N° d'ordre : 15

THÈSE

présentée devant

L'ÉCOLE NORMALE SUPÉRIEURE DE LYON

pour obtenir

le Titre de Docteur de l'École Normale Supérieure de Lyon
spécialité : Informatique

au titre de la formation doctorale d'Informatique de Lyon

par **Cyril GAVOILLE**

COMPLEXITÉ MÉMOIRE DU ROUTAGE DANS LES RÉSEAUX DISTRIBUÉS

Soutenue le 18 Janvier 1996

Après avis de : BERMOND Jean-Claude
PELEG David (Weizmann Institute, Israël)

Devant la commission d'examen formée de :

BERMOND Jean-Claude (Président du jury)
COSNARD Michel
FRAIGNIAUD Pierre (Directeur de thèse)
KÖNIG Jean-Claude
KRANAKIS Evangelos (Carleton University, Canada)

Complexité mémoire du routage dans les réseaux distribués

Au plus bas niveau des communications, le routage consiste à décider localement de la route des messages transitant par les nœuds intermédiaires d'un réseau distribué de processeurs dans un ordinateur parallèle, ou, plus généralement, dans un réseau distribué d'ordinateurs. Nous étudions les caractéristiques locales du « routeur » (coprocesseur effectuant le routage) par rapport à la globalité du réseau communiquant. Précisemment, nous mesurons la quantité minimale d'information nécessaire par chacun des routeurs pour assurer des communications rapides sur n'importe quel type de réseau.

Memory complexity of routing in distributed networks

Routing consists in locally decide routes of messages traveling trough intermediary nodes of a distributed network of processors in a parallel computer, or, more generally, in a computer's network. We study local characteristics of the "router" (coprocessor that computes routing) in comparison with the globality of the communication network. Precisely, we measure the minimum memory requirement of each router to ensure fast communications on arbitrary networks.

Remerciements

Je tiens particulièrement à remercier la DRET pour le financement de mes travaux. Je tiens également à remercier les organismes suivants :

- l'association AFIRST pour le financement de mon séjour en Israël ;
- la coopération PICS France-Canada pour le financement de mes différentes missions au Canada ;
- le projet MAP pour les missions Européennes ;
- la société IBM et spécialement le *T. J. Watson Research Center* pour mon séjour à New York ;
- enfin les projets PRC ANM, PRC PRS pour leurs supports.

Je tiens également à remercier les personnes suivantes :

- Jean-Claude Bermond pour avoir été rapporteur de ma thèse malgré ses nombreuses contraintes internationales ;
- David Peleg pour m'avoir invité au Weizmann Institute à Rehovot et pour avoir accepté la lourde tâche de rapporteur ;
- Michel Cosnard pour sa patience infinie et Jean-Claude König pour être membres de mon jury ;
- Evangelos Kranakis et Danny Krizanc pour mon séjour à Carleton, et spécialement Evangelos pour avoir accepté d'être membre de mon jury lors de ma soutenance ;
- Zvi Galil et Michele Flammini pour m'avoir invité à Columbia University ;
- Shay Kutten d'IBM Watson pour mon séjour à Manhattan et à Yorkstown ;
- Joe Peters et Art Liestman pour mon séjour à Vancouver ;
- tous les membres de "Rumeur" ;
- mais aussi toutes les personnes qui m'ont apporté leur soutien scientifique, en particulier Eli Upfal, Nicolas Santoro et Richard Tan ;
- et bien sûr Pierre Fraigniaud sans qui tout cela ne serait jamais arrivé.

Il serait injuste d'oublier tout ceux qui m'ont permis de réaliser ma thèse dans les meilleures conditions possibles :

- Marianne et Odile sans qui je n'aurai jamais découvert le LIP, en particulier Marianne pour ses traductions allemandes ;
- Sandrine et Éric, mes collègues de bureau, pour m'avoir soutenu sans compter surtout dans les derniers préparatifs de la soutenance ;
- Pascal pour les `\if` sioux de LaTeX ;
- Serge pour les commandes PostScripts et les transparents couleurs ;
- tous les athlètes de "Radio Sports LIP" que j'ai parfois torturés sans scrupules pour aller courir sur le stade ou pour sauter dans les couloirs du LIP, je pense à Jean-Marc, Xavier,

Richard, Frédéric, Michelle, Pierre, Éric, Didier, ... ;
- Laurent pour son support vidéo ;
- toutes les personnes du LIP que j'ai innondées de discours scientifiques pas toujours très clairs, dont les principales victimes ont été Pascal, Éric, Sandrine, Thierry, Jacques-Olivier, Arnaud et Bernard ;
- et enfin tout le personnel du LIP que j'ai harcelé de questions pas toujours des plus pertinentes il est vrai, merci Valérie, Éliane, Sylvie et Jocelyne.

Table des matières

Notations

Dans cette thèse, nous utilisons de nombreuse notions liées de la théorie des graphes. Nous en rappellerons les définitions au fur et à mesure de leur nécessité. $\mathbb{N}$ désigne l'ensemble des entiers naturels $\{0, 1, 2, \ldots\}$. Soient f et g des fonctions à valeur entière. Nous utilisons les notations suivantes pour comparer le comportement asymptotique de ces fonctions.

$f(n) = \Omega(g(n))$ exprime le fait que f est bornée inférieurement par g.

$$f(n) = \Omega(g(n)) \quad \Longleftrightarrow \quad \exists\, c > 0,\, \exists\, n_0,\, \forall\, n \geqslant n_0,\ |f(n)| \geqslant c|g(n)|$$

$f(n) = O(g(n))$ exprime le fait que f est bornée supérieurement par g.

$$f(n) = O(g(n)) \quad \Longleftrightarrow \quad g(n) = \Omega(f(n))$$

$f(n) = \Theta(g(n))$ exprime le fait que f est est asymptotiquement du même ordre que g.

$$f(n) = \Theta(g(n)) \quad \Longleftrightarrow \quad \left\{ \begin{array}{c} f(n) = \Omega(g(n)) \\ \text{et} \\ f(n) = O(g(n)) \end{array} \right.$$

$f(n) = o(g(n))$ exprime le fait que f est asymptotiquement négligeable devant g.

$$f(n) = o(g(n)) \quad \Longleftrightarrow \quad \forall\, \varepsilon > 0,\, \exists\, n_0,\, \forall\, n \geqslant n_0,\ |f(n)| \leqslant \varepsilon|g(n)|$$

Ainsi, $f(n) = g(n) + o(g(n))$ exprime le fait que f est asymptotiquement égale à g.
On notera $f(n) \sim g(n)$.

$$f(n) = g(n) + o(g(n)) \quad \Longleftrightarrow \quad \lim_{n \to \infty} \left| \frac{f(n)}{g(n)} \right| = 1$$

CHAPITRE 1 Introduction

L'interconnexion de plusieurs processeurs dans un même ordinateur permet d'augmenter la puissance de calcul de ce dernier par rapport à celle d'un ordinateur séquentiel. De la même manière, la connexion en réseaux d'un ensemble d'ordinateurs permet de résoudre potentiellement des problèmes de plus en plus grands et de plus en plus complexes, d'une part en distribuant les données et, d'autre part, en mettant en parallèle les calculs. Cependant, ces réseaux de processeurs ou d'ordinateurs diffèrent des ordinateurs séquentiels par le fait qu'ils engendrent d'importants mouvements de données entre leurs composantes.

Les communications entre les unités de calculs du réseau, appelé « nœuds », interagissent fortement les unes avec les autres lorsque le nombre de nœuds devient grand. En effet, pour des raisons évidentes liées à la complexité de réalisation matérielle, il est impossible de relier ensemble par un lien direct tous les nœuds d'un grand réseau. Par conséquent, certaines communications doivent transiter par des nœuds intermédiaires non nécessairement concernés par celles-ci. Pour ne pas ralentir les messages communiqués ou l'activité des nœuds intermédiaires, il convient d'associer à chacun des nœuds un mécanisme de routage des messages indépendant et automatique.

Au plus bas niveau des communications, le routage consiste à décider de la route des messages transitant par des nœuds intermédiaires. Cette décision est réalisée sur chaque nœud de façon automatique par un processeur dédié, appelé « routeur ». Le routeur détermine une route pour les messages le traversant. La décision s'effectue en fonction d'une information transportée par le message (le plus souvent l'adresse du destinataire) et d'une connaissance locale détenue par le routeur (table ou algorithme spécifique de routage).

Dans cette thèse nous allons étudier les caractéristiques locales du routeur. Pour introduire la problématique du routage « compact » considérons l'exemple suivant.

Imaginons un voyageur qui se déplace dans une ville qu'il ne connaît pas. Pour fixer les idées, supposons qu'il circule à Manhattan, un quartier de la ville de New York. Il désire se rendre au *500 West 120th Street*. Supposons qu'un agent de la circulation soit stationné à chaque intersection de rues et d'avenues. Notre voyageur peut demander sa route au premier agent rencontré qui répondra suivant le cas : au nord, au sud, à l'est ou à l'ouest. La question que l'on se pose est de mesurer la connaissance requise par l'agent pour permettre un routage vers n'importe quelle adresse de Manhattan. La règle de routage est fortement simplifiée car, à Manhattan, les rues sont numérotées par ordre croissant du sud vers le nord, et dans chaque rue les adresses sont numérotées de l'est vers l'ouest. Autrement dit, il est relativement « simple » pour l'agent de router notre voyageur connaissant son adresse de destination. Transposons cet exemple dans la ville de Paris. On peut se poser la même question à propos de la connaissance requise par un agent désirant router un voyageur à destination, par exemple, du *1er rue de la Paix*. Cependant, la situation pour Paris est plus complexe pour deux raisons : la structure du réseau routier n'est pas une simple grille et la dénomination des rues n'est pas explicite.

Le problème du routage compact consiste donc à mettre à jour des méthodes de routage requérant localement peu de connaissance, c'est-à-dire peu de bits mémoire, à stocker dans chacun des routeurs du réseau.

Cette thèse s'organise de la façon suivante :

Chapitre 2. Fonctions de routage

Dans ce court chapitre nous introduisons un modèle général de fonction de routage. La fonction de routage est le formalisme mathématique décrivant l'activité distribuée des routeurs. Nous donnons quelques exemples illustrant le modèle. C'est sur ce modèle que repose toute notre étude.

Chapitre 3. Routage par intervalles

Ce chapitre présente une méthode de routage appelée *routage par intervalles* : un routeur détermine la direction dans laquelle transmettre un message en déterminant un intervalle contenant l'adresse de destination du message. On suppose une correspondance entre intervalles et directions. Cette méthode de routage est séduisante, mais très peu de résultats sont connus sur les propriétés topologiques que doit satisfaire un réseau pour admettre une fonction de routage par intervalles réalisant des contraintes particulières : plus courts chemins, faible nombre d'intervalles associés à chaque direction *etc.* Nous caractérisons les réseaux qui admettent une fonction de routage par intervalles *linéaires* et *strictement linéaires*, avec un unique intervalle par direction.

Chapitre 4. Compacité maximale du routage par intervalles

Dans ce chapitre nous reprenons l'étude menée au chapitre précédent en imposant que le routage soit toujours selon des plus courts chemins. En effet, dans la pratique, il est souvent important de pouvoir garantir un certain temps de réponse pour les communications au travers du réseau. L'étude des routages de plus courts chemins est donc importante pour minimiser les temps de « latence ». De plus, des communications sur des plus courts chemins limitent l'engorgement provoqué par l'utilisation d'un trop grand nombre de ressources (liens de communication), et devrait permettre une meilleure bande passante globale du réseau. Nous décrivons de grandes classes de réseaux admettant une fonction de routage de plus courts chemins. Nous décrivons ensuite en terme d'intervalles les principales propriétés que satisfont les réseaux usuels utilisés pour la connexion des processeurs dans la plupart des ordinateurs parallèles à mémoires distribuées. Enfin, nous établissons des bornes inférieures asymptotiquement exactes sur le nombre d'intervalles requis par les fonctions de routage par intervalles sur des réseaux de n nœuds. Ces bornes montrent que le routage par intervalles est loin d'être une solution idéale.

Chapitre 5. Méthodes de routage universel

Dans ce chapitre, nous étudions les méthodes de routage universel, c'est-à-dire des méthodes de routage applicables à tous types de réseaux. Comme précédemment, nous ne considérons que des routages de plus courts chemins. Les derniers résultats du chapitre 4 tendent à montrer que le routage par intervalles n'est pas une méthode de routage universel compact. Nous nous proposons donc de répondre à la question suivante : existe-t-il une méthode de routage compact applicable à tous les réseaux ? Nous présentons les réponses proposées dans la littérature, et nous proposons une technique originale pour mettre en évidence une borne fondamentale sur la compressibilité de l'information de routage de toute méthode de routage universel. Ce résultat est étendu aux routages dont les longueurs de chemins peuvent s'éloigner au plus d'un facteur 2 des plus courtes possibles.

Chapitre 6. Incompressibilité des tables de routage

La limite de non-compressibilité établie au chapitre précédant n'est pas fine au sens où elle ne donne pas de solutions constructives pour une méthode de routage universel plus compact que la méthode standard de routage par tables. De plus, il peut être important de restreindre la classe des réseaux étudiés à celle des réseaux de degrés bornés. Nous établissons, de manière générale, qu'il n'existe pas de méthode de routage plus compact que les tables. Autrement dit nous démontrons qu'il existe une famille de réseaux $G_{n,d}$ de degré maximum d pour lesquels tout algorithme ou fonction de routage ne peut être codé localement en moins de $\Theta(n \log d)$ bits n représentant le nombre de nœuds de $G_{n,d}$. De plus cette limite locale est requise sur $\Theta(n)$ routeurs du réseau.

Ceci est le résultat fondamental de cette thèse. Il peut s'exprimer informellement de la façon suivante : les tables de routage (une entrée pour chaque destination donnant le lien de sortie correspondant) sont asymptotiquement incompressibles, et ce même si on se restreint à la classe des réseaux de degré maximum d fixé.

Chapitre 7. Perspectives

Ce chapitre contient un ensemble de propositions de nouveaux axes de recherches sur le problème du routage compact à la vue du résultat de non-compressibilité établi au chapitre précédent. Nous listons plusieurs questions non encore résolues : cas des réseaux planaires, des réseaux aléatoires, *etc.* Nous proposons ensuite de généraliser nos modèles de fonctions de routage au cas *adaptatif*. Finalement nous tentons de cerner les conséquences du résultat principal de cette thèse sur l'algorithmique distribuée en général.

