

***UNE SOLUTION OPTIMALE POUR UN PROBLEME DE ROUTAGE DANS UN CANAL
DENSE AVEC CONTRAINTES DE CAPACITE***

Maria Zrikem

Equipe Combinatoire - Laboratoire CEDRIC,
CNAM 292 rue St-martin 75141 Paris cedex
zrikem@cnam.fr + 33 1 45 18 12 73

Résumé : Le routage est un problème de base dans la conception des circuits VLSI. Une instance du problème dans un canal dense consiste en une grille rectangulaire où tous les points des extrémités sont occupés par des terminaux. Ces derniers sont séparés en paires appelés « liaisons ». Les terminaux de chaque liaison appartiennent à deux extrémités différentes et doivent être reliés entre eux. Une solution (routage) est donnée par un ensemble de chemins connectant les différents terminaux en respectant la capacité donnée de chaque ligne et de chaque colonne.

Dans ce papier nous nous intéressons au problème de routage dans un canal dense avec des capacités paires sur les lignes et unitaires sur les colonnes. Nous l'appelons « un problème de multichemins ». Nous proposons un algorithme polynomial qui calcule des solutions optimales pour le problème et nous montrons que ces solutions sont toutes composées de plus courts chemins.

Summary : Channel routing is a basic problem in the design VLSI-circuits. An instance the problem in a dense channel consists of a rectangular grid-graph where all points of the lower and upper boundary are occupied by terminals. These are partitioned into pairs, called nets with one terminal on each boundary. These nets have to be connected. A solution (layout) is given by paths that connect corresponding grid points with respect of capacities given on each line and column of the grid.

In this note we deal with the routing problem in a dense channel with even capacities on lines and unit capacities on columns. We called it a « multipaths problem ». We propose a polynomial algorithm computing optimal solutions of the problem and prove that these solutions are all composed of shortest paths.

Mots clés : Optimisation Combinatoire, Multichemins, Grille, Routage.

Une solution optimale pour un problème de routage dans un canal dense avec contraintes de capacité

1 - INTRODUCTION

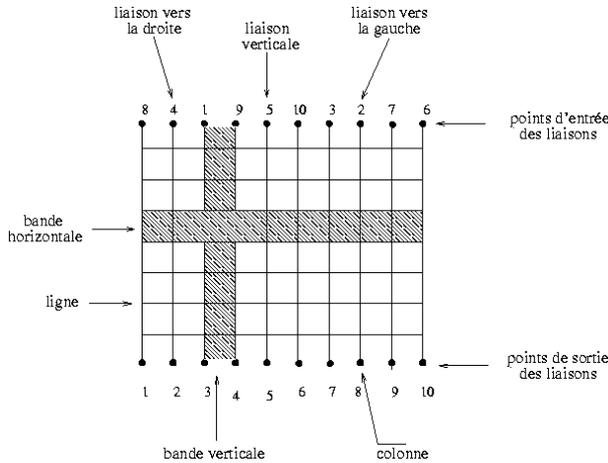


Figure 1 : Problème de multichemins dans une grille

Le problème de multichemins que nous traitons consiste en un ensemble de paires de sommets à relier sur une grille rectangulaire. Nous étudions en particulier sa résolutions dans une grille appelée par Forman, Wagner et Wagner (1993) « canal dense ». Chaque paire de sommets représente une liaison. Tous les tenants des liaisons sont situés sur l'une des deux extrémités de la grille et tous les aboutissants sur l'autre extrémité. De plus, les sommets des extrémités supérieure et inférieure sont tous occupés par un tenant ou un aboutissant (voir figure 1).

Chaque arête de la grille est munie d'une capacité fixe. Une solution du problème est donnée par un ensemble de chemins connectant les différents tenants et aboutissants, de longueur totale minimale et respectant les contraintes de capacité sur les arêtes de la grille.

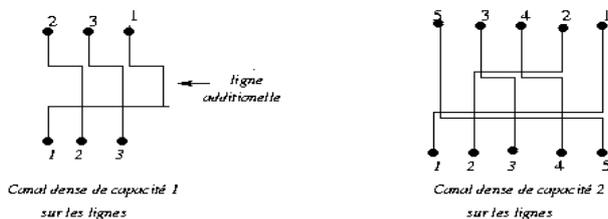


Figure 2 : Exemples de grilles avec différentes capacités

Le cas connu en littérature est celui d'une grille avec des capacités unitaires sur les arêtes. Dans ce cas, le problème devient un problème de multichemins arête-disjoints (voir figure 2). C'est un problème de base dans la conception des circuits VLSI. La plupart des travaux ont porté sur le problème de routage sans tenir compte de la longueur de la solution (Frank (1982), Gao, Kaufmann (1994), Kuchemn, Wagner, Wagner (1989),

Sarrafzadeh, Preparata (1985), Sarrafzadeh (1987a)). Peu de travaux ont pris en considération la longueur de la solution proposée : Sarrafzadeh (1987b) présente une borne de cette longueur. Kaufmann et Tollis (1988) proposent un algorithme calculant une solution de longueur totale minimale. Cependant les longueurs des solutions calculées par cet algorithme sur des exemples proposés par Sarrafzadeh (1987b), ont pu être améliorées plus tard par Forman, Dorothea Wagner et Frank Wagner (1993).

