ (a net having a terminal on the same column as e), thus

$$|\delta_{\mathcal{N}}(X - X')| \leq |\delta^v_G(X)| - |\delta^v_G(X')| = |\delta^v_G(X)| \tag{A.3}$$

since $|\delta^v_G(X')| = 0$. So, we have

$$\begin{aligned}
D(\delta_{\mathcal{N}}(X)) - D(\delta_{\mathcal{N}}(X')) &\leq D(\delta_{\mathcal{N}}(X - X')) \\
&\leq D^*|\delta_{\mathcal{N}}(X - X')| \\
&\underbrace{\leq}_{\textit{from } (b)} U_v|\delta_{\mathcal{N}}(X - X')| \\
&\underbrace{\leq}_{\textit{from } (A.3)} U_v|\delta^v_G(X)| = U(\delta^v_G(X))
\end{aligned} \tag{A.4}$$

Combining (A.2), (A.4), and $D(\delta_{\mathcal{N}}(X)) > U(\delta_G(X))$, we get

$$\begin{aligned}
D(\delta_{\mathcal{N}}(X')) &\geq D(\delta_{\mathcal{N}}(X)) - U(\delta^v_G(X)) \\
&> U(\delta_G(X)) - U(\delta^v_G(X)) = U(\delta^h_G(X)) \\
\Rightarrow D(\delta_{\mathcal{N}}(X')) &\underbrace{>}_{\textit{from } (A.2)} U(\delta^h_G(X')) = U(\delta_G(X'))
\end{aligned} \tag{A.5}$$

Hence, X' also violates the cut condition. (A.5) implies

$$D(\delta_{\mathcal{N}}(X')) = \sum_{l_k \text{ crossing } v_{\lambda-1}} D(l_k) + \sum_{l_k \text{ crossing } v_\rho} D(l_k) > U(\delta_G(X')) = 2mU_h$$
$$\Rightarrow \text{ either } \sum_{l_k \text{ crossing } v_{\lambda-1}} D(l_k) > mU_h \text{ or } \sum_{l_k \text{ crossing } v_\rho} D(l_k) > mU_h$$

Thus, either $v_{\lambda-1}$ or v_ρ violates the cut condition, and Lemma 11 follows. Now, we turn to the case where X reaches the leftmost and/or the rightmost column.

Assume that $\lambda = 1$ and $\rho < n$ (the case where $\rho = n$ and $\lambda > 1$ can be dealt with in a symmetrical way). If X does not reach both the uppermost and lowermost lines, (d) implies that $\delta_G(X)$ contains at least ρ edges valued by U_v, since it contains at least an horizontal edge belonging to v_ρ, has at least ρ vertical edges (at most one from each column) and, from (b), at most one column has edges whose capacities are less than U_v. This implies that contradiction (A.1) still holds, and hence X necessarily reaches both the uppermost and the lowermost lines.

Moreover, for each vertical edge $(u \in X, w \in V \setminus X)$ on the first column (such an edge is not in $\delta_G(X')$), there exists an horizontal edge f_w on the same line as w which is in $\delta_G(X)$ and not in $\delta_G(X')$, and which is such that, on this horizontal line, there is no vertex in X between w and the leftmost vertex of f_w (since otherwise $w \in X$, see case II in Figure A.1). From (b) and (d), either $U((u,w)) = U_v \geq D^*$ or $U(f_w) = U_v \geq \max(U((u,w)), D^*)$, and thus, by replacing "$= U(\delta_G^v(X))$" by "$\leq U(\delta_G^v(X) \cup \{f_w / \exists (u,w) \in \delta_G(X)\})$", (A.4) still holds in this case. Eventually, $U(\delta_G^h(X) \setminus \{f_w / \exists (u,w) \in \delta_G(X)\}) \geq mU_h = U(\delta_G^h(X'))$, hence a proof similar to the one given above shows that (A.5) remains true. $\delta_G(X')$ being the ρ^{th} vertical strip, Lemma 11 follows.