CHAPITRE 2 Fonctions de routage

2.1 Modèles et définitions

Nous considérons le modèle standard des réseaux de communications point-à-point. Un réseau qui est décrit par un graphe orienté, symétrique, connexe et fini $G = (V, E)$. Dans la suite n dénote l'*ordre* de G, c'est-à-dire le nombre de sommets de G. Les sommets représentent les nœuds ou les routeurs du réseaux : nous supposerons qu'un routeur est associé à chaque processeur (le modèle pourrait être facilement étendu à un modèle qui supposerait plusieurs processeurs connectés au même routeur). Les arcs représentent les liens de communications entre les routeurs. Le graphe G étant supposé orienté symétrique, on le représentera le plus souvent sous la forme d'un graphe non orienté en gardent en mémoire le fait qu'à chaque arête correspond deux arcs opposés. Un lien spécial (qui n'est pas un arc de E) connecte, à l'intérieur de chaque nœuds, le processeur au routeur (voir la figure 2.1). Un sommet ne peut communiquer directement qu'avec ses voisins, et les messages entre des sommets non adjacents sont routés le long d'un chemin connectant ces sommets dans le réseau. Le routeur est composé d'une *interface d'entrée*, d'une *interface de sortie* et d'une *fonction locale de routage*. Si un message M arrive sur un port p du routeur avec une entête h, alors l'interface d'entrée envoie p et h vers la fonction locale de routage qui calcule un nouvel entête h' et un port de sortie p'. Le couple (p', h') est communiqué à l'interface de sortie qui, finalement, retransmet le message M avec l'entête h' sur le port p'.

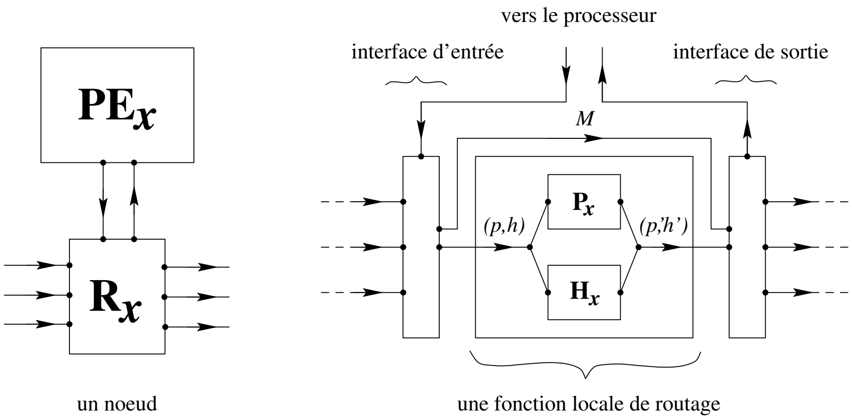


FIG. 2.1 – *Un modèle de routeur*

D'une certaine façon trouver une *fonction de routage* sur un graphe orienté symétrique

$G = (V, E)$ consiste à étiqueter les sommets et les arcs de G tel que la fonction locale de routage puisse calculer un chemin entre un couple source-destination quelconque de sommets. Sans perte de généralité nous pouvons supposer que chacune des étiquettes est un numéro entier, puisque les graphes considérés sont finis. Plus précisément une *numérotation des sommets* de G est une injection $\mathcal{L} : V \to \mathbb{N}$ qui associe à chaque sommet $x \in V$ un unique entier $\mathcal{L}(x) \in \mathbb{N}$. Dans toute la suite, nous noterons E_x l'ensemble des arcs de E incidents à x (en entrée ou en sortie), pour tout $x \in V$. Une *numérotation locale des arcs* ou encore une *numérotation des ports* d'un sommet $x \in V$ est une fonction $\mathcal{E}_x : E_x \to \mathbb{N}$ qui associe à chaque arc un entier tel que tous les arcs incidents à un même sommet ont des numéros différents, sauf les arcs symétrique qui ont le même numéro. La collection $\mathcal{E} = \{\mathcal{E}_x \mid x \in V\}$ est appelée une *numérotation des arcs* de G. Attention, cette numérotation des arcs n'est pas globale puisque les deux extrémités d'un arc induisent des numéros éventuellement différents. Sans perte de généralité nous supposons que le port spécial de connexion entre le processeur et le routeur est toujours numéroté 0 et que les entêtes sont également codés par des entiers naturels. Notez que nous aurions pu choisir n'importe quel autre ensemble dénombrable, comme celui des chaînes binaires par exemple. Nous donnons une définition formelle d'une fonction de routage qui est assez proche de celle donnée par Peleg et Upfal dans [67]. Toutefois la fonction d'initialisation n'est plus considérée dans notre modèle.

Définition 2.1 (fonction de routage) *Soit $G = (V, E)$ un graphe orienté symétrique. Une* fonction de routage *sur G est une fonction $R : V \times \mathbb{N} \times \mathbb{N} \to \mathbb{N} \times \mathbb{N}$ qui étant données une numérotation des sommets $\mathcal{L}$ et une numérotation des arcs $\mathcal{E} = \{\mathcal{E}_x \mid x \in V\}$ de G, satisfait pour tout couple de sommets (x, y) de G, il existe un entier $k \geqslant 1$ et :*

- *une suite $\{v_i \in V \mid 0 \leqslant i \leqslant k\}$ de $k+1$ sommets, formant un chemin de x à y dans G*
- *une suite $\{p_i \in \mathbb{N} \mid 0 \leqslant i \leqslant k\}$ de $k+1$ numéros de ports, numérotant les arcs entre deux sommets successifs du chemin*
- *une suite $\{h_i \in \mathbb{N} \mid 0 \leqslant i \leqslant k\}$ de $k+1$ entêtes, obtenus par modification successive de l'entête originale le long du chemin*

tels que :

- $v_0 = x$, $p_0 = 0$, $h_0 = \mathcal{L}(y)$
- $(p_i, h_i) = R(v_{i-1}, p_{i-1}, h_{i-1})$, *pour tout* $i \in \{1, \ldots, k-1\}$
- $v_k = v_{k-1} = y$, $p_k = 0$, $h_k = h_{k-1}$

Notation. La définition 2.1 implique que seulement les graphes fortement connexes supportent une fonction de routage car, implicitement, une fonction de routage sur un graphe impose l'existence d'un chemin entre toutes les paires de sommets du graphe. Dans toute la suite de cette thèse nous concentrerons notre attention sur les graphes orientés symétriques, finis, connexes et sans multi-arcs. Nous ferons toujours référence à de tels graphes par le simple terme « graphe ». Nous noterons les arcs par des couple de sommets (a, b) et les arêtes par des paires $\{a, b\}$.

Si R est une fonction de routage sur un graphe $G = (V, E)$, le schéma de construction des chemins de routage par R est le suivant : un message arrivant dans un routeur v sur le port p avec une entête h est routé vers le port p' avec l'entête h' tel que $(p', h') = R(v, p, h)$. Plus précisément, si le processeur connecté au routeur x veut envoyer un message M au processeur connecté au routeur y, il envoie le numéro de la destination $\mathcal{L}(y)$, sur son lien spécial 0. Le routeur x prépare une entête h et un numéro de port p en calculant le couple

$(p, h) = R(x, 0, \mathcal{L}(y))$. L'entête h est attaché au message M, puis les deux sont envoyés sur le port p. Supposons maintenant que M arrive sur un routeur intermédiaire z par le port p' avec un entête h'. La première étape du routage consiste à calculer le couple $(p'', h'') = R(z, p', h')$. Si $p'' = 0$, le processus de routage prend fin, *i.e.* le message M avec son entête $h' = h''$ est envoyé sur le lien spécial 0 de z. Sinon, z transmet sur son port p'' le message M avec un nouvel entête h''.

Notations. La représentation *distribuée* de R sur $G = (V, E)$ est la collection $\{R_x \mid x \in V\}$ telle que $R_x(p, h) = R(x, p, h)$, pour tout $x \in V$ et $(p, h) \in \mathbb{N}^2$. Pour tout $x \in V$, R_x est appelée la *fonction locale de routage* de R en x. C'est une fonction de $\mathbb{N}^2$ dans $\mathbb{N}^2$. De plus, R peut être décomposée comme un couple de fonctions (P, H), où P est la fonction de *ports* et où H est la fonction d'*entêtes*. P et H sont respectivement la première et la deuxième projection de R. De la même manière P_x et H_x dénotent la fonction *locale* de ports et la fonction *locale* d'entêtes en $x \in V$ (voir la figure 2.1). Elles sont définies par $P_x(p, h) = P(x, p, h)$ et $H_x(p, h) = H(x, p, h)$, pour tout $x \in V$ et $(p, h) \in \mathbb{N}^2$.

Dans ce modèle, nous considérons les fonctions de routage R telles que le chemin de x à y induit par R ne dépend que des sommets x et y. De telles fonctions sont aussi appelées des fonctions de routage *statiques* car elles sont équivalentes à la donnée d'un ensemble de chemins fixés à l'avance sur le graphe. Elles ne sont pas *adaptatives* dans le sens où, en fonction de l'entête et du port d'arrivée du message, le routeur ne peut pas adapter sa réponse en fonction d'autres paramètres, comme la charge du réseau par exemple. Cependant, la fonction locale d'entête H_x utilisée pour le calcul du nouvel entête dépend de l'entête courant et du port entrant du message. Il en est de même pour la fonction locale de ports P_x. Ceci permet d'employer des fonctions de routage dites par *approximations successives* ou par *tâtonnements*. Ceci veut dire qu'un chemin induit par R peut contenir des cycles avant d'atteindre sa destination. Cette propriété peut permettre de décentraliser la mémoire du routeur pour le stockage de l'information de routage car elle permet de visiter plusieurs nœuds avoisinant pour emmagasiner de l'information pouvant être transportée dans les entêtes. Cette possibilité a été efficacement utilisée par Peleg et Upfal dans [67].

Nous utilisons parfois une restriction du modèle général des fonctions de routage.

Définition 2.2 (fonction de routage simple) *Soit $G = (V, E)$ un graphe et R une fonction de routage sur un graphe G. R est une* fonction de routage simple *sur G s'il existe une fonction $R' : V \times \mathbb{N} \to \mathbb{N}$, telle que pour tout $x \in V$ et pour tout $(p, h) \in \mathbb{N}^2$, $R(x, p, h) = (R'(x, h), h)$.*

La définition 2.1 signifie que l'entête des messages contient seulement l'adresse de destination. Ceci implique en particulier que tous les chemins induits par des fonctions de routage simples ne contiennent pas de cycles. Une fonction de routage simple locale est donc entièrement déterminée par sa fonction de port et peut être modélisée par une fonction $R_x : V \to E_x \cup \{0\}$, pour tout $x \in V$.

Une fonction de routage R est dite de plus courts chemins si et seulement si tous les chemins induits par R sont de longueur minimum dans G.

2.2 Mémoire de routage

L'objectif de cette thèse est l'étude du codage des fonctions de routage afin de permettre une mise en œuvre « efficace » dans les routeurs. Nous allons surtout prendre en compte l'espace mémoire requis par les routeurs, ce qui donnera en particulier des bornes inférieures de la surface VLSI de ces routeurs.

Définition 2.3 (mémoire de routage d'un sommet) *Étant donné un graphe G, une fonction de routage R sur G et un sommet x de G, la* mémoire de routage *de x, notée* $\mathrm{MEM}(G,R,x)$*, est le nombre minimum de bits nécessaire pour « stocker » la fonction locale de routage R_x.*

Remarque. La mémoire de routage d'un sommet x est l'information nécessaire pour définir R_x, ou encore la taille minimum de mémoire ROM (de l'anglais *Read Only Memory*) pour décrire R_x. Cela peut être formellement définie comme la complexité de Kolmogorov [63] de la fonction R_x, défini comme la taille du plus petit programme qui retourne R_x. Tout comme la complexité de Kolmogorov, la mémoire de routage est bien évidemment définie pour un langage fixé. Bien sûr, le calcul proprement dit de la fonction R_x peut utiliser de la mémoire RAM (pour *Random-Access Memory*) qui n'est pas pris en compte ici. Le but de cette étude est de donner une limite inférieure absolue à la taille physique des routeurs. La mémoire de routage est une borne inférieure de l'espace mémoire et donc sur la surface physique requise par un routeur.

Clairement, en utilisant des astuces dans les numérotations, tout graphe G et tout sommet x de G supportent une fonction de routage R_x qui a une mémoire de routage de $O(1)$ bits. En effet les numérotations (des sommets et des arcs de G) ne sont pas prises en compte dans la mémoire de routage. L'étiquette de destination pourrait coder l'information complète du routage, incluant par exemple, en plus d'un identificateur particulier du sommet, la description complète de G par sa matrice d'adjacence. De même les numéros de ports de x pourraient contenir des éléments d'identification du sommet x lui-même de sorte que le calcul de la route d'un message arrivant sur un port p avec l'entête h soit réalisé par une fonction locale de routage R_x indépendante de x et de G, toutes les informations se trouvant dans les différentes étiquettes utilisées (entêtes et numéro de port). C'est pourquoi nous allons limiter la classe des fonctions de routage à celles qui utilisent des numérotations des sommets dans l'ensemble $\{1,\ldots,n\}$ et des numérotations locale d'arcs de x dans l'ensemble $\{1,\ldots,\deg(x)\}$, où $\deg(x)$ désigne le degré de x (le nombre d'arêtes connectées à x).

Sous cette hypothèse, considérons le graphe complet K_n d'ordre n. En général, dans un sommet x, une fonction locale de routage R_x sur K_n peut être stockée avec $\Theta(n\log n)$ bits pour une numérotation aléatoire des ports ou pour une numérotation des ports choisie par un adversaire. En effet, un adversaire peut choisir une permutation π sur les $n-1$ numéros de ports et les $n-1$ voisins de x de telle manière qu'atteindre un voisin de x implique la pleine connaissance de π, *i.e.* $\lceil\log_2((n-1)!)\rceil = \Theta(n\log n)$ bits puisqu'il existe $(n-1)!$ permutations possibles. Cependant certaines fonctions de routage sur K_n peuvent être stockées en x d'une manière plus compacte en utilisant au plus $O(\log n)$ bits par routeur. Par exemple les sommets peuvent être numérotés par les entiers de $\{1,\ldots,n\}$ et les ports par des entiers de $\{1,\ldots,n-1\}$ tels que le port connectant les sommets x et y est : y si $x>y$, et $y-1$ si $y>x$. Pour simplifier nous confondons le sommet x avec son numéro. Si les entêtes sont les numéros des destinations, la fonction de port P'_x du sommet x pourrait effectuer l'algorithme suivant : si la destination est x, alors le message est arrivé, sinon envoyer le message sur le port qui correspond à sa destination. Il peut être formalisé par :

$P'_x(p,y) =$ « *si* $y = x$ *alors retourner* 0
sinon si $y < x$ *alors retourner* y
sinon retourner $y - 1$ ».