2 - DESCRIPTION DU PROBLEME

Nous présentons dans cette section certaines notions que nous utilisons dans le reste de ce papier. Le schéma représenté dans la figure 1 illustre toutes ces notions.

Grille

Une grille rectangulaire est déterminée par H lignes horizontales et V colonnes verticales. H est aussi dite la largeur de la grille.

On appelle bande verticale l'espace entre deux colonnes j et $j+1$. Elle est notée \vec{j} ou encore $\overleftarrow{j+1}$. De la même façon, l'espace limité par deux lignes est dit bande horizontale.

Liaison

Une liaison à router est définie par un tenant qu'on appellera *point d'entrée* et un aboutissant désigné par *point de sortie*. Une liaison l est donc donnée par la paire (E_l, S_l) , $1 \leq E_l \leq V$, $1 \leq S_l \leq V$ où E_l est la position du point d'entrée de la liaison sur la ligne supérieure et S_l est la position du point de sortie sur la ligne inférieure.

Une liaison est dite « vers la droite » (resp. « vers la gauche ») si $E_l < S_l$ (resp. $E_l > S_l$). Si $E_l = S_l$, on dit que la liaison l est « verticale ».

Une colonne j est simultanément le point d'entrée d'une liaison et le point de sortie d'une autre. Suivant que ces liaisons sont « vers la droite » ou « vers la gauche », j est de l'un des cinq types suivants (voir figure 3. La partie supérieure correspond au point d'entrée et la partie inférieure au point de sortie) :

- \lrcorner : si j est occupée par le point d'entrée d'une liaison «vers la gauche» et par le point de sortie d'une liaison «vers la gauche».
- \llcorner : si j est occupée par le point d'entrée d'une liaison «vers la droite» et par le point de sortie d'une liaison «vers la droite».

- \lrcorner : si j est occupée par le point d'entrée d'une liaison «vers la gauche» et par le point de sortie d'une liaison «vers la droite».
- \llcorner : si j est occupée par le point d'entrée d'une liaison «vers la droite» et par le point de sortie d'une liaison «vers la gauche».
- \lrcorner : si j est occupée par les deux points d'entrée et de

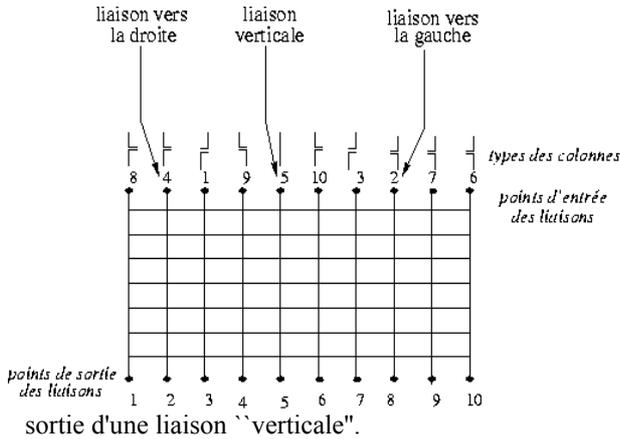


Figure 3 : Description d'un canal dense

Si on considère l'exemple de la figure 3, la colonne 2 est du type \llcorner car elle est occupée par le point d'entrée de la liaison 4 qui est «vers la droite», et le point de sortie de la liaison 2 qui est «vers la gauche».

Densité

On définit la densité locale d'une bande verticale \vec{j} comme étant le nombre de liaisons qui doivent traverser cette bande, on la note $d(\vec{j})$. La densité du problème est le maximum des densités locales $\max_j d(\vec{j})$ (pour des exemples voir figure 4). Une grille où toutes les liaisons sont verticales a donc une densité nulle.

La densité est au plus égale au nombre total des liaisons. Si les capacités sur les lignes sont unitaires, la densité d représente une borne inférieure de la largeur de la grille (nombre de lignes) nécessaire pour réaliser le routage, car la bande verticale de densité locale égale à d sera traversée par d liaisons.

Signalons aussi que dans le cas des capacités unitaires, il a été montré par Frank (1982) que le problème admet une solution si la largeur de la grille est au moins égale à la densité et si la grille a au moins une colonne additionnelle à droite ou à gauche.