Now, assume that $\lambda = 1$ and $\rho = n$. If X does not reach both the uppermost and the lowermost lines, then we have a contradiction since there exist edges $f_1, \ldots, f_n$ in $\delta_G(X)$ such that $\sum_{p=1}^{n} U(f_p) \geq \sum_{k=1}^{K} D(l_k)$ $(\geq D(\delta_{\mathcal{N}}(X)))$. Indeed, either each vertical edge is valued by $U_v \geq D^*$ (so we choose a vertical edge belonging to $\delta_G(X)$ on each one of the n columns and we are done), or $\delta_G(X)$ is either an horizontal strip or the union of two horizontal strips (so we apply (c), see case III in Figure A.1), or $\delta_G(X)$ contains an horizontal edge $\hat{e}$ (see case IV in Figure A.1), and, from (d), $\hat{e}$ is valued by U_v (so we construct a subset S of $\delta_G(X)$ by choosing f_n to be $\hat{e}$ and, for $p \in \{1, \ldots, n-1\}$, f_p to be a vertical edge from the p^{th} column; then, from (b), the horizontal strip h_i containing f_1 satisfies $(U(\delta_G(X)) \geq)\ U(S) \geq \sum_{\text{e is an edge of } h_i} U(e)$, and we can apply (c)).

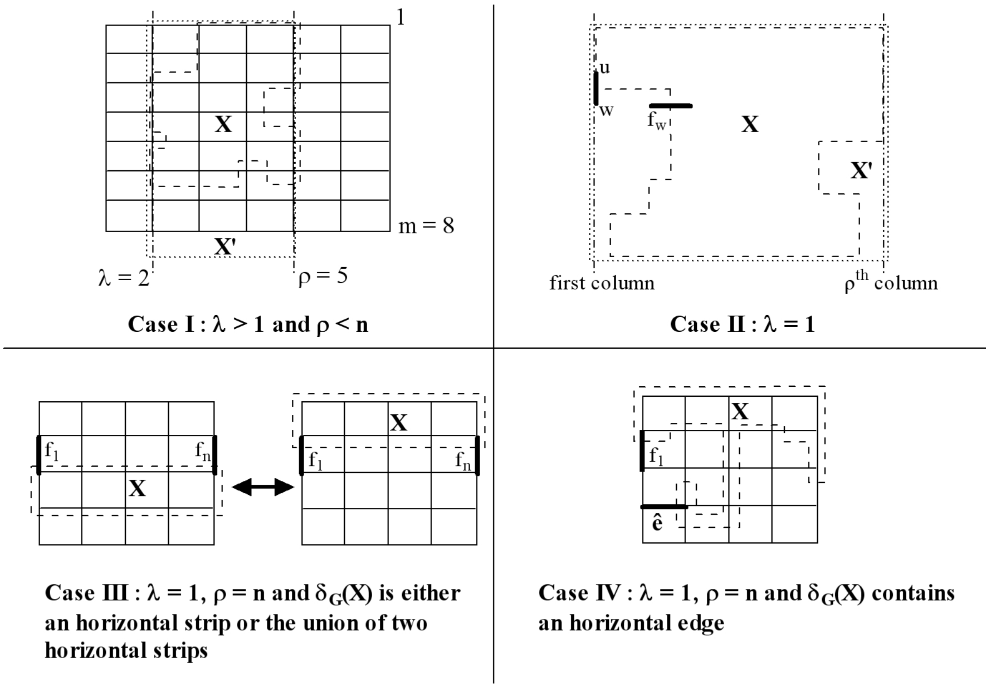


Figure A.1. Examples for Lemma 11.

Hence, X reaches the uppermost and lowermost lines. Then, for the same reasons as in the case where $\lambda = 1$ and $\rho < n$, (A.4) continues to hold, and since $U(\delta^h_G(X')) = 0$, it can be proved as previously that (A.5) remains true. Since $|\delta_G(X')| = |\delta_{\mathcal{N}}(X')| = 0$, we get a contradiction. $\square$

B Proof of Theorem 15

Recall that $K^* = Opt(CNP)$ if $m < d$, and $K^* = K$ otherwise.

Theorem 15 *If* (5) *does not hold,* c *is odd,* $K = n$ *and* $d \leq m < \lceil \frac{dc}{c-1} \rceil$*, then* $Opt(\text{MaxIMFUG}) = K\text{c} - 1$*; Otherwise* $Opt(\text{MaxIMFUG}) = K^*\text{c}$.