Clairement, cette fonction définit une fonction de routage de plus courts chemins sur K_n et cet algorithme peut être décrit avec seulement $O(\log n)$ bits (pour stocker la valeur x). En général, nous sommes intéressés de trouver la « meilleure » fonction de routage sur un graphe donné. Cette considération mène à la définition ci-dessous de la mémoire de routage d'un graphe. Dans un souci de clarté, nous restreignons provisoirement cette définition aux cas particuliers des fonctions de routage de plus courts chemins. Nous verrons au chapitre 5 une définition plus générale de la mémoire de routage incluant les notions de « facteur d'élongation » et de « classe » de fonctions de routage.

Définition 2.4 (mémoire de routage d'un graphe) *Étant donné un graphe* $G = (V,E)$*, la* mémoire de routage du graphe G *est notée* $\mathrm{MEM}_{\mathit{locale}}(G)$ *et est définie par :*

$$\mathrm{MEM}_{\mathit{locale}}(G) = \min_R \max_{x \in V} \mathrm{MEM}(G,R,x)$$

où la minimisation s'effectue sur toutes les fonctions de routage de plus courts chemins sur G*, pour lesquelles les sommets sont numérotés dans* $\{1,\ldots,|V|\}$ *et les numéros de ports de tout* $x \in V$ *sont pris dans* $\{1,\ldots,\deg(x)\}$.

Le problème du routage compact consiste à trouver une *méthode de routage* qui, pour tout graphe G, retourne une fonction de routage sur G qui nécessite localement le nombre minimum de bits pour être stockée. Par exemple la mémoire de routage du graphe H_n, l'hypercube d'ordre n, est $O(\log n)$ bits en utilisant un routage classique par dimensions. Il est connu que tout arbre [75], tout graphe planaire-externe [30] et tout graphe arc-circulaire unitaire [25] d'ordre n et de degré maximum d, a une mémoire de routage de $O(d \log n)$ bits car la méthode du routage par intervalles avec un intervalle par lien s'applique à ces réseaux. Le routage par intervalles sera étudié au chapitre 3.

Nous nous intéressons à la mémoire *locale* de routage. Encore une fois, nous insistons sur les espaces autorisés des numérotations : nous invitons le lecteur à vérifier que si l'on avait accepté une numérotation des arcs de x dans l'ensemble $\{1,\ldots,\deg(x)+1\}$, alors on aurait obtenu pour le graphe complet $\mathrm{MEM}_{\mathit{locale}}(K_n) = O(1)$ car ce ne serait plus la peine de stocker le nom du sommet pour router (chaque port portant un numéro identique au numéro du sommet qu'il connecte).

Certains auteurs ont considéré la somme de l'information de routage distribuée sur tout le réseau. En effet, la mémoire de routage n'est pas toujours équilibrée entre les nœuds d'un même réseau.

Définition 2.5 (mémoire globale de routage d'un graphe) *Étant donné un graphe* $G = (V,E)$*, la* mémoire globale de routage du graphe G *est notée* $\mathrm{MEM}_{\mathit{globale}}(G)$ *et est définie par :*

$$\mathrm{MEM}_{\mathit{globale}}(G) = \min_R \sum_{x \in V} \mathrm{MEM}(G,R,x)$$

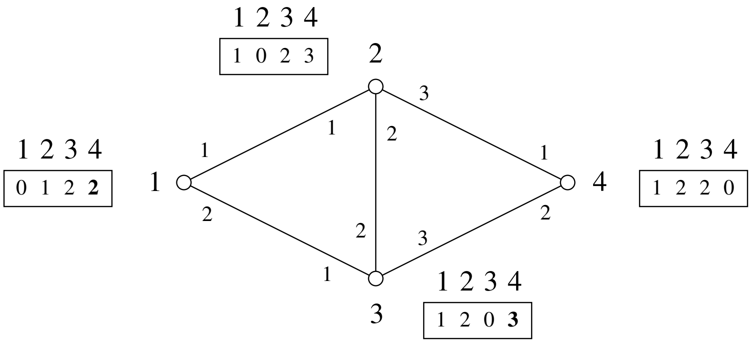
où la minimisation s'effectue sur toutes les fonctions de routage de plus courts chemins sur G, pour lesquelles les sommets sont numérotés dans $\{1,\dots,|V|\}$ et les numéros de ports de tout $x \in V$ sont pris dans $\{1,\dots,\deg(x)\}$.

Par exemple, les arbres et les graphes planaires-externes requièrent une mémoire globale de routage de $O(n \log n)$ bits, car ils possèdent $O(n)$ arcs au total et supportent un routage par intervalles avec au plus un intervalle par arc. Puisque notre approche est essentiellement locale la mémoire de routage sera toujours supposée être, par défaut, la mémoire *locale* de routage. Notez que ces définitions n'imposent pas de limitation quant à la longueur des entêtes.

2.3 Tables et algorithmes de routage

Les algorithmes de routage de nombreux réseaux sont mis en œuvre de façon matérielle sur le routeur. C'est le cas du routage XY [11] sur la grille et du routage e-cube [12] sur l'hypercube, réalisés tous deux par comparaison ou calcul d'un XOR entre l'adresse de destination du message et l'adresse locale du routeur. Cependant, les réseaux n'ont pas toujours la structure simple des grilles ou des hypercubes utilisés surtout pour connecter des processeurs au sein d'un ordinateur parallèle. Aussi, certains réseaux peuvent même ne pas avoir de structure du tout, si bien qu'aucun algorithme ne peut clairement être défini. Dans ce cas, des tables de routage sont utilisées : quand un message arrive sur un nœud, le routeur consulte une table locale et détermine le port de sortie du message. Cette méthode est rapide et permet l'implantation de n'importe quelle fonction de routage simple sur n'importe quel réseau.

La figure 2.2 représente une table de routage pour un graphe de 4 sommets. Le graphe est représenté avec une numérotation de ses sommets et de ses arcs. Par exemple, le chemin d'un message allant du sommet 1 au sommet 4 utilise le port numéroté 2, puis le port numéroté 3, en passant par le sommet 3. Chaque ligne de la table est stockée sur le routeur correspondant.



↗	1	2	3	4
1	0	1	2	**2**
2	1	0	2	3
3	1	2	0	**3**
4	1	2	2	0

FIG. 2.2 – *Table de routage.*

En ce qui concerne la mémoire de routage nous avons donc :

Proposition 2.1 *Soit G un graphe d'ordre n de degré maximum d. Pour toute fonction de routage simple R sur G et pour tout sommet x de G,* $\mathrm{MEM}(G,R,x) = O(n \log d)$.

La proposition 2.1 implique en particulier que, pour tout graphe G, $\mathrm{MEM}_{\mathit{globale}}(G) = O(n^2 \log d)$ bits.

Contrairement aux tables, nous avons vu dans le cas des grilles, de l'hypercube ou du graphe complet, les algorithmes de routages peuvent fournir une mise en œuvre très compacte d'une fonction de routage sur un graphe donné. Dans certains cas, la connaissance de l'adresse locale du processeur n'est même pas nécessaire au routeur, pouvant ainsi avoir une mémoire locale de routage constante. Cette possibilité est interdite dans le modèle de Peleg et d'Upfal dans [67], qui impose dans leur définition l'adresse locale comme nécessaire à la mémoire de routage. Bien sûr, il est raisonnable de supposer que chaque processeur possède une mémoire suffisante pour stocker sa propre adresse, par exemple pour éviter l'envoi de messages à lui-même. Il est cependant possible que certaines fonctions de routage R_x n'utilisent pas le numéro de x comme le suggère l'exemple suivant. Soit $K_{m,n-m}$ le graphe bi-parti complet de n sommets où m est une constante. Nous allons définir une fonction de routage simple de plus courts chemins définie par la fonction de port P_x. Les sommets de la première partition sont numérotés de 1 à m, et ceux de la seconde sont numérotés de $m+1$ à n. Le port entre x et y est numéroté $y-m$ si $x<y$ et y sinon. Dans ce cas les fonctions locales de port peuvent être décrites comme suit : pour tout port p, pour toute destination y et pour tout routeur x de la première partition, $x \in \{1,\ldots,m\}$:

$$P_x(p,y) = « \textit{si } y = x \textit{ alors retourner } 0 \quad \textit{sinon retourner } \max\{y-m, 1\} »$$

et pour tout routeur x de la seconde partition,

$$P_x(p,y) = « \textit{si } y \leqslant m \textit{ alors retourner } y \quad \textit{sinon si } p = 0 \textit{ alors retourner } 1 \quad \textit{sinon retourner } 0 ».$$

L'algorithme de routage ci-dessus définit une fonction de routage simple et de plus courts chemins. De plus, il peut être décrit avec $O(1)$ bits puisque m et x, pour tout $x \in \{1,\ldots,m\}$, peuvent être codés sur $O(1)$ bits. Notons cependant que si les processeurs de la seconde partition s'envoient des messages à eux-mêmes alors les chemins construits sont de longueur 2.

En fait, si m n'est pas constant, il est facile de montrer, en utilisant le même algorithme, que $\mathrm{MEM}_{\mathit{locale}}(K_{m,n-m}) = O(\log n)$ pour tout $0 < m < n$.

Remarque. En autorisant le processeur à s'envoyer un message à lui même par l'intermédiaire du routeur, et en imposant que le calcul de la fonction locale de routage de celui-ci réponde 0, alors cela revient à dire que le routeur « connaît » son adresse locale. Dans ces hypothèses la mémoire globale de routage est, pour tout graphe, au moins $\Omega(n \log n)$ bits.

CHAPITRE 3

Routage par intervalles

3.1 Introduction

Nous avons vu au chapitre précédent que les fonctions de routages sont mises en œuvres localement dans chaque routeur du réseau. Les tables de routages sont une méthode de codage applicable à tout réseau de n nœuds et de degré maximum d. Nous avons vu que la mémoire de routage d'une telle méthode nécessite $O(n \log d)$ bits pour chaque nœud, et donc $O(n^2 \log d)$ bits d'espace mémoire globale, juste pour le routage.

Pour réduire la taille des tables de routages, Khatib et Santoro ont introduit dans [75] une méthode originale pour coder ces tables. Les adresses de destinations empruntant le même port de sortie du routeur sont regroupées en intervalles. Un routeur trouve la direction sur laquelle transmettre un message en déterminant l'intervalle contenant l'adresse de destination du message, chaque intervalle étant associé à une direction particulière. Cette méthode de mise en œuvre des fonctions de routage est très attirante mais très peu de propriétés sont connues à propos de la topologie que doit satisfaire un réseau pour admettre une fonction de routage par intervalles. Notre approche, dans ce chapitre, à pour sujet la caractérisation des réseaux supportant certains types de fonctions de routage par intervalles, en particulier celles ne nécessitant qu'un seul intervalle par direction.

La section 3.2 définit formellement le routage par intervalles en soulignant les paramètres liés à cette notion : nombre d'intervalles (compacité) et types d'intervalles (linéaires, cycliques, stricts). La section 3.3 décrit brièvement l'état de l'art sur le routage par intervalles. Dans la section 3.4 nous caractérisons les graphes supportant un routage par intervalles linéaires. Nous étudions ensuite en section 3.5 l'efficacité d'une fonction de routage par intervalles en terme de longueur des chemins et de compacité. Nous nous intéressons aux fonctions de routage par intervalles de plus courts chemins, ainsi qu'à la longueur des chemins générés pour une compacité fixée. Dans la section 3.6 nous appliquons notre étude à de nombreux réseaux d'interconnexions « usuels » (grille, hypercube, *Shuffle-Exchange*, *etc*).

Une grande partie des résultats présentés dans ce chapitre ont été élaborés en collaboration avec Pierre Fraigniaud, et ont donné lieu à plusieurs publications internationales et rapports de recherches soumis à des revues internationales (se reporter à [23, 24, 25, 37]).

3.2 Définitions

Un *intervalle* $[a, b]$ de l'ensemble $\{1, \ldots, n\}$ est l'ensemble des entiers $\{a, \ldots, b\}$ numérotés consécutivement de manière cyclique. Par exemple, l'intervalle $[7, 2]$ de $\{1, \ldots, 9\}$ est le l'ensemble $\{7, 8, 9, 1, 2\}$. Si $a \leqslant b$, on dira que $[a, b]$ est un intervalle *linéaire*, et sinon que $[a, b]$ est un intervalle *cyclique*.

La définition formelle du routage par intervalles est la suivante :

Définition 3.1 (routage par intervalles) *Étant donné un graphe $G = (V, E)$ d'ordre n, une fonction de* routage par intervalles *sur G est une fonction de routage simple $R = \{R_x : V \to E_x \cup \{0\} \mid x \in V\}$ telle qu'il existe :*
- *une numérotation des sommets, $\mathcal{L} : V \to \{1, \ldots, n\}$, et*
- *une famille $\mathcal{I} = \{I_e \mid e \in E\}$, où I_e est un sous-ensemble de $\{1, \ldots, n\}$ associé à e chacun codé par un ensemble d'intervalles $[a_i, b_i]$, $i \in \{1, \ldots, k\}$, tel que $I_e = [a_1, b_1] \cup \cdots \cup [a_k, b_k]$*

satisfaisant les propriétés suivantes :
- *pour tout $x \in V$, $\mathcal{L}(x) \cup \{I_{(x,y)} \mid (x, y) \in E\} = \{1, \ldots, n\}$ (union)*
- *pour tout arc (x, y) et (x, z) de E, $I_{(x,y)} \cap I_{(x,z)} = \emptyset$ (disjonction)*
- *pour tout couple $(x, y) \in V \times V$:*
  - *si $x = y$ alors $R_x(y) = 0$*
  - *sinon, il existe $(x, z) \in E$ tel que $\mathcal{L}(y) \in I_{(x,z)}$ et $R_x(y) = (x, z)$*

Une fonction de routage par intervalles est notée par un couple $(\mathcal{L}, \mathcal{I})$ vérifiant la définition 3.1. Si une fonction de routage par intervalles $R = (\mathcal{L}, \mathcal{I})$ est définie sur le graphe G, alors le routage des messages est réalisé comme suit : à la réception d'un message, un routeur x compare l'entête $\mathcal{L}(y)$ du message avec son propre numéro $\mathcal{L}(x)$ pour vérifier si le message est arrivé à destination. Si ce n'est pas le cas, le message et son entête sont transmis par l'unique arc (x, z) tel que $\mathcal{L}(y) \in I_{(x,z)}$.