Diagramme de densité

On désigne par un «segment unitaire horizontal» un segment horizontal de longueur égale à la largeur d'une bande. Le diagramme de densité est un diagramme obtenu en représentant les densités locales des bandes

verticales de la façon suivante : à chaque bande verticale de densité $2k$ sont associés k segments unitaires horizontaux superposés verticalement (voir figure 4). Nous dirons que ces segments sont «de flux 2», ils seront saturés dans la solution proposée dans la suite.

Le diagramme obtenu est composé de différents niveaux de densité où chaque niveau est constitué d'un ou plusieurs segments horizontaux. Un segment horizontal est la concaténation de plusieurs «segments unitaires horizontaux» (Voir figure 4).

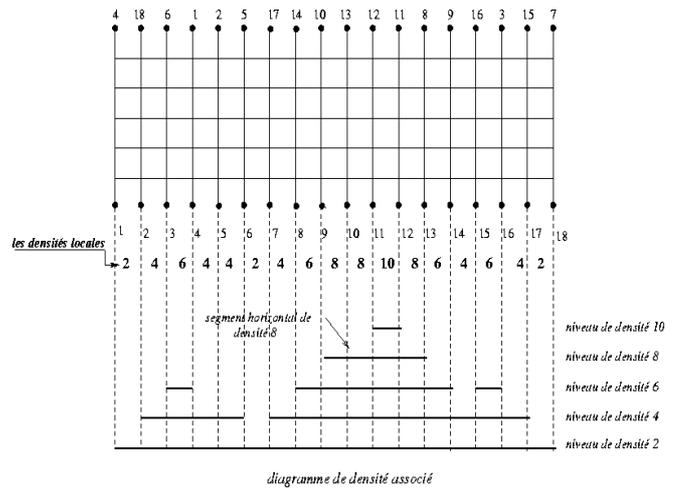


Figure 4 : Diagramme de densité associé à un problème de densité 10)

3 - GRILLE AVEC CAPACITES EGALES A 2 SUR LES LIGNES ET A 1 SUR LES COLONNES

Nous considérons dans ce paragraphe le cas d'une grille où les capacités sont égales à 2 sur toutes les lignes et à 1 sur les colonnes. Nous proposons un algorithme qui construit une solution optimale où chaque liaison est routée suivant l'un de ses plus courts chemins. La solution proposée sature toutes les arêtes horizontales de la grille correspondant au diagramme de densité. Les autres arêtes horizontales ne sont pas utilisées. Les arêtes verticales de la grille sont toutes utilisées entièrement. Autrement dit, le diagramme de densité nous donne l'allure de la solution qu'on va obtenir puisqu'il représente les arêtes horizontales qui vont être utilisées avec le flux sur chaque arête.

Nous allons voir dans la section 4 où nous traitons le cas général avec une capacité paire $c = 2p$ sur les lignes que les segments du diagramme de densité n'ont pas tous un flux égal $2p$ (ils ne sont pas tous saturés dans la solution proposée).

3.1 - Préliminaires

Les preuves de toutes les propriétés fournies dans cette section sont détaillées dans la thèse de Zrikem (2001).

Propriété 1 : La densité est toujours paire.

Dans la suite, si nous considérons une grille de densité $d = 2k$, nous utiliserons le nombre minimale de lignes nécessaires et qui est égal à $d/2 = k$. Nous supposons aussi sans aucune perte de généralité, qu'aucune liaison n'est «verticale».

Avant d'énoncer notre algorithme, nous avons besoin d'établir les deux propriétés suivantes concernant les segments du diagramme de densité.

Propriété 2 : Si \vec{j} est une bande verticale de densité $d(\vec{j})$, alors les bandes verticales adjacentes (à droite et à gauche) sont de densité $d(\vec{j})$ ou $d(\vec{j}) - 2$ ou $d(\vec{j}) + 2$ (voir figure 5).

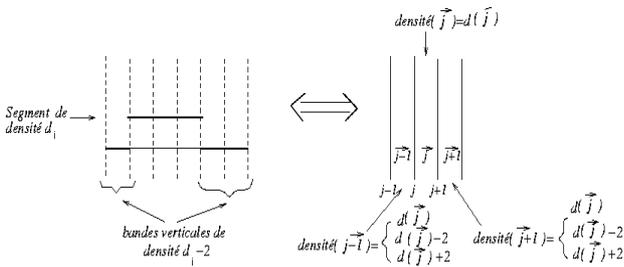


Figure 5 : Les segments de densité

Propriété 3 : Sur un segment de densité, la bande verticale de l'extrémité droite (resp. gauche) est du type \lrcorner (resp. \llcorner), les autres sont du type \lrcorner ou du type \llcorner (voir figure 6).