Proof. We need to distinguish between several cases. The proof of Theorem 15 will directly follow from the proofs of the next four lemmas. In the following, as in Section 4, whenever $m \geq d$ holds, we do not distinguish between $(G, \mathcal{N})$ and $(G, \mathcal{N}^-)$. Also recall that, whenever we deal with an instance of the demand multiflow problem, we can transform this instance into an equivalent one having a non augmented grid (see Section 4.1). So, when considering this problem, we always assume that grids are non augmented.

Table B.1 sums up the results of Lemmas 17, 18, 20 and 21.

<table>
<tr><td colspan="6">$c \geq 2$</td></tr>
<tr><td>c is even</td><td colspan="5">c is odd</td></tr>
<tr><td>Lemma 17</td><td>$m < d$</td><td colspan="2">$d \leq m < \lceil \frac{dc}{c-1} \rceil$</td><td colspan="2">$m \geq \lceil \frac{dc}{c-1} \rceil$</td></tr>
<tr><td rowspan="2">Opt. value
$= K^*c$</td><td rowspan="2">Lemma 18
Opt. value
$= K^*c$</td><td>$K < n$ or (5)
holds</td><td>$K = n$ and (5)
does not hold</td><td>$K < n$ or (5)
holds</td><td>$K = n$ and (5)
does not hold</td></tr>
<tr><td>Lemma 18
Opt. value
$= Kc$</td><td>Lemma 21
Opt. value
$= Kc - 1$</td><td>Lemma 18
Opt. value
$= Kc$</td><td>Lemma 20
Opt. value
$= Kc$</td></tr>
</table>

Table B.1
Summary of the proof of Theorem 15 ($K^* = K$ whenever $m \geq d$)

The case where c is even is straightforward.

Lemma 17 *Assume* c *is even. Then Opt*(MAXIMFUG) $= K^*c$.

Proof. As in the proof of Lemma 13, we consider $(G, \mathcal{N}^-)$ and define an instance $\mathcal{I} = (G, \mathcal{N}^-, U, D)$ of the demand multiflow problem, such that $D(l_k) = c$ for each net l_k in $\mathcal{N}^-$. Recall that the capacity function is defined by $U(e) = c$, for each edge e. Obviously, c being integral, U and D are integer-valued. Moreover, $U(e)$ and $D(l_k)$ being even for each edge e and net l_k respectively, the eulerian condition (12) holds. Since $(G, \mathcal{N}^-)$ satisfies $m \geq d$, then $\sum_{l_k \text{ crossing } v_j} D(l_k) \leq dc \leq mc = \sum_{e \text{ is an horizontal edge of } v_j} U(e)$ holds for each j, and so the cut condition holds for each vertical strip. Furthermore, $\mathcal{I}$ satisfies (a), (b), (c) and (d) (see Appendix A), so, from Lemma 11, Theorem 10 applies. From Corollary 14, this provides an integer multiflow having the same value as a multicut, and thus which is optimal: Lemma 17 follows. The proof of Theorem 10 being constructive, it also provides an algorithm to route the integral flows. $\square$

In the following of Section 5, we assume that c is odd. Lemma 18 settles several cases.

Lemma 18 *Assume that* c *is odd. Assume that* $m < d$, *or that* $m \geq d$ *and either* $K^* = K < n$ *or* (5) *is satisfied. Then, the optimal value of* MAXIMFUG *is* K^*c.

Proof. If $m < d$ and m is odd then, by Corollary 8, *Opt*(MAXEDP) $= K^*$. If $m < d$, m is even and $(G, \mathcal{N}^-)$ satisfies (5) or (6) or (7), then, by Theorem 4, we also have *Opt*(MAXEDP) $= K^*$. The same holds when either $m \geq d$ and (5) is satisfied, or $m > d$ and $K < n$. Moreover, if $m = d$ and m is odd then, by Proposition 7, $(G, \mathcal{N})$ satisfies (6) and hence $Opt(\text{MaxEDP}) = K = K^*$. Note that this is also true when $m = d$, m is even and $(G, \mathcal{N})$ satisfies either (6) or (7). In all these cases, we prove Lemma 18 by using the fact that, for MaxIMFUG, a feasible solution of value $K^*\mathrm{c}$ is obtained by routing c units of flow on each one of the K^* edge disjoint paths. As in the proof of Lemma 17, Corollary 14 implies that this provides an integer multiflow having the same value as a multicut, and thus which is optimal. In particular, this shows that whenever $(G, \mathcal{N}^-)$ satisfies the assumptions of Theorem 4, solving MaxIMFUG is achieved by solving MaxEDP.