Considérons la figure 3.1 comme exemple d'une fonction de routage par intervalles $R = (\mathcal{L}, \mathcal{I})$. Dans cet exemple, la numérotation $\mathcal{L}$ des sommets associe aux sommets a, b, e, g, d, f, c les entiers $1, 2, \ldots, 7$ respectivement. Pour tout arc (x, y) du graphe — dessiné avec des arêtes représentant ses arcs symétriques — l'ensemble $I_{(x,y)}$ des numéros de destination associé à (x, y) est codé en intervalles notés à coté du sommet x. Par exemple, les intervalles $[1, 2]$ et $[5]$ proches du sommet c numéroté 7 correspondent à l'ensemble d'entiers $\{1, 2, 5\}$ et forment l'ensemble $I_{(c,a)}$ qui est associé à l'arc (c, a). En conséquence, si le sommet b veut envoyer un message au sommet f en utilisant la fonction de routage R, le message prendra dans l'ordre les arcs : (b, a), (a, c) et (c, f).

Notez que, dans ce modèle, le numéro d'un sommet peut ne pas appartenir à l'un des intervalles associés à ses arcs (sur la figure 3.1, le sommet numéroté 5 appartient à l'un de ses intervalles, ce qui n'est pas le cas du sommet numéroté 3). La conception physique des routeurs dépend néanmoins de ce type de particularités. En effet, si l'adresse du routeur appartient à l'un de ses intervalles, il devra mettre en œuvre un pré-calcul de vérification de la destination des messages reçus avant d'utiliser ses intervalles. De même, les comparaisons utilisées dans le routeur devront être mises en œuvre de façon différentes selon que les intervalles soient linéaires ou parfois cycliques. De plus, comme nous le verrons plus tard, ces distinctions sur le type d'intervalles se révéleront être primordiales pour l'étude des réseaux obtenus par produit

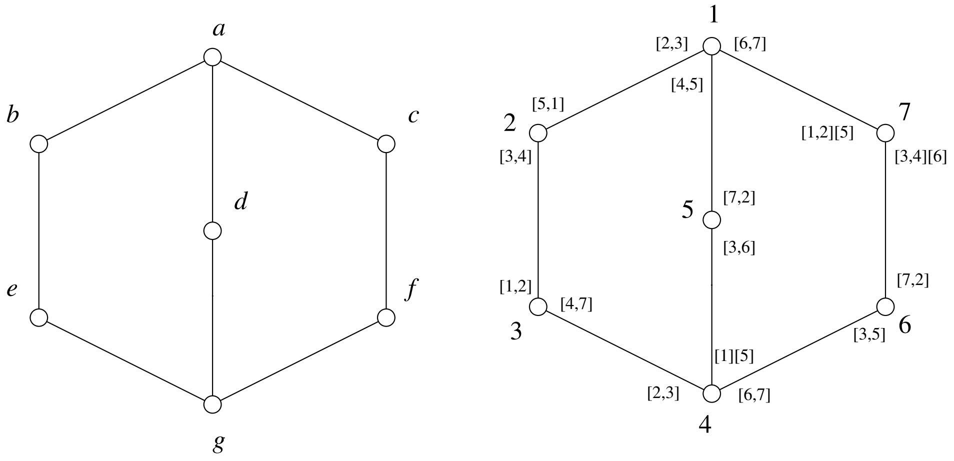


FIG. 3.1 – *Un graphe G et une fonction de routage par intervalles sur G.*

cartésien.

Définition 3.2 (fonction de routage par intervalles stricts) *Soit $R = (\mathcal{L}, \mathcal{I})$ une fonction de routage par intervalles sur $G = (V, E)$. R est une* fonction de routage par intervalles stricts *si pour tout $x \in V$ et pour tout $(x, y) \in E$, $\mathcal{L}(x) \notin I_{(x,y)}$.*

Puisque le routage par intervalles a été proposé pour réduire l'espace mémoire utilisé par les routeurs, nous nous sommes intéressés à la limitation du nombre d'intervalles nécessaire à la représentation de chaque ensemble de destinations. Idéalement, on souhaite avoir au plus un intervalle par arc sur chaque sommet.

Définition 3.3 (compacité d'une fonction de routage par intervalles)
Soit G un graphe d'ordre n et $R = (\mathcal{L}, \mathcal{I})$ une fonction de routage par intervalles sur G. Pour tout $I \subset \{1, \ldots, n\}$ on note $\alpha(I)$ (respectively $\beta(I)$) le plus petit nombre d'intervalles (respectively d'intervalles linéaires) dont l'union est I. La compacité *de R et la* compacité linéaire *de R sont respectively notées* Irs(R) *et* Lirs(R) *et sont définies par :*

$$\mathrm{Irs}(R) = \max_{I \in \mathcal{I}} \alpha(I) \quad \textit{et} \quad \mathrm{Lirs}(R) = \max_{I \in \mathcal{I}} \beta(I)$$

La compacité de la fonction de routage de l'exemple 3.1 est de compacité 2 à cause des arcs (c, a), (c, f) et (g, d). Sa compacité linéaire est également 2, à cause des arcs (b, a), (d, a) et (f, c). Notez que l'on aurait pu prendre comme critère le nombre total d'intervalles sur tous les arcs du graphe. Cependant ce paramètre ne permet pas de contrôler localement le nombre d'intervalles par routeur. De même, nous aurions pu concevoir une définition basée sur le nombre total d'intervalles par sommets. Cependant, comme nous le verrons dans la suite, la compacité telle que définie dans la définition 3.3 permet d'obtenir des résultats plus intéressants.

Dans la suite, nous dirons qu'une fonction de routage par intervalles est *linéaires* si sa compacité linéaire est 1.

3.3 Travaux antérieurs

La notion de routage par intervalles a été introduit par Khatib et Santoro dans [75]. Ils ont montré principalement que tout graphe acyclique supporte une fonction de routage par intervalles de plus courts chemins de compacité 1. De plus, si le graphe contient des cycles, ils ont montré qu'il existe une fonction de routage par intervalles de compacité 1 telle que la longueur maximum de la route entre deux sommets est au plus deux fois le diamètre. Van Leeuwen et Tan ont ensuite étudié dans [80] le problème du routage par intervalles pour les graphes non orientés. Ils ont montré que tout graphe supporte une fonction de routage par intervalles de compacité 1. Ils ont donné des exemples simples de graphes (arbres, graphe complet, cycles, *etc*) supportant des fonctions de routage par intervalles et de plus courts chemins. Ils ont aussi étudié la compacité des fonction de routage par intervalles de plus courts chemins sur le tore. Dans [5], Bakker, van Leeuwen et Tan ont introduit routage par intervalles linéaires. Ils ont caractérisé les arbres supportant des fonctions de routage par intervalles de plus courts chemins de compacité linéaire 1. Ils ont listé des exemples simples de graphes supportant ou non des fonctions de routage par intervalles linéaires de plus courts chemins. Ils ont montré notamment que l'hypercube et la grille à d dimensions supportent de telles fonctions de routage. Bakker, van Leeuwen et Tan ont également étudié le routage par intervalles linéaires avec des contraintes sur les coûts des communications entre les voisins. Ce modèle à été introduit par Frederickson et Janardan dans [30]. Un état de l'art du routage par intervalles avec des contraintes de coûts entre communications de voisin à voisin est présenté dans [36], puis repris plus tard par van Leeuwen et Tan dans un résumé de l'état de l'art du routage par intervalles dans [81].

Notons que ce travail a un intérêt pratique pour la construction des réseaux d'interconnexions des machines parallèles à mémoires distribuées. Par exemple le C104 [21, 65] est un coprocesseur dédié au routage conçu par la société Inmos pour l'interconnexion de processeurs T9000 [13, 47]. Son fonctionnement est basé sur le principe du routage par intervalles.

3.4 Caractérisation de la classe 1-Lirs

Dans cette section, nous allons caractériser les graphes supportant des fonction de routage par intervalles de compacité 1, selon que les intervalles considérés soient stricts et/ou linéaires.

Notations. Pour tout entier $k \geqslant 1$, on note k-Irs (respectively k-Lirs), l'ensemble des graphes supportant une fonction de routage par intervalles de compacité (resp de compacité linéaire) au plus k. De manière similaire, on note k-Irss (respectively k-Lirss) l'ensemble des graphes supportant une fonction de routage par intervalles stricts de compacité (respectively de compacité linéaire) au plus k.

Tout d'abord, une bonne nouvelle due à van Leeuwen et Tan [80]:

Théorème 3.1 [80] *Tout graphe appartient à* 1-Irss.

Preuve. Soit r un sommet de G et T un arbre couvrant de G, de racine r. Nous définissons $\mathcal{L}$ par une numérotation des sommets de T en profondeur d'abord à partir de la racine r avec $\mathcal{L}(r) = 1$. Pour tout sommet x de T, soit $l_x = \max \mathcal{L}(y)$ où le maximum est pris sur tous les sommets y appartenant au sous-arbre de T de racine x. À chaque arc (x, y), où y est un fils de

x dans T (s'il existe), on pose $I_{(x,y)} = [\mathcal{L}(y), l_y]$. À chaque arc (x, y), où y est le père de x dans T (s'il existe), on pose $I_{(x,y)} =]l_x, \mathcal{L}(x)[$ (pouvant être cyclique). Nous affectons l'intervalle vide aux deux extrémités de toutes les arêtes de G n'appartenant pas à T.

Clairement, une telle fonction de routage par intervalles est par intervalles stricts. □

Notez que les chemins construits par le routage défini dans la preuve du théorème 3.1 sont plongés dans un arbre. La longueur de ces chemins est donc nécessairement plus petite que deux fois la hauteur de l'arbre. Cependant ce routage n'est certainement pas de plus courts chemins. Ceci doit donc modérer la bonne nouvelle !

Caractérisons maintenant les graphes appartenant à 1-Lirs. Tout d'abord une simple conséquence du théorème 3.1.

Corollaire 3.1 *Tout graphe appartient à* 2-Lirss.

Preuve. D'après la définition 3.3, $\beta(I) \leqslant \alpha(I) + 1$ pour tout intervalles $I \subset \{1, \ldots, n\}$. Donc toute fonction de routage par intervalles de compacité k est de compacité linéaire $\leqslant k + 1$. □

Il existe cependant des graphes qui n'appartiennent pas à 1-Lirs. Par exemple, considérons le graphe de gauche de la figure 3.2, pris dans [5], que nous appellerons le graphe en Y. Supposons qu'il supporte une fonction de routage par intervalles linéaires. Il existe alors une branche telle que ni le sommet x situé sur le milieu de la branche, ni le sommet y situé sur la feuille de la même branche est numéroté 1 ou 7. Soit z le sommet central. Nécessairement $I_{(x,z)}$ doit contenir 1 et 7, donc $I_{(x,z)} = [1, 7]$ et y n'est pas atteignable depuis x : contradiction. Nous allons caractériser les graphes appartenant à 1-Lirs.

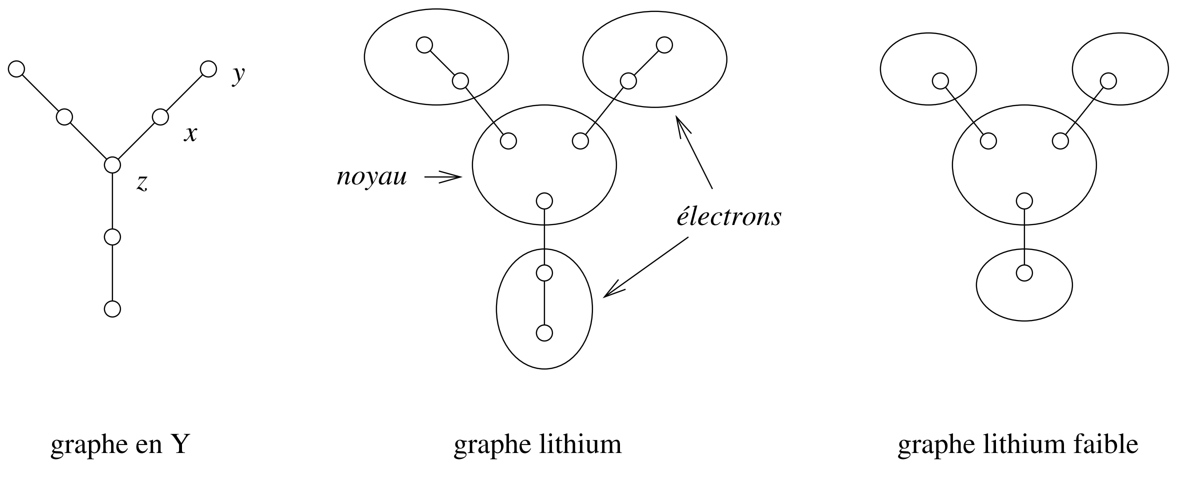


FIG. 3.2 – *Le graphe en Y, un « lithium » et un « lithium faible ».*

Nous rappelons qu'un isthme d'un graphe est une arête dont la suppression déconnecte le graphe. Dans la suite, nous appellerons un « isthme fort » une arête dont la suppression déconnecte le graphe en deux composantes connexes d'au moins deux sommets chacune.

Définition 3.4 (lithium) *Un* lithium *est a graphe possédant au moins trois isthmes forts incidents à une même composante connexe.*

La figure 3.2 représente la forme générale d'un lithium (le graphe en Y est un lithium). En accord avec la définition 3.4 et avec la figure 3.2, on appelle « noyau » d'un lithium une composante connexe possédant trois isthmes forts incidents, et « électrons » les trois autres composantes du graphes. Par le même genre d'arguments que pour le graphe en Y, on peut prouver facilement que :

Lemme 3.1 *Tout lithium n'appartient pas à* 1-Lirs.