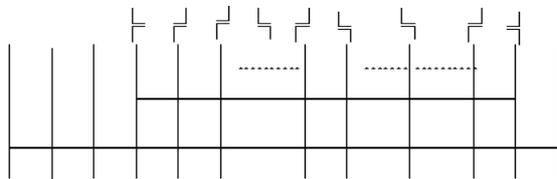


Figure 6 : Types des bandes verticales

3.2 - Un algorithme pour une solution optimale saturant le diagramme de densité

L'algorithme que nous proposons calcule une solution optimale où chaque liaison est routée suivant l'un de ses plus courts chemins. Les arêtes verticales ont une capacité égale à 1 et elles sont toutes utilisées dans la solution proposée. Si un canal est composé de H lignes, tous les plus courts chemins d'une liaison donnée comportent exactement $H+1$ arêtes verticales de la grille. Cela vient du fait que pour descendre de l'extrémité supérieure vers l'extrémité inférieure de la grille, il faut absolument traverser les H lignes de la grille (voir figure 7).

En ce qui concerne les arêtes horizontales dont la capacité est égale à 2, l'algorithme propose une solution qui sans créer de détours, sature les segments horizontaux du diagramme de densité. Les autres arêtes horizontales de la grille ne sont pas utilisées. La figure 7 schématise des chemins avec détours.

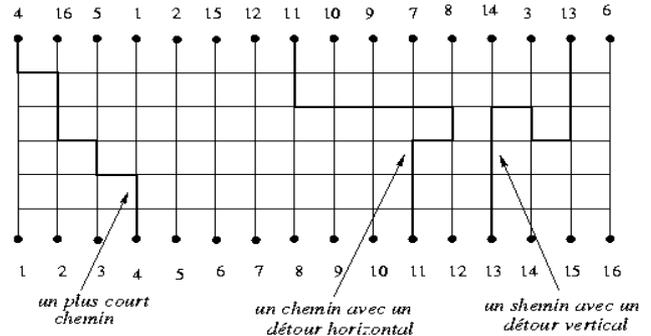


Figure 7 : Exemples de chemins sans ou avec détours

L'algorithme est basé sur le principe suivant :

On sature les niveaux de densité du diagramme un par un, du plus haut au plus bas.

– On sature un par un les segments de chaque niveau de la droite vers la gauche.

– Pour chaque segment $[a,b]$ du niveau courant on procède à :

1. Une phase de remplissage de droite à gauche qui consiste à «tirer» vers la gauche certaines liaisons du type « liaisons vers la gauche ». A la fin de cette phase, une unité de capacité du segment $[a,b]$ (la moitié) sera utilisée (figure 8).

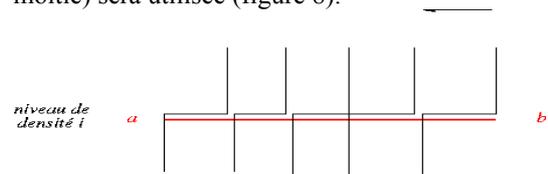


Figure 8 : Demi saturation du segment de densité $[a,b]$ à la fin d'une phase de remplissage de droite à gauche

2. Une phase de remplissage de gauche à droite qui est la symétrique de la première. Elle consiste à «tirer» vers la droite certaines liaisons du type « liaisons vers la droite » et permet de saturer entièrement le segment courant (figure 9).

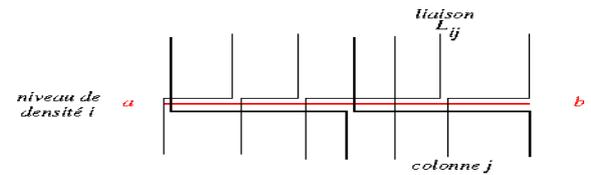


Figure 9 : Saturation entière à la fin d'une phase de remplissage de gauche à droite

Ainsi on peut dire que dans cet algorithme, à chaque étape i nous saturons le niveau de densité d_i , et nous obtenons un problème résiduel de densité $d_i - 2$.

En chaque point (colonne) j d'un segment de densité $[a,b]$ sur un niveau i , on a une liaison qui « arrive » verticalement du niveau précédent ; on note le numéro de cette liaison $L_{i,j}$ (voir figure 10).

Étant donné un canal dense de densité $d=2k$ et de n liaisons, les étapes de cet algorithme sont décrites par le pseudo-code qui suit :

Procédure min-cap-multichemins

Router toutes les liaisons vers le bas d'un segment unitaire vertical;

Pour j de 1 à n **Faire**

$L_{k,j} \leftarrow j$;

/ au niveau initial k , chaque liaison j « arrive » verticalement sur la colonne j */*

FinFaire

*/*descendre d'une arête verticale*/*

Pour i de k à 1 avec un pas -1 **Faire**

*/*Pour chaque niveau de densité du haut en bas*/*

Pour tout segment $[a,b]$ du niveau i **Faire**

Remplissage_droite_a_gauche($[a,b],i$);

*/*saturer le segment à moitié */*

Remplissage_gauche_a_droite($[a,b],i$);