The last case to consider in this lemma is the case where c is odd, $m \leq d$, $K^* < n$, m is even and $(G, \mathcal{N}^-)$ satisfies none of the three conditions (5), (6) and (7).

To settle this case, we only need to show that the instance of the demand multiflow problem $\mathcal{I} = (G, \mathcal{N}^-, U, D)$, with $U(e) = \mathrm{c}$ for each edge e and $D(l_k) = \mathrm{c}$ for each net l_k in $\mathcal{N}^-$, admits an integer solution. This will yield an integer multiflow of value $|\mathcal{N}^-|\mathrm{c} = K^*\mathrm{c}$. We start by transforming the instance into a new one by decreasing the capacity of several edges and by adding virtual nets to $(G, \mathcal{N}^-)$, so that the resulting instance satisfies the eulerian condition (12), and then we apply Theorem 10. Obviously, if the new instance is solvable, then the initial one admits an integer solution. First, for $(G, \mathcal{N}^-)$, we claim the following:

Claim 19 *Let s be the number of saturated vertical strips in $(G, \mathcal{N}^-)$. When $(G, \mathcal{N}^-)$ satisfies none of the three conditions* (5), (6) *and* (7)*, then it is such that, in each set X_j, $j \in \{2, \ldots, s\}$, containing all the vertices of the uppermost and lowermost lines which are located between the $j-1^{th}$ and the j^{th} saturated vertical strip, there is exactly zero or two non terminal vertices. Furthermore, let X_1 (resp. X_{s+1}) be the set of vertices of the uppermost and lowermost lines which are located on the left (resp. on the right) of the leftmost (resp. rightmost) saturated strip: then, any vertex in X_1 (resp. X_{s+1}) is a terminal.*

Proof. Consider the second part of Claim 19 first: from Theorem 4, X_1 (resp. X_{s+1}) is such that all of its vertices are terminal vertices, since otherwise $(G, \mathcal{N}^-)$ satisfies (6). Furthermore, in every X_j, $j \in \{2, \ldots, s\}$, there is at most two non terminal vertices (one being on the uppermost line, the other on the lowermost line), since otherwise $(G, \mathcal{N}^-)$ satisfies (7). To prove the first part of Claim 19, assume there exists a j such that in X_j, there is a unique non terminal vertex u. Let v_i^* be the i^{th} saturated vertical strip in $(G, \mathcal{N}^-)$, and let d_i^* denote its density. Note that in $(G, \mathcal{N}^-)$ there is an even number of free vertices on the lowermost and uppermost lines, since a net has two terminals. Thus, we can pair these non terminal vertices together, forming new *virtual nets* and obtaining a full grid: let $(G, \hat{\mathcal{N}}^-)$ be this new grid. u

has been paired with a vertex w: assume without loss of generality that w is on the right of v_j^*. Recall that in $(G, \mathcal{N}^-)$, both d_{j-1}^* and d_j^* are equal to m. Because of the structure of $(G, \mathcal{N}^-)$, all the new virtual nets crossing v_j^* (resp. v_{j-1}^*) cross v_{j-1}^* (resp. v_j^*), except (u, w) which crosses only v_j^*. Thus, if $\hat{K}_{j-1}$ denotes the number of virtual nets crossing v_{j-1}^*, the new densities of v_{j-1}^* and v_j^* in $(G, \hat{\mathcal{N}}^-)$, respectively denoted by $\hat{d}_{j-1}$ and $\hat{d}_j$, are given by:

$$\hat{d}_{j-1} = d_{j-1}^* + \hat{K}_{j-1} = m + \hat{K}_{j-1} \tag{B.1}$$

and

$$\hat{d}_j = d_j^* + (\hat{K}_{j-1} + 1) = m + \hat{K}_{j-1} + 1 \tag{B.2}$$

$(G, \hat{\mathcal{N}}^-)$ being a full grid, all its densities are even (see (3) in Section 3). However, from (B.1) and (B.2), $\hat{d}_{j-1}$ and $\hat{d}_j$ have a different parity, a contradiction. Claim 19 follows. $\square$