En fait, nous avons :

Théorème 3.2 $G \in$ 1-Lirs $\Leftrightarrow$ *G n'est pas un lithium.*

Notons que puisque $K_2 \in$ 1-Lirs, nous ne considérerons dans la suite que des graphes ayant au moins 3 sommets. Nous allons montrer que tout graphe qui n'est pas un lithium appartient à 1-Lirs. Pour le faire, nous avons besoin de certains résultats préliminaires. On rappelle qu'un graphe est k-arête-connexe si entre toute paire de sommets il existe k chemins n'ayant aucune arête en commun.

Lemme 3.2 *Tout graphe $G = (V, E)$ 2-arête-connexe appartient à* 1-Lirss. *De plus, pour tout $x, y \in V$, il existe une fonction de routage par intervalles stricts $R = (\mathcal{L}, \mathcal{I})$ de compacité linéaire* 1 *et satisfaisant :*

i. $\mathcal{L}(x) = 1$
ii. $\forall z \in V, \mathcal{L}(z) < \mathcal{L}(y), \forall (z,u) \in E : \mathcal{L}(y) \in I_{(z,u)} \Rightarrow |V| \in I_{(z,u)}$
iii. *soit z le sommet tel que $\mathcal{L}(z) = |V|$, $\exists (z,u) \in E, \mathcal{L}(u) \leqslant \mathcal{L}(y)$ et $I_{(z,u)} = \emptyset$*

Preuve. Nous allons procéder de manière itérative : à chaque étape, on considère un sous-graphe $H = (V_H, E_H)$ de G contenant x et y et supportant une fonction de routage par intervalles linéaires satisfaisant les propriétés (i), (ii) et (iii). Nous mettons à jour successivement la construction en préservant les bonnes propriétés et en ajoutant un ou plusieurs sommets à H jusqu'à obtenir $|V_H| = |V|$.

Initialisation.

Si $x \neq y$ alors par le théorème de Menger, soient P_1 et P_2 deux chemins arête-disjoints de x à y. Il se peut que ces chemins aient un certain nombre de sommets communs distincts de x et de y. Notons $u_1, u_2, \ldots, u_{r-1}$ ces sommets sont rencontrés dans le même ordre en allant de x à y sur P_1 ou sur P_2. Soit alors $u_0 = x$, $u_r = y$ et, pour $i \in \{0, \ldots, r-1\}$, $\mathcal{C}_i$ le cycle composé d'une part du chemin P_1 de u_i à u_{i+1} et d'autre part du chemin P_2 de u_{i+1} à u_i.

Si $x = y$, alors soit $\mathcal{C}_0$ un cycle de G (d'au moins trois sommets) passant par x. Posons $u_0 = x = y = u_1$.

Soit H le sous-graphe de G composé des $\mathcal{C}_i$. Nous numérotons les sommets de H comme suit : on pose $\mathcal{L}(x) = 1$ et on numérote les sommets de $\mathcal{C}_0$ par ordre croissant et dans le même sens dit sens des aiguilles d'une montre (voir figure 3.3(a)–(b)). S'il existe plus d'un cycle (figure 3.3(c)), alors on poursuit à partir de u_1 et on numérote les cycles $\mathcal{C}_i$, $i \in \{2, \ldots, r-1\}$, par ordre croissant dans le même sens dit sens des aiguilles d'une montre, jusqu'à ce que tous les sommets des cycles soient numérotés.

Maintenant, nous affectons les intervalles comme suit (voir figure 3.3(d)). Soit $n_H = |V_H|$. Soit n_i le nombre de sommets du cycle $\mathcal{C}_i$, $0 \leqslant i < r$. Considérons un cycle $\mathcal{C}_i$, $0 \leqslant i < r$

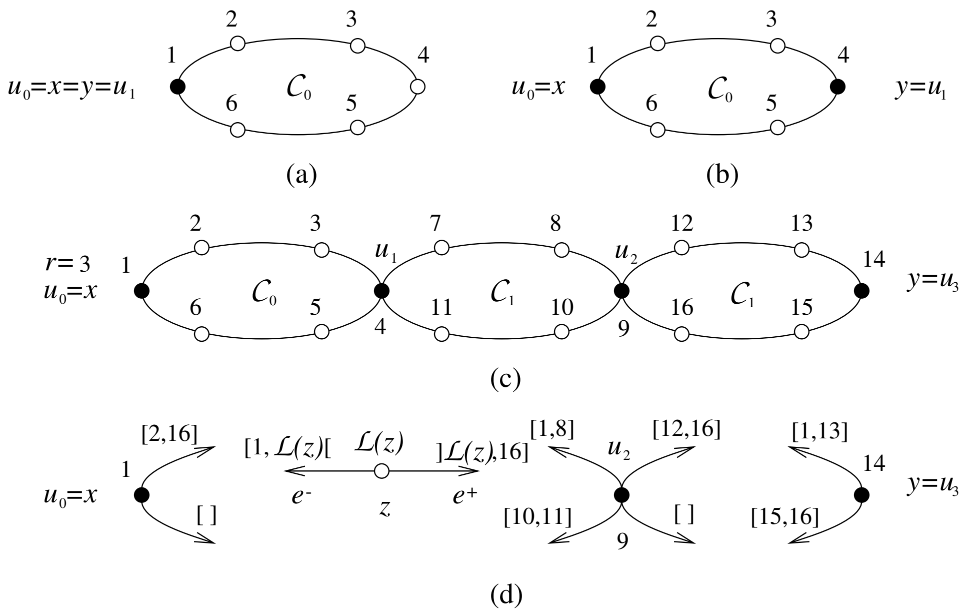


FIG. 3.3 – *Numérotation et affectation des intervalles pour la preuve du lemme 3.2.*

et un sommet z de $\mathcal{C}_i$. Soit e^+ (respectively e^-) l'arc incident à z orienté suivant le sens (respectively sens contraire) des aiguilles d'une montre de $\mathcal{C}_i$. Nous posons :

$$I_{e^+} = \begin{cases}]\mathcal{L}(z), n_H] & \text{si } z \neq u_{i+1} \text{ ou } u_{i+1} = y \\ ]\mathcal{L}(z), \sum_{j=0}^{i} n_j - i] & \text{si } z = u_{i+1} \text{ et } u_{i+1} \neq y \end{cases} \quad \text{et} \quad I_{e^-} = \begin{cases} [1, \mathcal{L}(z)[& \text{si } z \neq u_i \\ \emptyset & \text{si } z = u_i \end{cases}$$

On peut facilement vérifier que cette numérotation et cette affectation des intervalles construisent une fonction de routage par intervalles linéaires stricts sur H. Concernant les trois propriétés du lemme 3.2, (i) est clairement satisfaite ($\mathcal{L}(x) = 1$). D'autrement, si $\mathcal{L}(z) < \mathcal{L}(y)$ alors $\mathcal{L}(y)$ peut appartenir seulement à I_{e^+}. Si $I_{e^+} =]\mathcal{L}(z), n_H]$ alors (ii) est satisfaite, et si $I_{e^+} =]\mathcal{L}(z), \sum_{j=0}^{i} n_j - i]$ alors $\mathcal{L}(y)$ ne peut pas appartenir à I_{e^+} et (ii) est satisfaite également. Finalement, d'après la définition de I_{e^+}, si z est un sommet numéroté n_H, $I_{e^+} =]n_H, \sum_{j=0}^{r-1} n_j - r + 1] =]n_H, n_H] = \emptyset$, et donc (iii) est satisfaite.

Mise à jour.

Soit H un sous-graphe de G de $n_H < |V|$ sommets et contenant les sommets x et y. Soit $R = (\mathcal{L}, \mathcal{I})$ une fonction de routage par intervalles linéaires stricts sur H satisfaisant les conditions (i), (ii) et (iii). Il existe un chemin $P = (v_0, v_1, \ldots, v_k, v_{k+1})$, $k \geqslant 1$, dans G tel que $v_0 \in V_H$, $v_{k+1} \in V_H$ et $v_i \notin V_H$, $1 \leqslant i \leqslant k$. Considérons $H' = H \cup P$ et supposons $\mathcal{L}(v_0) < \mathcal{L}(v_{k+1})$. Considérons la numérotation suivante $\mathcal{L}'$ des sommets de H' : pour $v \in V_H$, si $\mathcal{L}(v) \leqslant \mathcal{L}(v_0)$ alors $\mathcal{L}'(v) = \mathcal{L}(v)$, sinon $\mathcal{L}'(v) = \mathcal{L}(v) + k$. Pour $i \in \{1, \ldots, k\}$, $\mathcal{L}'(v_i) = \mathcal{L}(v_0) + i$.

Nous mettons à jour les intervalles comme suit : soit $v \in V_H$, et $e \in E_H$ un arc incident à v. Supposons $I_e = [a, b]$, nous posons

$$I'_e = \begin{cases} [a, b] & \text{si } b < \mathcal{L}(v_0) \\ [a, b + k] & \text{si } a \leqslant \mathcal{L}(v_0) \leqslant b \\ [a + k, b + k] & \text{si } \mathcal{L}(v_0) < a \end{cases}$$

Nous devons maintenant fixer les intervalles sur les arcs du chemin P. Soit e un arc de P incident à v_0, nous fixons $I'_e = [\mathcal{L}'(v_1), \mathcal{L}'(v_k)]$. Soit e un arc de P d'extrémité v_{k+1}, nous fixons $I'_e = \emptyset$. Pour tout $i \in \{1, \ldots, k\}$, nous fixons $I'_{(v_i,v_{i+1})} = [\mathcal{L}'(v_{i+1}), n_H + k]$ et $I'_{(v_i,v_{i-1})} = [1, \mathcal{L}'(v_{i-1})]$. Il est facile de vérifier que $R' = (\mathcal{L}', \mathcal{I}')$ est une fonction de routage par intervalles linéaires stricts sur H'. Les propriétés (i) et (iii) sont bien sûr encore satisfaites. Supposons $v \in V_H$, $e \in E_H$ et $I_e = [a, b]$. Si $\mathcal{L}'(y) \in I'_e$ alors $b \geqslant \mathcal{L}(v_0)$ car sinon $\mathcal{L}'(y)$ serait strictement plus petit que $\mathcal{L}(v_0)$, on aurait alors $\mathcal{L}'(y) = \mathcal{L}(y)$ et $b = |V|$, ce qui est impossible si $b < \mathcal{L}(v_0)$. Il y a deux cas : soit $\mathcal{L}(y) \leqslant \mathcal{L}(v_0)$, soit $\mathcal{L}(y) > \mathcal{L}(v_0)$. Dans les deux cas, si $\mathcal{L}'(y) \in I'_e$ alors $\mathcal{L}(y) \in I_e$ et $b = n_H$, ce qui implique $n_H + k \in I'_e$. Il est tout aussi facile de vérifier que (ii) est réalisée sur les intervalles de P. Donc R' satisfait la propriété (ii) sur H'.

On pose $H = H'$ et on répète le procédé jusqu'à $n_H = |V|$. □

Lemme 3.3 *Soit $G \in$ 1-Lirss. Soit H un graphe obtenu en ajoutant à G un ensemble de sommets S tel que pour tout $x \in S$, x est de degré 1 et connecté à un sommet de G. Alors $H \in$ 1-Lirs.*

Preuve. Soit $R = (\mathcal{L}, \mathcal{I})$ réalisant $G = (V, E) \in$ 1-Lirss. Pour tout $x \in V$, on dénote $\nu(x)$ le nombre de sommets de S qui sont connectés à x dans $H = (V', E')$, $V' = V \cup S$. Nous définissons $R' = (\mathcal{L}', \mathcal{I}')$ sur H comme suit : pour tout $x \in V$, $\mathcal{L}'(x) = \mathcal{L}(x) + \sum_{y \in V | \mathcal{L}(y) < \mathcal{L}(x)} \nu(y)$. Si x est connecté dans H à p sommets $s_1, \ldots, s_p$ de S alors $\mathcal{L}'(s_i) = \mathcal{L}'(x) + i$ pour tout $i \in \{1, \ldots, p\}$. Un intervalle $[\mathcal{L}(u), \mathcal{L}(v)] \in \mathcal{I}$ est transformé en $[\mathcal{L}'(u), \mathcal{L}'(v) + \nu(v)] \in \mathcal{I}'$ et nous fixons les nouveaux intervalles : $I'_{(x,s_i)} = [\mathcal{L}'(s_i)]$ et $I'_{(s_i,x)} = [1, |V'|]$. Il est facile de vérifier que $R' = (\mathcal{L}', \mathcal{I}')$ est une fonction de routage par intervalles de compacité linéaire 1 sur H'. □

Preuve du théorème 3.2.

Soit $G = (V, E)$ un graphe qui n'est pas un lithium. Décomposons G, en supprimant tous ses isthmes forts, en un nombre maximum de composantes connexes $G_0 = (V_0, E_0), \ldots, G_k = (V_k, E_k)$, où $k \geqslant 0$ est le nombre total d'isthmes forts de G. Notez que certaines composantes peuvent ne contenir qu'un seul sommet.

Puisque G n'est pas un lithium, chaque composante G_i est connectée à au plus deux autres composantes. Par conséquent, nous pouvons supposer que G est un « chemin » de graphe G_i de la forme $G_0 - G_1 - \cdots - G_k$. Soit $x_i \in V_i$ et $y_{i-1} \in V_{i-1}$ les sommets tels que les isthmes forts soient les arêtes $\{y_{i-1}, x_i\}$, $i \in \{1, \ldots, k\}$. Définissons aussi $x_0 = y_0$ et $y_k = x_k$.