/ saturer le segment entièrement */*

FinFaire

Router toutes les liaisons vers le bas d'un segment unitaire vertical;

Pour tout point intermédiaire du niveau i **Faire**

/ point où la liaison qui "arrive" verticalement n'a été tirée ni à droite ni à gauche*/*

$L_{i-1,j} \leftarrow L_{i,j}$;

FinFaire

Finfaire

FinProcédure

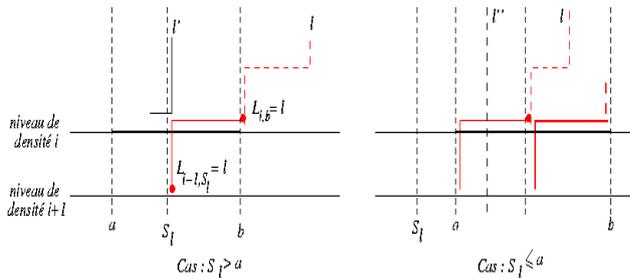


Figure 10 : Routage des liaisons dans une phase de remplissage de droite à gauche

La procédure `remplissage_gauche_a_droite()` est la symétrique de la procédure `remplissage_droite_a_gauche()`. Nous présentons uniquement le pseudo code de la deuxième procédure qui est illustrée par la figure 10.

Procédure `Remplissage_droite_a_gauche([a,b],i)`

$l \leftarrow L_{i,b}$; */* liaison qui "arrive" verticalement en b */*

$a_est_atteint \leftarrow Faux$;

Tant que `non(a_est_atteint)` **Faire**

Si $S_i > a$ **Alors** */* on n'a pas atteint a (cas1)*/*

/ S_i point de sortie de l est rencontré avant a */*

Router la liaison l horizontalement vers la gauche jusqu'à S_i ;

$L_{i-1,S_i} \leftarrow l$;

/ l « arrive » verticalement au niveau $i-1$ en colonne S_i */*

$l \leftarrow L_{i,S_i}$;

*/*On considère la nouvelle liaison qui « arrive » horizontalement au niveau i au point S_i */*

Sinon */* a est atteint : cas2 */*

Tirer la liaison l horizontalement vers la gauche jusqu'en a ;

$L_{i-1,a} \leftarrow l$;

/ l "arrive" verticalement au niveau $i-1$ en colonne a */*

$a_est_atteint \leftarrow Vrai$;

Finsi

Finfaire

FinProcédure

La procédure `remplissage_gauche_a_droite()` est obtenue en échangeant partout a et b ainsi que droite et gauche, le test «si $S_i > a$ » étant remplacé par «si $S_i < b$ ».

Un exemple

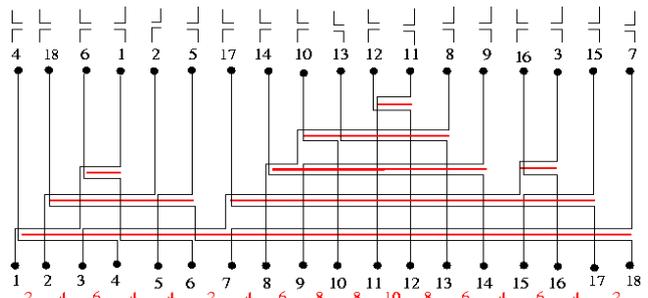


Figure 11 : Le routage obtenu pour un problème de densité 10

Nous présentons dans la figure 11 la solution obtenue par notre algorithme pour la grille schématisée dans la figure 4. La densité étant égale à 10, seulement 5 lignes sont utilisées

3.3 - Preuve de l'algorithme

Nous discutons dans cette section la validité de l'algorithme que nous avons proposé dans la section précédente. Cette discussion s'intéresse uniquement à la procédure de `remplissage_gauche_a_droite()`. Nous montrons qu'à chaque étape de cette procédure il est possible de router les liaisons considérées sans créer de détours ni verticaux ni horizontaux. Ensuite, nous établissons qu'à la fin de cette procédure, le segment de densité concerné est saturé.

Le routage est toujours possible

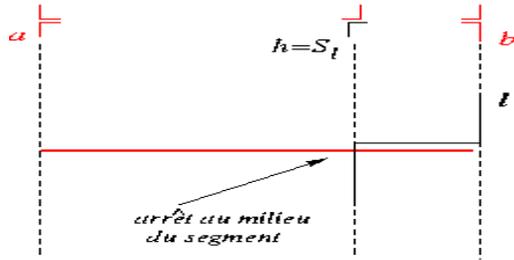


Figure 12 : Routage toujours possible sur un segment de densité

L'algorithme tel qu'il est décrit ci-dessus permet que le routage des liaisons soit toujours effectué. En effet :

– On a vu précédemment que l'extrémité droite de chaque segment de densité est du type \downarrow . Ainsi, il est

toujours possible de tirer vers la gauche la première liaison « vers la gauche » située sur la colonne b . Quand l'algorithme arrête la phase de remplissage avant l'extrémité du segment courant (cas 1) à la colonne h , pour que le routage ne se bloque pas, la colonne h doit être du type \downarrow . Lorsque l'algorithme

arrête la phase remplissage avant l'extrémité du segment, la liaison courante l , qui est « vers la gauche », a donc son point de sortie sur la colonne $h = S_l$. Par la suite cette colonne est du type \downarrow (voir

figure12). Or, toute liaison au milieu d'un segment de densité ne peut être que du type \downarrow ou du type \uparrow .