Now, let us transform the (non augmented) grid into a new one satisfying the eulerian condition. For each odd vertex u in a X_j, $j \in \{2, \ldots, s\}$, u is a non terminal vertex (since all the other vertices have a degree equal to 4c), and, from Claim 19, there always exists another unique non terminal vertex on the lowermost or on the uppermost line that belongs to X_j, say w: we add the net $(u,\ w)$ to $(G, \mathcal{N}^-)$. Call such nets *virtual nets*, as in Claim 19. Let $u_1, \ldots, u_m$ and $u_1^{'}, \ldots, u_m^{'}$ be the vertices on the leftmost and rightmost columns respectively, such that, for each i, u_i and $u_i^{'}$ are on the i^{th} horizontal line. For each $i \in \{1, \ldots, \frac{m}{2}\}$, we decrease the capacity of the edges (u_{2i-1}, u_{2i}) and $(u_{2i-1}^{'}, u_{2i}^{'})$ by one.

Let $(G, \hat{\mathcal{N}}^-)$ be the grid obtained from $(G, \mathcal{N}^-)$ by adding the virtual nets and then decreasing the capacities as explained above. We have $U((u_{2i-1}, u_{2i})) = U((u_{2i-1}^{'}, u_{2i}^{'})) = \mathrm{c} - 1$ for each i, and $U(e) = \mathrm{c}$ for any other edge e. We define $D((u, w)) = 1$ for each virtual net (u, w), and $D(l_k) = \mathrm{c}$ for each net l_k in $\mathcal{N}^-$. Let $\hat{\mathcal{I}} = (G, \hat{\mathcal{N}}^-, U, D)$. U and D are integer-valued, and $(G, \hat{\mathcal{N}}^-)$ satisfies the eulerian condition, since, for each $v \in V$, $U(\delta_G(v)) + D(\delta_{\hat{\mathcal{N}}^-}(v)) \in \{3\mathrm{c}-1, 3\mathrm{c}+1, 4\mathrm{c}\}$. Moreover, a virtual net does not cross any saturated vertical strip, so $(G, \hat{\mathcal{N}}^-)$ satisfies $m \geq d$, since $(G, \mathcal{N}^-)$ does. Obviously, $\hat{\mathcal{I}}$ satisfies (a), (b) and (d). Eventually, we show that (c) also holds. On the one hand, for each horizontal strip h_i, $\sum_{e \text{ is an edge of } h_i} U(e) \geq n\mathrm{c} - 2$. On the other hand,

$$\begin{aligned}\sum_{l_k \in \hat{\mathcal{N}}^-} D(l_k) &= \sum_{l_k \in \mathcal{N}^-} D(l_k) + \sum_{(u,w) \text{ is a virtual net}} D((u, w)) \\ &= K^*\mathrm{c} + |\{\text{virtual nets}\}|\end{aligned}$$

$$\leq K^*\mathrm{c} + (n - K^*) = K^*(\mathrm{c} - 1) + n$$

Since $K^* \leq n-1$, $K^*(\mathrm{c}-1)+n \leq n\mathrm{c}+(1-\mathrm{c})$. We have $\mathrm{c} \geq 2$ and c is odd, so $1 - \mathrm{c} \leq -2$. Hence $\sum_{l_k \in \hat{\mathcal{N}}^-} D(l_k) \leq n\mathrm{c} - 2 \leq \sum_{e \text{ is an edge of } h_i} U(e)$. Then (c) holds, and thus Lemma 11 and Theorem 10 apply. As a consequence, $\hat{\mathcal{I}}$ is solvable and admits an integer solution, so we can route an integer multiflow of value $K^*\mathrm{c} + |\{\text{virtual nets}\}|$ for $\hat{\mathcal{I}}$, and thus of value $K^*\mathrm{c}$ for the initial instance $\mathcal{I}$. Lemma 18 follows. □

We still have to deal with the case where $m \geq d$, $K^* = K = n$, c is odd and $(G, \mathcal{N})$ does not satisfy (5).