Pour tout $i \in \{0, \ldots, k\}$, soit $G'_i = (V'_i, E'_i)$ le graphe obtenu à partir de G_i en supprimant tous les sommets de degré 1 dans G, et soit $G' = \cup_{i=0}^{k} G'_i$. D'après le lemme 3.2, chaque $G'_i \in$ 1-Lirss et, plus précisément, on peut trouver une fonction de routage par intervalles linéaires stricts $R_i = (\mathcal{L}_i, \mathcal{I}_i)$ sur G'_i, pour chaque $i \in \{0, \ldots, k\}$, telle que :

i. $\mathcal{L}_i(x_i) = 1$
ii. $\forall z \in V'_i, \mathcal{L}_i(z) < \mathcal{L}_i(y_i), \forall (z, u) \in E'_i : \mathcal{L}_i(y_i) \in I_{(z,u)} \in \mathcal{I}_i \Rightarrow |V'_i| \in I_{(z,u)}$
iii. soit z le sommet tel que $\mathcal{L}_i(z) = |V'_i|$, $\exists (z, u) \in E'_i, \mathcal{L}_i(u) \leqslant \mathcal{L}_i(y_i)$ et $I_{(z,u)} = \emptyset$.

D'après les fonctions de routage R_i sur G'_i, $i \in \{0, \ldots, k\}$, on peut définir une fonction de routage par intervalles linéaires stricts $R = (\mathcal{L}, \mathcal{I})$ sur G' comme suit. Soit z un sommet quelconque de G' et soit i tel que $z \in V'_i$. On définit $\mathcal{L}(z)$ par un simple décalage de $\mathcal{L}_i(z)$: $\mathcal{L}(z) = \mathcal{L}_i(z) + \sum_{j=0}^{i-1} |V'_j|$. Les intervalles de $\mathcal{I}_i$ sont décalés en ajoutant $\sum_{j=0}^{i-1} |V'_j|$ aux deux extrémités de chaque intervalles excepté dans les cas suivants : si une des extrémités est 1, alors cette extrémité reste inchangée ; si une des extrémités est $|V'_i|$, nous la fixons à $|V'|$. Nous remplaçons également les intervalles vides $I_{(z,u)}$ définis par la propriété (iii) par $I_{(z,u)} =]\mathcal{L}(z), |V'|]$. Finalement, nous posons $I_{(x_i,y_{i-1})} = [1, \mathcal{L}(x_i)[$ et $I_{(y_{i-1},x_i)} =]\sum_{j=0}^{i-1} |V'_j|, |V'|]$, pour tout $i \in \{1, \ldots, k\}$.

Avec cette numérotation et ces intervalles, la propriété (i) assure que la route d'un sommet dans G'_i vers un sommets de G'_j, $j < i$, traverse x_i et quitte G'_i par un isthme fort (x_i, y_{i-1}), puis traverse x_{i-1} et quitte G'_{i-1} par un isthme fort (x_{i-1}, y_{i-2}), *etc.* La propriété (ii) assure que la route d'un sommet z de G'_i, avec $\mathcal{L}(z) \leqslant \mathcal{L}(y_i)$, vers un sommet de G'_j, $j > i$, traverse y_i et quitte G'_i par un isthme fort (y_i, x_{i+1}). La propriété (iii) assure que la route d'un sommet z de G'_i, avec $\mathcal{L}(z) > \mathcal{L}(y_i)$, vers un sommet de G'_j, $j > i$, traverse le sommet de numéro le plus élevé de G'_i, atteint un sommet de G'_i avec un numéro plus petit que y_i, puis traverse y_i et quitte G'_i par un isthme fort (y_i, x_{i+1}). Cette route passe ensuite par y_{i+1} et quitte G'_{i+1} par un isthme fort (y_{i+1}, x_{i+2}), *etc.* Nous obtenons ainsi $G' \in$ 1-Lirss.

Nous concluons la preuve en rajoutant les sommets de degré 1 par application du lemme 3.3. □

D'après le théorème 3.2, il est assez facile de vérifier si un graphe supporte ou non une fonction de routage par intervalles linéaires. Par exemple :

Corollaire 3.2 *Tout graphe d'intervalles appartient à* 1-Lirs.

Rappelons qu'un graphe d'intervalles est un graphe dont les sommets sont des intervalles de $\mathbb{R}$ et dont les arêtes sont les paires d'intervalles qui s'intersectent (voir [42]).

Preuve. Soit $\tilde{Y}$ une forme particulière de lithium : les électrons sont composés exactement de deux sommets. Notez que les sommets du noyau incidents aux isthmes forts peuvent être distincts ou non. Il est facilement de voir que tout $\tilde{Y}$ n'est pas un graphe d'intervalles (voir [42, 43]). Or, tout lithium possède un $\tilde{Y}$ comme sous-graphe. Par conséquent un lithium n'est pas un graphe d'intervalles — tout sous-graphe d'un graphe d'intervalles est un graphe d'intervalles —, ce qui est équivalent à dire d'après le théorème 3.2 que tout graphe d'intervalles appartient à 1-Lirs. □

Notez que C_n (le cycle d'ordre n) pour $n \geqslant 4$, n'est pas un graphe d'intervalles [43]. Cependant, pour tout n, $C_n \in$ 1-Lirs de manière triviale par le théorème 3.2. Par conséquent, la classe des graphes 1-Lirs n'est pas réduite à celle des graphes d'intervalles. En fait, cette classe contient la plupart des graphes utilisés comme réseaux d'interconnexions. Cela signifie que les intervalles cycliques ne sont pas nécessaire à la construction des fonctions de routage par intervalles sur les réseaux usuels. Cependant, tout comme la bonne nouvelle du théorème 3.1, celle du théorème 3.2 doit être modérée car les routes induites par ces fonctions de routage ne sont pas nécessairement de plus courts chemins.

Caractérisons maintenant la classe des graphes 1-Lirss.

Définition 3.5 (lithium faible) *Un* lithium faible *est un graphe possédant au moins trois isthmes incidents à une même composante connexe.*

Notez que tout lithium est un lithium faible (un lithium est donc un lithium faible où chacun des électrons a au moins deux sommets, voir figure 3.2).

Théorème 3.3 $G \in$ 1-Lirss $\Leftrightarrow$ *G n'est pas un lithium faible.*

Preuve. Soit G d'ordre n.

$\Rightarrow$ Supposons $G \in$ 1-Lirss et que G soit un lithium faible. Considérons les sommets des électrons appartenant aux isthmes reliant les électrons au noyau. Nécessairement, un de ces sommets n'est pas numéroté 1 ou n. Soit x ce sommet et e l'isthme entre x et le noyau. On a $I_e = [1, n]$ car sinon la fonction de routage ne serait pas connexe. Cependant, par hypothèse, $1 < \mathcal{L}(x) < n$, et donc I_e contient $\mathcal{L}(x)$: contradiction.

$\Leftarrow$ Si G n'est pas un lithium faible alors il n'est pas un lithium et peut être décomposé comme dans la preuve du théorème 3.2 en un chemin $G_0 - G_1 - \cdots - G_k$, où les G_i sont connectés par des isthmes forts. Puisque G n'est pas un lithium faible, nous avons $\deg(x) \neq 1$ pour tout sommet x de G_i, $i \in \{1, \ldots, k-1\}$. Or, dans G_0 et G_k il existe au plus un sommets de degré 1 si $k > 0$, et au plus deux sommets de degré 1 si $k = 0$. Cela signifie que nous pouvons décomposer G en un chemin $\{x\} - G'_0 - G_1 - \cdots - G'_k - \{y\}$ où x (respectively y) est un sommet de degré 1 de G_0 (respectively de G_k) s'il existe. G'_0 (respectively G'_k) est obtenu à partir de G_0 (respectively de G_k) en supprimant x (respectively y). Nous pouvons alors appliquer la même construction que celle de la preuve de théorème 3.2 sur le chemin $\{x\} - G'_0 - G_1 - \cdots - G'_k - \{y\}$. Puisque chaque composante est d'arête-connectivité 2, la fonction de routage construite utilise seulement un intervalle strict et linéaire par arc (d'après le lemme 3.2). □

Le théorème 3.3 a une conséquence directe sur le produit cartésien. On rappelle que le produit cartésien d'un graphe $G = (V_G, E_G)$ par un graphe $H = (V_H, E_H)$ est le graphe noté $G \times H$, dont les sommets sont les éléments du produit cartésien de V_G par V_H, et les arêtes les paires $\{(x, y), (x', y')\}$ telles que soit $(x, x') \in E_G$ qu'ou bien $(y, y') \in E_H$.

Corollaire 3.3 *Soit G et H deux graphes d'ordre au moins deux,* $G \times H \in$ 1-Lirss.

La figure 3.4 résume les résultats que nous avons obtenu dans cette section. Dans la section suivante, nous étudions la longueur des chemins induits par une fonction de routage par intervalles.

3.5 Efficacité d'une fonction de routage par intervalles

Nous avons caractérisé précédemment les graphes admettant une fonction de routage par intervalles de compacité (linéaire) 1. Nous avons vu que les constructions des théorèmes 3.1 et 3.2 ne fournissaient pas nécessairement des fonctions de routage de plus courts chemins. Dans cette section nous allons étudier la relation entre la compacité et la longueur des chemins de routage induits.

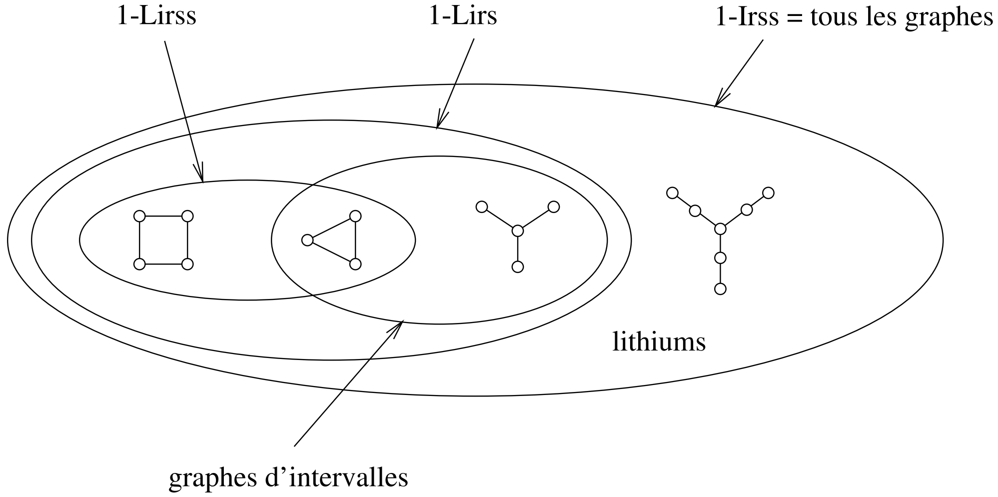


FIG. 3.4 – *Les classes* 1-Irss, 1-Lirs *et* 1-Lirss.

3.5.1 Routage par intervalles de plus courts chemins

Définition 3.6 (compacité d'un graphe) *Soit G un graphe d'ordre au moins* 2*. La* compacité *et la* compacité linéaire *de G, notées respectively* IRS(G) *et* LIRS(G)*, vérifient :*

$$\mathrm{IRS}(G) = \min_R \mathrm{Irs}(R) \quad et \quad \mathrm{LIRS}(G) = \min_R \mathrm{Lirs}(R)$$

où les minimisations sont effectuées sur les fonctions de routage par intervalles de plus courts chemins sur G.

La compacité d'un graphe est un paramètre important. En effet, elle permet de mesurer directement l'efficacité du routage par intervalles sur ce graphe en terme d'espace mémoire requis pour le routeur afin d'assurer l'acheminement des messages sur des plus courts chemins. Clairement, la mémoire de routage d'un sommet est au plus $O(\mathrm{IRS}(G)\, d \log n)$ bits, pour les graphes G d'ordre n et de degré maximum d.

Notations. Pour tout entier $k \geqslant 1$, on note k-IRS (respectively k-LIRS), l'ensemble des graphes supportant une fonction de routage par intervalles de plus courts chemins de compacité (respectively de compacité linéaire) au plus k. De manière similaire, on note k-IRSS (respectively k-LIRSS), l'ensemble des graphes supportant une fonction de routage par intervalles stricts de plus courts chemins et de compacité (respectively de compacité linéaire) au plus k. On a donc : $\mathrm{IRS}(G) \leqslant k \Leftrightarrow G \in k$-IRS et $\mathrm{LIRS}(G) \leqslant k \Leftrightarrow G \in k$-LIRS.

Nous allons tout d'abord donner plusieurs classes de graphes à 1-LIRS et 1-IRS.

a) Familles de graphes de compacité 1

Théorème 3.4 *Tout graphe d'intervalles unitaires appartient à* 1-LIRSS.

Un graphe d'intervalles unitaires est un graphes d'intervalles où tous les intervalles, représentant les sommets, ont même longueur (voir [43]).

Preuve. Soit G un graphe d'intervalles unitaires d'ordre n. Chaque sommet x de G peut être représenté par un intervalle J_x de $\mathbb{R}$, tous les intervalles étant de même longueur. Pour tout sommet x, soit $\alpha(x)$ et $\beta(x)$ tels que $J_x = [\alpha(x), \beta(x)]$. Nous construisons une fonction de routage par intervalles linéaires de plus courts chemins $R = (\mathcal{L}, \mathcal{I})$ comme suit.

Numérotons les sommets de la gauche vers la droite en respectant l'ordre des $\alpha(x)$. Soit x un sommet de G, soit alors $x_{\min}$ (respectively $x_{\max}$) le sommet satisfaisant $\alpha(x_{\min}) = \min_{y \neq x, J_y \cap J_x \neq \emptyset} \alpha(y)$ (respectively $\alpha(x_{\max}) = \max_{y \neq x, J_y \cap J_x \neq \emptyset} \alpha(y)$).

Supposons d'abord que $x_{\min} \neq x_{\max}$. Soit alors $K_{\min} = [1, \mathcal{L}(x_{\min})]$, $K_{\max} = [\mathcal{L}(x_{\max}), n]$, et pour tout voisin y de x distinct de $x_{\min}$ et de $x_{\max}$, $K_y = [\mathcal{L}(y)]$. Posons $I_{(x,x_{\min})} = K_{\min}$, $I_{(x,x_{\max})} = K_{\max}$, $I_{(x,y)} = K_y$ (voir figure 3.5). On a pour tout y (s'il existe), $K_{\min} \cap K_y = K_{\max} \cap K_y = K_{\min} \cap K_{\max} = \emptyset$, ce qui satisfait la propriété de disjonction de la fonction de routage par intervalles (voir la définition 3.1). De plus, $K_{\min} \cup K_{\max} \cup (\bigcup_y K_y) = [1, n]$ ce qui satisfait la propriété d'union. Finalement les intervalles $K_{\min}$, $K_{\max}$ et K_y ne contiennent pas $\mathcal{L}(x)$.