Donc la colonne h est nécessairement du type \downarrow , ce

qui permet de continuer le routage là où la liaison l s'est arrêtée.

Saturation de chaque segment de densité

Dans une phase de remplissage de droite à gauche, on commence à partir de l'extrémité droite et on tire continûment vers la gauche jusqu'à atteindre l'extrémité gauche du segment courant. Chaque liaison continue là où la précédente s'est arrêtée. De ce fait, à la fin de cette phase le segment est saturé à moitié.

Le même principe permet que le segment de densité soit entièrement saturé à la fin d'une phase de remplissage de gauche à droite.

Pas de détours

L'algorithme proposé ne crée pas de détours dans les chemins calculés. En effet :

– les liaisons ne sont jamais tirées vers le haut. Ceci permet d'éviter les détours verticaux.

– toute liaison du type « liaison vers la droite » est toujours tirée vers la droite et toute liaison du type « liaison vers la gauche » est toujours tirée vers la gauche. Ceci permet d'éviter des détours horizontaux.

Donc tous les chemins sont de longueur minimale.

Propriété de persistance

A chaque itération i de la procédure *min-cap-multichemins*, après la saturation du niveau i de densité

$d_i = 2i$, on obtient un problème résiduel de multichemins de longueur totale minimale dans une grille de densité $d_i - 2$ (voir figure 13)

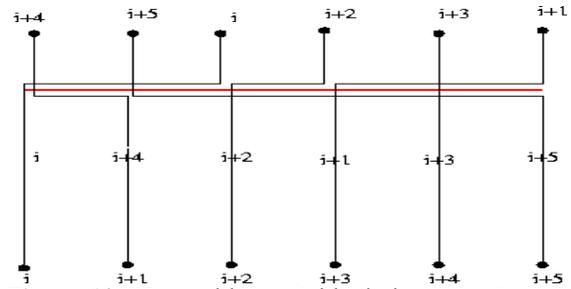


Figure 13 : Le problème résiduel obtenu après une itération

Après l'itération i , les segments du niveau i sont saturés. Chaque bande verticale de ces segments a été traversée par deux liaisons (une de droite à gauche et l'autre de gauche à droite) qui ne vont plus la traverser dans les étapes suivantes du fait que l'algorithme ne génère pas de détours. Les bandes verticales qui ne sont pas couvertes par les segment du niveau i , ont des densités locales inférieures ou égale à $d_i - 2$ (propriété2). Ces densités ne changent pas après l'itération i puisque les liaisons de chaque segment du niveau i n'avancent pas au delà de ce segment. Donc, le problème résiduel obtenu a une densité égale à $d_i - 2$.

De plus, il y a un point d'entrée et un point de sortie sur chaque colonne et la position des points de sortie est invariante. Les valeurs $L_{i-1,j}$, fournissent les nouveaux points d'entrée. On a correctement calculé la valeur $L_{i-1,j}$ pour chaque $j=1, \dots, n$. En effet, dans les deux procédures de remplissage, le résultat de ce calcul consiste à permuter entre elles les colonnes j où les liaisons sont tirées à droite ou à gauche. Dans la procédure principale, ce calcul consiste à garder la même position pour les colonnes intermédiaires (la permutation identité).

A l'étape finale, le problème résiduel est de densité $0 = d - \underbrace{2 - 2 - \dots - 2}_{d/2 \text{ fois}}$ et toutes les liaisons sont

verticales. Alors chaque point d'entrée a atteint le point de sortie qui lui correspond.

3.4 - Complexité de l'algorithme

L'algorithme que nous proposons a un temps d'exécution proportionnel à la superficie de la section du routage : soit à $(n - 1) * d/2$, où n est le nombre des liaisons et d la densité de la grille (i.e. $d/2$ est le nombre de lignes).

A chaque itération, dans les procédures *Remplissage_droite_à_gauche([a,b],i)* et *Remplissage_gauche_à_droite([a,b],i)*, la boucle « Tant que » est itérée au maximum $b-a$ fois (longueur du segment $[a,b]$), puisque partant de la colonne b en se déplace au moins d'une

arête à chaque passage jusqu'à atteindre la colonne a . Or, $(b - a) \leq n$, donc la boucle « *Tant que* » est en $O(n)$. Comme la procédure principale est effectuée $k = d/2$ fois, on peut conclure que l'algorithme que nous proposons est en $O(nd)$.