Lemma 20 *If* c *is odd,* $K = n$ *and* $m \geq \lceil \frac{d\mathrm{c}}{\mathrm{c}-1} \rceil$, *then* $Opt(\text{MaxIMFUG}) = K\mathrm{c}$.

Proof. For each net l_k in $\mathcal{N}$, we define $D(l_k) = \mathrm{c}$. We use the same idea as in Lemma 18 and transform the (non augmented) grid into a new one that satisfies the eulerian condition. The only odd vertices are the ones located on the leftmost and rightmost vertical lines, since all the other vertices have a degree equal to 4c. We construct the new grid by decreasing the capacity of each horizontal edge by one, and so we have $U(e) = \mathrm{c} - 1$ for each horizontal edge e and $U(e') = \mathrm{c}$ for each vertical edge e'. Let $(G, \hat{\mathcal{N}})$ be this new grid and let $\hat{\mathcal{I}} = (G, \hat{\mathcal{N}}, U, D)$. U and D are integer-valued, and $(G, \hat{\mathcal{N}})$ satisfies (a), (b), (c) and (d), and the eulerian condition as well, since, for each $v \in V$, $U(\delta_G(\{v\})) + D(\delta_{\hat{\mathcal{N}}}(\{v\})) \in \{3\mathrm{c}-1, 4\mathrm{c}-2\}$. Moreover, $(G, \hat{\mathcal{N}})$ satisfies the cut condition on each vertical strip v_j since, for each j, $\sum_{l_k \text{ crossing } v_j} D(l_k) \leq d\mathrm{c} \leq (\lceil \frac{d\mathrm{c}}{\mathrm{c}-1} \rceil)(\mathrm{c}-1) \leq m(c-1) = \sum_{e \text{ is an edge of } v_j} U(e)$. Hence, Lemma 11 and Theorem 10 apply, and $\hat{\mathcal{I}}$ is solvable and admits an integer solution: Lemma 20 follows. □

Lemma 21 settles the last case.

Lemma 21 *Assume* $(G, \mathcal{N})$ *does not satisfy* (5), c *is odd,* $K = n$ *and* $d \leq m < \lceil \frac{d\mathrm{c}}{\mathrm{c}-1} \rceil$. *Then, the optimal value of* MaxIMFUG *is* $K\mathrm{c} - 1$.

Proof. First, we show that one can route at most $K\mathrm{c} - 1$ units of flow. To do this, we just have to prove that the instance of the demand multiflow problem $\mathcal{I} = (G, \mathcal{N}, U, D)$, with $U(e) = \mathrm{c}$ for each edge e and $D(l_k) = \mathrm{c}$ for each net l_k in $\mathcal{N}$, does not admit an integer solution. Assume it does. The main point is that only an even amount of flow can cross each horizontal edge. Indeed, the total amount of flow to be routed is exactly equal to the capacity of each horizontal strip, hence, for each horizontal edge e, for each unit of flow routed "from right to left" through e, there must be one unit of flow routed "from left to right" through e. Thus, given a feasible routing, the unused capacity on each horizontal edge is at least one, and we can reduce each horizontal capacity from c to c $-$ 1 without affecting the routing. But then, $m(\mathrm{c}-1) < d\mathrm{c}$ (since $m < \lceil \frac{d\mathrm{c}}{\mathrm{c}-1} \rceil$) and the cut condition no longer holds, a contradiction.