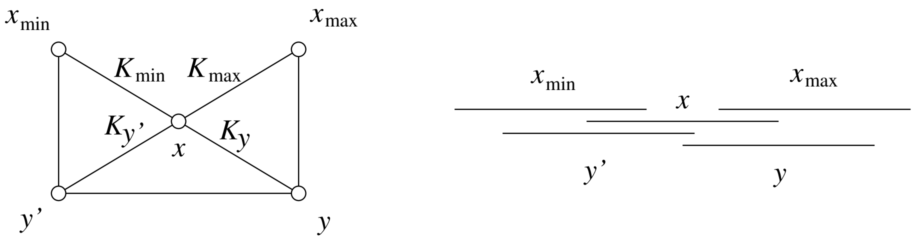


FIG. 3.5 – *Affectation des intervalles pour un graphe d'intervalles unitaires.*

Si $x_{\min} = x_{\max}$ alors $\mathcal{L}(x) = 1$ ou $\mathcal{L}(x) = n$. Dans le premier cas (respectively le second cas), l'intervalle de l'unique arc d'extrémité x est affecté à $]1, n]$ (respectively $[1, n[$).

Prouvons la propriété $\mathcal{P}_k$ suivante : la fonction de routage définie ci-dessus construit un plus court chemin entre deux sommets quelconques distant d'au plus k. $\mathcal{P}_1$ est vraie. Supposons $\mathcal{P}_{k'}$ vraie pour tout $k' \in \{1, \dots, k-1\}$ et soit x et y deux sommets à distance k. Puisque tous les intervalles sont de même longueur, si $\mathcal{L}(x) < \mathcal{L}(y)$ alors $x_{\max}$ est sur un plus court chemin entre x et y, et $\mathcal{L}(y) \in K_{\max}$; sinon $\mathcal{L}(x) > \mathcal{L}(y)$ et alors $x_{\min}$ est sur un plus court chemin entre x et y, et $\mathcal{L}(y) \in K_{\min}$. Donc $\mathcal{P}_k$ est vraie puisque $\mathcal{P}_{k-1}$ est vraie. □

Notez que la réciproque de ce théorème est fausse puisque $C_4 \in$ 1-LIRSS et C_4 n'est pas un graphe d'intervalles unitaires. De même, le graphe $K_{1,3} \in$ 1-LIRS, et est un lithium faible. Ce n'est cependant pas un graphe d'intervalles unitaires. Notez aussi qu'il y a des graphes d'intervalles n'appartenant pas à 1-LIRS. Par exemple, le graphe de la figure 3.6(a). Supposons en effet qu'il appartienne à 1-LIRS. Soit alors x et y deux sommets distincts de 1 et de 7. L'intervalle $I_{(x,z)}$ doit contenir 1 et 7, donc $I_{(x,z)} = [1, 7]$ et, par conséquent, la route de x à y n'est pas un plus court chemin : contradiction. Bien sûr, comme l'exemple du lithium le suggère, il est facile de généraliser les graphes où l'argument précédent s'applique. Cependant contrairement à la classe 1-Lirs, aucune caractérisation simple de la classe 1-LIRS n'est connue à ce jour.

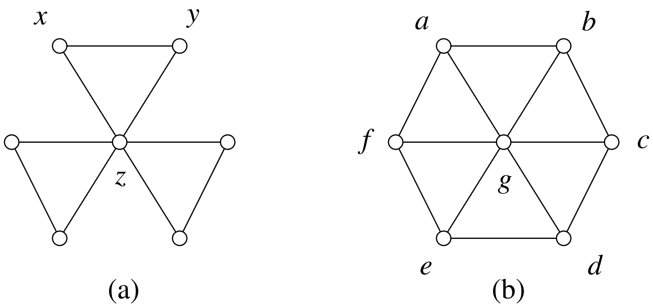


FIG. 3.6 – (a) *Un graphe d'intervalles n'appartenant pas à* 1-LIRS *et* (b) *un graphe d'arcs circulaires n'appartenant pas à* 1-IRS.

Le lecteur peut facilement vérifier que $C_5 \notin$ 1-LIRS (une preuve est donnée dans [5]). Ce résultat sera montré plus tard pour tout C_n, $n \geqslant 5$. Toutefois, on peut prouver que $C_n \in$ 1-IRS pour tout n, en construisant à la main une fonction de routage par intervalles sur C_n. Ceci est en fait une conséquence du résultat plus général ci-dessous.

Théorème 3.5 *Tout graphe d'arcs circulaires unitaires appartient à* 1-IRSS.

Rappelons qu'un graphe d'arcs circulaires est un graphe où les sommets sont représentés par un arc de cercle et où deux sommets sont adjacents si et seulement si leurs arcs s'intersectent. Un graphe d'arcs circulaires unitaires est un graphe d'arcs circulaires tel que tous les arcs ont la même longueur (voir [43]).

Preuve. Soit $G = (V, E)$ un graphe d'arcs circulaires unitaires. Si G est un graphe d'intervalles unitaires alors $G \in$ 1-LIRSS, par le théorème 3.4. Supposons que G n'est pas un graphe d'intervalles unitaires. Considérons une représentation de G sur le cercle trigonométrique. Une fois la représentation fixée, nous pouvons associer un angle $\theta \in [0, 2\pi[$ à cette représentation : l'écartement des arcs de cercles. Chaque sommet x peut ainsi être représenté par un angle $\theta_x \in [0, 2\pi[$ (par exemple l'angle entre l'axe horizontal et la ligne joignant le centre du cercle et le milieu de l'arc représentant x). Deux sommets x et y sont alors adjacents si et seulement si $|\theta_x - \theta_y| \leqslant \theta$.

Nous numérotons x par son rang dans l'ensemble de tous les angles $\theta_y, y \in V$. Nous fixons les intervalles comme suit. Soit $x \in V$ et soit z le sommet à distance excentricité de x (qui est la distance maximum entre x et un quelconque sommet de G) de x dans le sens trigonométrique en maximisant l'angle $\theta_z - \theta_x$. Soit alors x^- (respectively x^+) le voisin de x maximisant $\theta_y - \theta_x \in [0, 2\pi[$ (respectively $\theta_x - \theta_y \in [0, 2\pi[$) parmi les voisins y de x satisfaisants $0 \leqslant \theta_y - \theta_x \leqslant \theta$ (respectively $0 \leqslant \theta_x - \theta_y \leqslant \theta$). Puisque G n'est pas un graphe d'intervalles, $x^- \neq x^+$. Soit $K^- =]\mathcal{L}(z), \mathcal{L}(x^-)]$, $K^+ = [\mathcal{L}(x^+), \mathcal{L}(z)]$ et, pour tout voisin y de x distinct de x^+ et de x^-, $K_y = [\mathcal{L}(y)]$. Fixons enfin $I_{(x,x^-)} = K^-$, $I_{(x,x^+)} = K^+$ et $I_{(x,y)} = K_y$. Les propriétés d'union et de disjonctions sont clairement satisfaites et $\mathcal{L}(x)$ n'appartient pas à $I_{(x,y)}$, pour tout $(x, y) \in E$.

Comme dans la preuve du théorème 3.4, on peut vérifier, par récurrence sur la longueur des chemins, que les chemins construits par la fonction de routage sont des plus courts chemins. □

Comme dans la preuve du théorème 3.4 la réciproque du théorème 3.5 n'est pas vraie puisque $K_{1,3} \in$ 1-IRSS est n'est pas un graphe d'arcs circulaires unitaires. Notez également qu'il existe des graphes d'arcs circulaires qui n'appartiennent pas à 1-IRS. Par exemple, le graphe de la figure 3.6(b). Supposons qu'il appartienne à 1-IRS et considérons $I_{(a,g)}$. Cet intervalle doit contenir $\mathcal{L}(g)$ et $\mathcal{L}(d)$, mais ni $\mathcal{L}(b)$ ni $\mathcal{L}(f)$. De manière similaire, $I_{(c,g)}$ doit contenir $\mathcal{L}(g)$ et $\mathcal{L}(f)$, mais ni $\mathcal{L}(b)$ ni $\mathcal{L}(d)$. Finalement, $I_{(e,g)}$ doit contenir $\mathcal{L}(g)$ et $\mathcal{L}(b)$, mais ni $\mathcal{L}(d)$ ni $\mathcal{L}(f)$, ce qui est impossible.

b) Outils pour la reconnaissance des routages par intervalles de plus courts chemins

Les deux résultats suivants sont très utiles pour savoir si un graphe appartient à k-IRS ou à k-LIRS, pour k fixé. Rappelons la définition d'un sous-graphe des plus courts chemins déjà utilisée dans [5].

Définition 3.7 (sous-graphe des plus courts chemins) *Un graphe G est un* sous-graphe des plus courts chemins *de H si et seulement si G est un sous-graphe de H et si tous les plus courts chemins de H entre tout couple de sommets de G sont contenus dans G.*

Par exemple C_5 est un sous-graphe des plus courts chemins du graphe représenté sur la figure 3.1 page 15, mais pas C_6.

Théorème 3.6 *Pour tout entier $k \geqslant 1$ et toute classe de graphes k-$\mathcal{G}$ égale à k-*LIRSS*, k-*LIRS*, k-*IRSS *ou k-*IRSS*, si $G \in k$-$\mathcal{G}$ alors pour tout sous-graphe des plus courts chemins G' de G, $G' \in k$-$\mathcal{G}$.*

La compacité d'un graphe est donc toujours au moins aussi grande que la compacité d'un de ses sous-graphes des plus courts chemins. Notez que la réciproque du théorème 3.6 est une trivialité puisque G est sous-graphe des plus courts chemins de lui-même.

Preuve. Soit $G' = (V', E')$ un sous-graphe des plus courts chemins de $G = (V, E)$. Soit $R = (\mathcal{L}, \mathcal{I})$ une fonction de routage par intervalles de plus courts chemins sur G. On définit $R' = (\mathcal{L}', \mathcal{I}')$ sur G' de la manière suivante : $\mathcal{L}'(x)$ est le rang de x dans $\mathcal{L}(V')$ trié par ordre croissant.

Soit f et g deux fonctions de $[1, |V|]$ vers V' telles que $f(a)$ retourne le sommet de V' qui a le plus petit numéro $\geqslant a$ et $g(b)$ retourne le sommet de V' qui a le plus grand numéro $\leqslant b$. Pour tout $e \in E'$, si $I_e = [a, b]$ alors nous posons $I'_e = [\mathcal{L}'(f(a)), \mathcal{L}'(g(b))]$. Puisque $\mathcal{L}'^{-1}([\mathcal{L}'(f(a)), \mathcal{L}'(g(b))]) = \mathcal{L}^{-1}([a, b]) \cap V'$, les propriétés d'union et de disjonction sont clairement satisfaites.

Soit $x, y \in V'$ et $(x, z) \in E'$ tel que $I'_{(x,z)} = [a, b]$. Nous avons :

$$\mathcal{L}'(y) \in I'_{(x,z)} = [\mathcal{L}'(f(a)), \mathcal{L}'(g(b))] \quad \Leftrightarrow \quad \mathcal{L}(y) \in I_{(x,z)} = [a, b]$$

Cette propriété est évidemment encore vérifiée si $I_{(x,z)}$ est représenté avec plus d'un intervalle. Ainsi, les routes induites par R' sont les mêmes que celles induites par R, restreint aux sommets de V'. De plus le nombre d'intervalles de $I'_{(x,z)}$ est au plus égale à celui nécessaire pour représenter $I_{(x,z)}$. Enfin, si $\mathcal{L}(x) \notin I_{(x,z)}$ alors $\mathcal{L}'(x) \notin I'_{(x,z)}$, ce qui garantit la conservation des intervalles stricts le cas échéant. □

Dans la section 3.6, nous verrons plusieurs applications du théorème 3.6. Ce résultat est aussi utilisé pour prouver le prochain théorème établissant des résultats concernant le produit cartésien de deux graphes. Ce résultat est particulièrement intéressant pour la construction des réseaux de processeurs (grille, tore, hypercube, ...).

Théorème 3.7 *Pour tout entier $k \geqslant 1$,*

i. $G \in k$-LIRSS *et* $H \in k$-LIRSS $\Rightarrow$ $G \times H \in k$-LIRSS
ii. $G \in k$-LIRS *et* $H \in k$-LIRSS $\Rightarrow$ $G \times H \in k$-LIRS
iii. $G \in k$-LIRSS *et* $H \in k$-IRSS $\Rightarrow$ $G \times H \in k$-IRSS
iv. $G \in k$-LIRS *et* $H \in k$-IRSS $\Rightarrow$ $G \times H \in k$-IRS
v. $G \notin k$-LIRS $\Rightarrow$ $G \times H \notin k$-LIRS *pour tout graphe* H
vi. $G \notin k$-IRS $\Rightarrow$ $G \times H \notin k$-IRS *pour tout graphe* H

Notez qu'on ne peut rien dire sur le produit cartésien de deux graphes appartenant à k-IRS, ni même de deux graphes de k-IRSS (le tore 5×5 en est un exemple pour $k = 1$). La distinction des fonctions de routage par intervalles stricts et/ou linéaires trouve ici tout son intérêt. Pour pouvoir appliquer le théorème 3.7, il faut en effet que l'un des deux graphes supporte une fonction de routage avec des intervalles stricts et le second des intervalles linéaires.

Preuve. Les deux derniers résultats (v) et (vi) sont des conséquences directes du théorème 3.6 puisque le graphe $G \times H$ contient G comme sous-graphe des plus courts chemins.

Soit n_G l'ordre de G et n_H l'ordre de H.