4 - GENERALISATION AU CAS D'UNE GRILLE AVEC DES CAPACITES PAIRES ≥ 2 SUR LES LIGNES ET EGALE A 1 SUR LES COLONNES

Les résultats et l'algorithme proposés dans le paragraphe précédent peuvent être généralisés à une grille avec des capacités unitaires sur les colonnes et des capacités paires sur les lignes d'une valeur $c = 2p \geq 2$.

Notons que dans ce cas, si la densité est d alors le nombre de lignes nécessaires H vaut d/c si $\text{mod}(d/c) = 0$ ($\text{mod}()$: reste de la division entière), sinon, il vaut $d/c + 1$.

Le cas général admet aussi une solution optimale formée des plus courts chemins. Pour obtenir cette solution, nous procédons de la façon suivante :

4.1 - Du cas général au cas simple

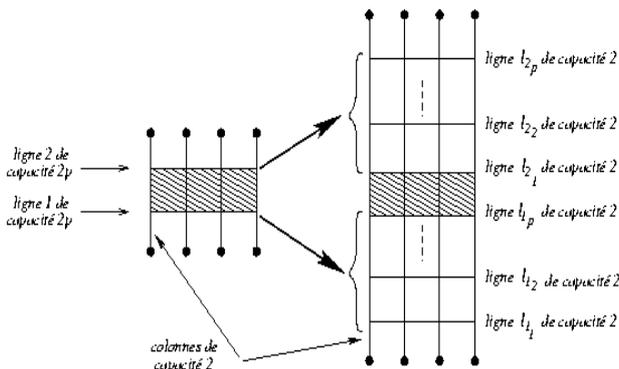


Figure 14 : Du cas général au cas simple

Soit une grille de densité d composé de H lignes, où la capacité sur chaque ligne est égale à $2p$ et est unitaire sur chaque colonne. Nous transformons cette grille en éclatant chaque lignes i (de capacité $2p$) en p lignes i_1, i_2, \dots, i_p , chaque ligne étant de capacité 2. La figure 14 illustre cette transformation.

De cette façon, nous obtenons une grille de même densité d et ayant $H.p$ lignes de capacité 2. La capacité des colonnes est unitaire. Nous pouvons donc appliquer l'algorithme *min-cap-multichemins* pour obtenir une solution optimale qui route chaque liaison sur un plus court chemin.

4.2 - Vers une solution du cas général

Pour déduire une solution pour le cas général (grille initiale), nous proposons de regrouper dans la grille obtenue chaque paquet de lignes $\{i_1, i_2, \dots, i_p\}$

($1 \leq i \leq H$) et de le remplacer par une seule ligne i .

Ceci revient à annuler dans la solution du cas simple toutes les transitions verticales effectuées par les liaisons au niveau des paquets $\{i_1, i_2, \dots, i_p\}$, $1 \leq i \leq H$ (voir figure 15).

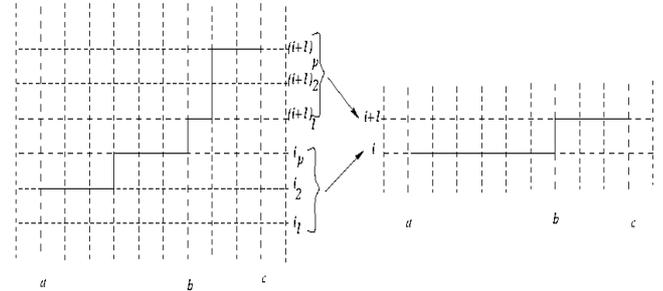


Figure 15 : Déduction d'une solution du cas général

Nous avons montré dans Zrikem (2001) que la solution ainsi obtenue est une solution admissible pour le cas général et qu'elle est sans détours.