Now, we show that we can route an integer multiflow of value $K\mathrm{c}-1$. Let $l_1 = (s_1, t_1)$ be the net in $\mathcal{N}$ whose left terminal is u_1, the left upper corner of the (non augmented) grid (assume without loss of generality that u_1 is s_1). We remove lines from $(G, \mathcal{N})$ until $m = d$. Obviously, d (and thus m) is even since $K = n$ implies that $(G, \mathcal{N})$ is a full grid (see (3) in Section 3). We add the new virtual net (u_m, t_1) (if $u_m \neq t_1$), and we decrease by one the capacity of each edge (u_{2i}, u_{2i+1}), $i \in \{1, \ldots, \frac{m}{2} - 1\}$, and of each edge (u'_{2i-1}, u'_{2i}), $i \in \{1, \ldots, \frac{m}{2}\}$. We define demands as $D(l_k) = \mathrm{c}$ for each net $l_k \neq l_1$ in $\mathcal{N}$, $D(l_1) = \mathrm{c} - 1$ and $D((u_m, t_1)) = 1$ (if $u_m \neq t_1$). We have $U(e) = \mathrm{c}$ for each edge e, except the (vertical) ones whose capacities have been decreased to $\mathrm{c}-1$ as explained. Let $(G, \hat{\mathcal{N}})$ be the grid obtained from $(G, \mathcal{N})$ by adding (u_m, t_1) (if $u_m \neq t_1$) and then decreasing the capacities, and let $\hat{\mathcal{I}} = (G, \hat{\mathcal{N}}, U, D)$.

$(G, \hat{\mathcal{N}})$ satisfies the cut condition on each vertical strip since (i) $(G, \mathcal{N})$ does, (ii) $D(l_1)$ has been decreased by 1, (iii) $D((u_m, t_1))$ has been set to 1 and (iv) l_1 crosses exactly the same vertical strips as $(u_m,\ t_1)$ does. Moreover, U and D are integer-valued, and the eulerian condition (12) holds, since, for each $v \in V$, $U(\delta_G(\{v\})) + D(\delta_{\hat{\mathcal{N}}}(\{v\})) \in \{3\mathrm{c}-1, 3\mathrm{c}+1, 4\mathrm{c}\}$ if $u_m \neq t_1$, $\{3\mathrm{c}-1, 4\mathrm{c}\}$ otherwise. Because of the way we decreased the capacities, there is exactly one vertical edge on each horizontal strip whose capacity has been decreased to c $-$ 1, so (c) holds. (a), (b) and (d) also holding, Lemma 11 and Theorem 10 apply, so $\hat{\mathcal{I}}$ admits an integer solution. Lemma 21 follows. $\square$

This completes the proof of Theorem 15. $\square$

Figure and table legends:

Legend of Figure 1: An instance with only $K^* - 1$ edge disjoint paths (in dashed lines).

Legend of Figure 2: A cut given by (CD).

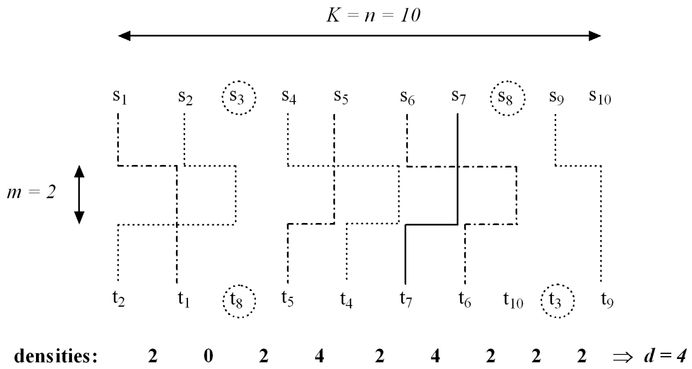
Legend of Figure A.1: Examples for Lemma 11.

Legend of Table B.1: Summary of the proof of Theorem 15 ($K^* = K$ whenever $m \geq d$)

Table B.1:

<table>
<tr><td colspan="6">$c \geq 2$</td></tr>
<tr><td>c is even</td><td colspan="5">c is odd</td></tr>
<tr><td rowspan="3">**Lemma 17**
Opt. value
$= K^*c$</td><td>$m < d$</td><td colspan="2">$d \leq m < \lceil \frac{dc}{c-1} \rceil$</td><td colspan="2">$m \geq \lceil \frac{dc}{c-1} \rceil$</td></tr>
<tr><td rowspan="2">**Lemma 18**
Opt. value
$= K^*c$</td><td>$K < n$ or (5) holds</td><td>$K = n$ and (5) does not hold</td><td>$K < n$ or (5) holds</td><td>$K = n$ and (5) does not hold</td></tr>
<tr><td>**Lemma 18**
Opt. value
$= Kc$</td><td>**Lemma 21**
Opt. value
$= Kc - 1$</td><td>**Lemma 18**
Opt. value
$= Kc$</td><td>**Lemma 20**
Opt. value
$= Kc$</td></tr>
</table>