5 - CAS DES CAPACITES IMPAIRES SUR LES LIGNES

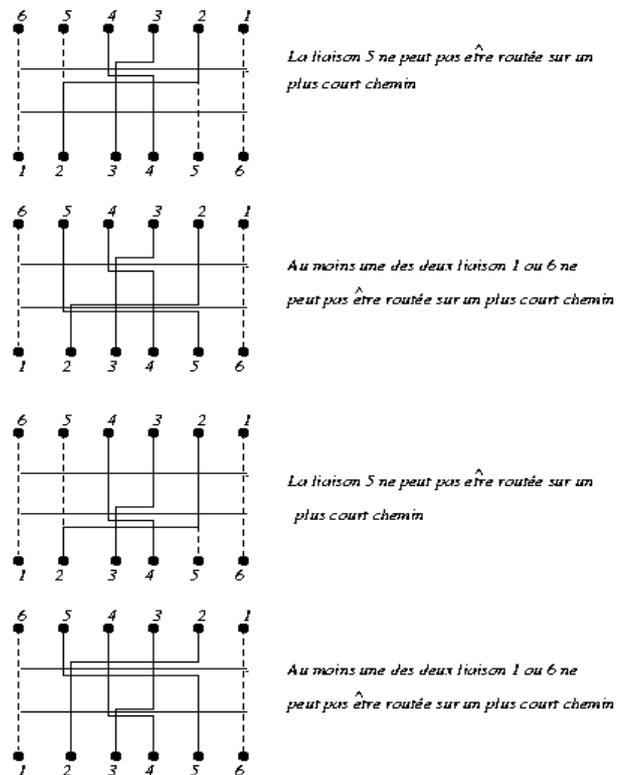


Figure 16 : Exemple n'admettant aucune solution des plus courts chemins

Contrairement au cas d'une grille avec des capacités paires sur les lignes, le problème de multichemins avec longueur totale minimale dans une grille avec des capacités impaires sur les lignes et unitaires sur les colonnes, n'admet pas toujours une solution optimale des plus courts chemins.

Le cas d'un canal dense avec des capacités unitaires sur les lignes et les colonnes admet une solution si le canal a au moins une colonne additionnelle tout à fait à droite ou tout à fait à gauche (d'après un résultat de Frank (1982)). Cette solution peut contenir des détours. En ce qui concerne le cas général avec des capacités impaires >1 sur les lignes, nous donnons dans la figure 16 un exemple de capacité égale à 3 sur les lignes pour lequel toutes les solutions qui routent les liaisons sur des plus courts chemins sont non admissibles : dans cet exemple, pour respecter les contraintes de capacité égales à 1 sur les colonnes, la liaison 3 et la liaison 4 doivent se croiser horizontalement sur la même ligne, et également la liaison 2 avec 5 et 1 avec 6. Ainsi, on peut énumérer quatre cas de figures. Dans chaque cas, au moins une liaison ne peut pas être routée sur un plus court chemin.

6 - CONCLUSION

Nous avons considéré dans ce travail un problème de multichemins dans une grille sous contraintes de capacité et de longueur totale minimale.

Nous avons proposé un algorithme linéaire en la taille de la grille qui résout le problème si les capacités sont unitaires sur les colonnes et paires sur lignes. Cet algorithme route chaque liaison sur un plus court chemin.

Nous avons montré également sur un exemple que le cas des capacités impaires sur les lignes n'admet pas toujours une solution des plus courts chemins. Nous essayons actuellement de voir si l'algorithme proposé dans [Forman, 1993] pour le cas des capacités unitaires sur les lignes et sur les colonnes peut être généralisé au cas des capacités impaires quelconques sur les lignes.

7 - BIBLIOGRAPHIE

- Forman M., Wagner D. Wagner F. (1993), "Routing through a dense channel with minimum total wire length". *Journal of Algorithms*, Vol. 15, p. 267-283 (1993).
- Frank A. (1982), "Disjoint paths in a rectilinear grid". *Combinatorica*, Vol. 2, p. 361-371 (1982).
- Frank A. (1990), "Paths, flows, and VLSI-layout, chap. Packing paths, circuit, and cuts: a survey". Springer Verlag ED, p. 47-1000 (1990).
- Gao S., Kaufmann M. (1994), "Channel routing of multiterminal nets". *Journal of the ACM (JACM)*, Vol. 41, N. 4, p. 791-818 (1994).
- Kaufmann M., Tollis I. G. (1988), "Channel routing with short wires". *Proceedings Aegean Workshop on computing (AWOC'88)*, Lecture notes in Computer Science, Springer -Verlag ed. Vol. 319, p. 226-236, New York/Berlin (1988).
- Kuchem R., Wagner D., Wagner F. (1989), "Atrea optimal three-layer channel routing". *Proceeding of 30th Annual symposium foundations of computer science (FOCS'89)*, p. 506-5011 (1989).
- Sarrafzadeh M., Preparata F. P. (1985), "Compact channel routing of multiterminal nets". *Ann. Discrete Math* Vol. 25, p. 255-280 (1985).
- Sarrafzadeh M. (1987a), "On the complexity of the general channel routing problem in the Knock-knee mode". *IEEE Transactions, Computer-Aided Design Integrated Circuits Systems*, Vol. 4, p. 503-506 (1987).
- Sarrafzadeh M. (1987b), "Channel routing with provably short wires", *IEEE Transactions, Circuits systems*, Vol. 34, p. 1133-1135 (1987).
- Zrikem M. (2001), "Optimisation du routage de câbles dans les installations de productions d'électricité". Thèse de doctorat, Département d'informatique, CNAM, Paris (2001).