Figure 1:



- $K^* = 8$ ((s_3, t_3) and (s_8, t_8) are removed)
- *Opt(MaxEDP)* = 7 ((s_3, t_3), (s_8, t_8) and (s_{10}, t_{10}) are not routed)

Figure 2:

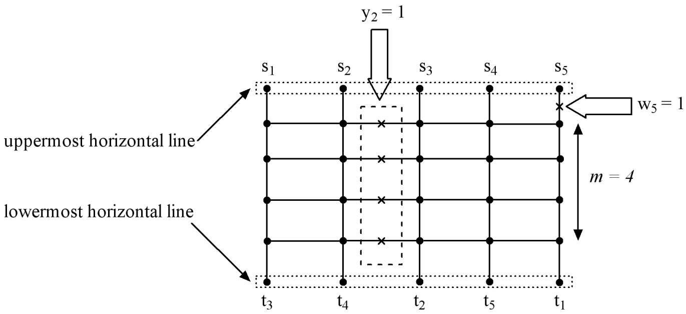
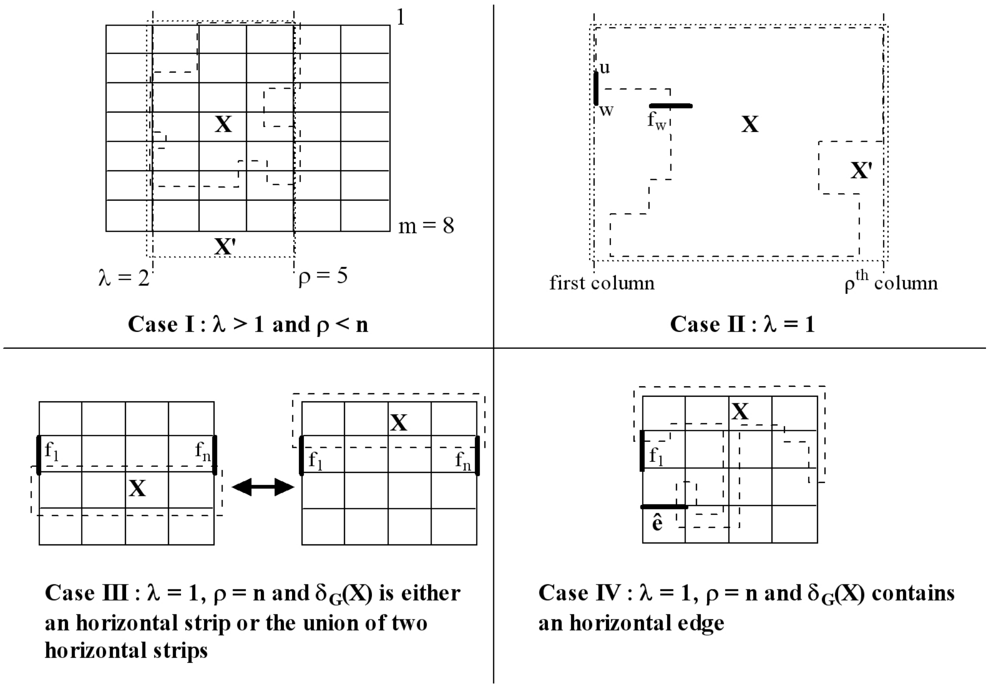

$y_2 = 1$
s_1
s_2
s_3
s_4
s_5
$w_5 = 1$
uppermost horizontal line
$m = 4$
lowermost horizontal line
t_3
t_4
t_2
t_5
t_1

Figure A.1:


X
X'
1
m = 8
λ = 2
ρ = 5
Case I : λ > 1 and ρ < n
u
w
f_w
X
X'
first column
ρ^th column
Case II : λ = 1
f_1
f_n
X
X
f_1
f_n
Case III : λ = 1, ρ = n and δ_G(X) is either an horizontal strip or the union of two horizontal strips
X
f_1
ê
Case IV : λ = 1, ρ = n and δ_G(X) contains an horizontal edge