Pour prouver (i), posons $R_G = (\mathcal{L}_G, \mathcal{I}_G)$ et $R_H = (\mathcal{L}_H, \mathcal{I}_H)$ les fonctions de routage par intervalles définies sur les graphes G et H respectively et qui réalisent G et $H \in k$-LIRSS. Soit f la bijection $f : [1, n_G] \times [1, n_H] \to [1, n_G n_H]$ qui satisfait $f(a,b) = a + (b-1)n_G$. Nous définissons la numérotation $\mathcal{L}$ des sommets de $G \times H$ par $\mathcal{L}(z) = f(\mathcal{L}_G(x), \mathcal{L}_H(y))$ pour tout $z = (x,y) \in V_G \times V_H$. Soient $z = (x,y)$ et $z' = (x',y')$ deux sommets voisins de $G \times H$ et soit $e = (z,z')$. Nous posons :

$$I_e = \begin{cases} [f(a, \mathcal{L}_H(y)), f(b, \mathcal{L}_H(y))] & \text{si } y = y' \text{ et } I_{(x,x')} = [a,b] \in \mathcal{I}_G \\ [f(1,c), f(n_G, d)] & \text{si } x = x' \text{ et } I_{(y,y')} = [c,d] \in \mathcal{I}_H \end{cases}$$

Clairement, si $[a,b] \cap [a',b'] = \emptyset$ alors $[f(a, \mathcal{L}_H(y)), f(b, \mathcal{L}_H(y))] \cap [f(a', \mathcal{L}_H(y)), f(b', \mathcal{L}_H(y))] = \emptyset$. De manière similaire, si $[c,d] \cap [c',d'] = \emptyset$ alors $[f(1,c), f(n_G,d)] \cap [f(1,c'), f(n_G,d')] = \emptyset$. Puisque $H \in k$-LIRSS, $\mathcal{L}_H(y) \notin [c,d]$ et donc $[f(a, \mathcal{L}_H(y)), f(b, \mathcal{L}_H(y))] \cap [f(1,c), f(n_G,d)] = \emptyset$. La propriété de disjonction est donc satisfaite. Il est facile de vérifier que la propriété d'union est aussi satisfaite. La fonction de routage $R = (\mathcal{L}, \mathcal{I})$ construite sur $G \times H$ comme ci-dessus route les messages comme suit. Le chemin d'un sommet (x,y) vers un autre sommet (x',y') passe d'abord par (x,y') à l'intérieur de la x-ième copie de H. Il circule ensuite dans la y'-ième copie de G jusqu'à (x',y'). Ces deux parties de routes utilisent des plus courts chemins. Par conséquent $G \times H \in k$-LIRS. De plus, puisque $\mathcal{L}_G(x) \notin [a,b]$, $G \times H \in k$-LIRSS.

Si $\mathcal{L}_G$ n'est pas strict, nous avons seulement $G \times H \in k$-LIRS et la propriété (ii) est démontrée. Si $\mathcal{I}_H$ contient un intervalle cyclique $[c, d]$, alors l'intervalle $[f(1, c), f(n_G, d)]$ est aussi cyclique et $G \times H \in k$-IRS prouvant le point (iv). Finalement, si $\mathcal{L}_G$ est strict, nous avons $G \times H \in k$-IRSS, et la propriété (iii) est démontrée. $\square$

Les points (i) et (ii) du théorème précédent ont été démontrés de manière indépendante par Kranakis, Krizanc et Ravi dans [58].

Les théorèmes 3.6 et 3.7 sont des outils de bases. Un troisième outil sera décrit dans la section 3.6. Le dernier résultat de cette section concerne les classes de graphes supportant une fonction de routage par intervalles stricts et de plus courts chemins. Il montre comment, en augmentant la compacité, on peut toujours construire une fonction de routage par intervalles stricts.

Proposition 3.1 *Pour toute fonction de routage par intervalles non strict de compacité k sur G, il existe un fonction de routage par intervalles stricts de compacité au plus $k+1$ sur G, construisant les mêmes ensembles de chemins entre toute paire de sommets de G.*

Preuve. Pour tout $I_{(x,y)} \in \mathcal{I}$ tel que $\mathcal{L}(x) \in I_{(x,y)}$, on pose $I'_{(x,y)} = I_{(x,y)} \backslash \{\mathcal{L}(x)\}$ et $\mathcal{L}' = \mathcal{L}$. Clairement la compacité de $R' = (\mathcal{L}', \mathcal{I}')$ a augmenté au plus de un. $\square$

3.5.2 Fonctions de routage efficaces et compactes

D'un point de vue pratique, les concepteurs de systèmes de routage doivent établir un compromis entre les contraintes matérielles et l'efficacité des communications. On peut imposer d'avoir des communications très rapides et accepter d'accroître la surface physique du routeur par exemple, ou bien d'imposer une surface aussi petite que possible même si cela implique une dégradation de l'efficacité des communications.

a) Temps minimum pour une compacité fixée

Définition 3.8 (temps d'une fonction de routage par intervalles et d'un graphe) *Soient un graphe G et un entier $k \geqslant 1$.*

- *Le* temps *d'une fonction de routage par intervalles R sur G est la longueur maximum des chemins construits par R.*
- *Le k-*temps *(respectively le k-*temps linéaire*) de G et le minimum pris sur toutes les fonctions de routage par intervalles R sur G de compacité (respectively de compacité linéaire) k du temps de R.*

Le résultat suivant découle du théorème 3.1. Il montre que le routage par intervalles est assez efficace puisqu'il est toujours possible de définir une fonction de routage qui construit des chemins de longueur maximum petite par rapport au diamètre.

Corollaire 3.4 *Soit G un graphe de diamètre D, de rayon r et de 1-temps t. Alors $D \leqslant t \leqslant 2r$.*

Ce résultat est particulièrement intéressant pour les graphes de diamètre $2r$, comme par exemple le graphe de la figure 3.6(b).

Preuve. Par définition, $D \leqslant t$. Pour montrer $t \leqslant 2r$ il suffit de choisir, dans la preuve du théorème 3.1, un arbre couvrant de plus courts chemins enraciné en un sommet d'excentricité minimum r. □

Cette borne est la meilleure possible dans la mesure où, pour tout r, il existe un graphe de rayon r de 1-temps $t = 2r$. Par exemple un chemin de $2r+1$ sommets. Notons que l'on doit modérer l'utilité de ce corollaire en remarquant que, dans la fonction de routage utilisée dans la preuve, beaucoup de messages vont traverser un seul sommet (la racine de l'arbre) et par conséquent générer des conflits. Nous résumons ci-dessous les résultats les plus récents sur le pire cas du k-temps en fonction du diamètre D, pour $k = 1$ et $k = 2$:

Théorème 3.8 *Pour tout entier D :*
[73] *Il existe un graphe de diamètre D de 1-temps au moins $3D/2 + 1/2$.*
[78] *Il existe un graphe de diamètre D de 2-temps au moins $5D/4 - 1$.*

Notons que de nombreuses expérimentations peuvent être trouvées dans [51] où il est étudié le 1-temps de certains réseaux comme le tore et le 3-temps d'un graphe dans le cas général.

Concernant les intervalles linéaires, la preuve du théorème 3.2 ne permet pas de donner une borne supérieure utilisable sur le 1-temps linéaire d'un graphe non lithium. De plus, la figure 3.7 montre un graphe sur lequel la preuve constructive du théorème 3.2 donne une fonction de routage par intervalles très inefficace. Le 1-temps linéaire de ce graphe d'ordre n, décomposé suivant la construction en $G_0 - G_1 - G_2$, est $n - 4$ alors que son diamètre est 5. En effet, le chemin de routage de z vers y ne peut pas passer par x selon la construction du théorème 3.2 appliquée à ce graphe. Cependant, nous donnons ci-dessous une borne du pire cas du 1-temps linéaire en fonction du diamètre.

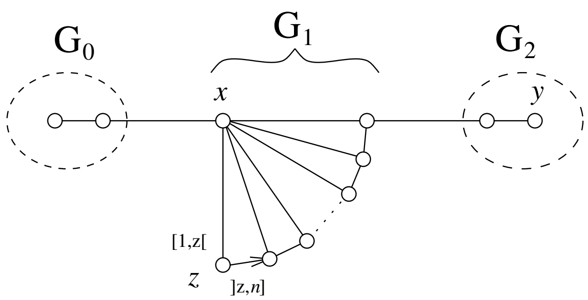


FIG. 3.7 – *La preuve constructive du théorème 3.2 peut induire des fonctions de routage inefficaces.*

Théorème 3.9 *Pour tout D, il existe un graphe de diamètre D et de 1-temps linéaire au moins $2D - 1$.*

La preuve de ce théorème est une simple conséquence de la proposition suivante.

Proposition 3.2 *Pour tout $n \geqslant 3$, le 1-temps linéaire de C_n est $n - 2$.*

Cette borne est un pire cas puisque le temps d'une fonction de routage par intervalles sur un graphe d'ordre n est au plus $n-1$ (clairement pour toute fonction de routage simple un message ne peut traverser qu'au plus $n-1$ arêtes).

Preuve. Si $n = 3$ ou 4, alors il existe une fonction de routage par intervalles linéaires sur C_n de temps 1 ou 2 respectively, et il n'est pas possible de faire mieux.

Montrons que, pour $n \geqslant 5$, le 1-temps linéaire de C_n est $\tau \geqslant n-2$. Supposons $\tau < n-2$, et soit $R = (\mathcal{L},\mathcal{I})$ la fonction de routage par intervalles linéaires sur C_n de temps τ. Soit P le chemin $y' - x' - t - x - y$ dans C_n où $\mathcal{L}(t) = 1$. Soit z le sommet de P qui a le numéro le plus élevé parmi les numéros des sommets de P.

Si $z = x$, alors l'intervalle $I_{(x',t)}$ doit contenir $\mathcal{L}(t)$ et $\mathcal{L}(z)$ car sinon le temps pour atteindre t ou z depuis x' serait $\geqslant n-2$ par l'arc (x', y'). Pour la même raison $I_{(x',t)}$ ne peut pas contenir $\mathcal{L}(y')$. Cependant $\mathcal{L}(y') \in [\mathcal{L}(t), \mathcal{L}(z)] \subset I_{(x',t)}$: contradiction.

Si $z = y$, alors l'intervalle $I_{(t,x)}$ doit contenir $\mathcal{L}(x)$ et $\mathcal{L}(z)$ mais pas $\mathcal{L}(x')$. Si $\mathcal{L}(x') > \mathcal{L}(x)$ alors $\mathcal{L}(x') \in [\mathcal{L}(x), \mathcal{L}(z)]$: contradiction. Si $\mathcal{L}(x') < \mathcal{L}(x) < \mathcal{L}(y')$ alors puisque $I_{(t,x')}$ doit contenir $\mathcal{L}(x')$ et $\mathcal{L}(y')$ mais pas $\mathcal{L}(x)$, nous obtenons également une contradiction. Finalement, si $\mathcal{L}(x') < \mathcal{L}(x)$ et $\mathcal{L}(y') < \mathcal{L}(x)$ alors, puisque $I_{(x',t)}$ doit contenir $\mathcal{L}(t)$ et $\mathcal{L}(x)$ mais pas $\mathcal{L}(y')$, nous obtenons une autre contradiction éliminant le cas $z = y$.

Si $z = x'$ ou $z = y'$, nous obtenons des contradictions en inversant le rôle des sommets x, y et x', y'. Donc $\tau \geqslant n-2$. Il existe une fonction de routage par intervalles linéaires sur C_n de temps $n-2$: la numérotation des sommets est cyclique et on pose $I_{(i,i+1)} =]i, n]$ pour $1 \leqslant i < n$, $I_{(n,1)} = [1]$, $I_{(i,i-1)} = [1, i[$ si $1 < i \leqslant n$, et $I_{(1,n)} = [n]$. □

En utilisant le théorème 3.6, nous pouvons déduire de cette proposition une autre preuve du résultat suivant déjà montré dans [5] :

Corollaire 3.5 $C_n \in$ 1-LIRS $\Leftrightarrow n \leqslant 4 \Leftrightarrow C_n \in$ 1-LIRSS.

Le prochain résultat permet, sous certaines conditions, d'obtenir le k-temps d'un produit cartésien.

Théorème 3.10 *Soient k un entier, G un graphe de k-temps linéaire t_G et H un graphe supportant une fonction de routage par intervalles stricts R de temps t_H.*

i. *Si* $\mathrm{Irs}(R) = k$ *alors le k-temps de $G \times H$ est au plus $t_G + t_H$.*
ii. *Si* $\mathrm{Lirs}(R) = k$ *alors le k-temps linéaire de $G \times H$ est au plus $t_G + t_H$.*

Preuve. Considérons la construction de la preuve du théorème 3.7. Les deux conditions pour la construction sont remplies : G supporte une fonction de routage par intervalles de compacité linéaire k et H supporte une fonction de routage par intervalles stricts de compacité ou compacité linéaire k. Clairement, les chemins construits sont de longueur égale à au plus la sommes des plus longs chemins dans G et dans H. Donc ils sont de longueur au plus $t_G + t_H$. □

b) Compacité minimum pour des fonctions de routage de plus courts chemins

Le théorème suivant améliore le résultat donné dans [80].

Théorème 3.11 *Pour tout G d'ordre $n \geqslant 2$,* $\mathrm{LIRS}(G) \leqslant \dfrac{n}{2}$ *et* $\mathrm{IRS}(G) \leqslant \dfrac{n-1}{2}$.

Preuve. Il est toujours possible de décrire par une table de routage une fonction de routage de plus courts chemins sur G. Or, pour tout sommet x, il y a au plus $n-1$ destinations associées à chaque arc de E_x. Ces destinations peuvent être codées par au plus $\lfloor n/2 \rfloor$ intervalles linéaires et $\lfloor (n-1)/2 \rfloor$ intervalles pouvant être cycliques. □

Comme pour le paramètre *temps*, il est possible de déduire la compacité d'un produit cartésien de graphes.

Théorème 3.12 *Pour tous graphes G et H :*

- $\max\{\mathrm{LIRS}(G), \mathrm{LIRS}(H)\} \leqslant \mathrm{LIRS}(G\times H) \leqslant \max\{\mathrm{LIRS}(G), \mathrm{LIRS}(H)+1\}$
- $\max\{\mathrm{IRS}(G), \mathrm{IRS}(H)\} \leqslant \mathrm{IRS}(G\times H) \leqslant \max\{\mathrm{LIRS}(G), \mathrm{IRS}(H)+1\}$

Preuve. Comme G et H sont tous deux des sous-graphes des plus courts chemins de $G\times H$, il découle du théorème 3.6 que $\max\{\mathrm{LIRS}(G), \mathrm{LIRS}(H)\} \leqslant \mathrm{LIRS}(G\times H)$ et $\max\{\mathrm{IRS}(G), \mathrm{IRS}(H)\} \leqslant \mathrm{IRS}(G\times H)$. Soient $k_L = \mathrm{LIRS}(H)$ and $k_I = \mathrm{IRS}(H)$. D'après la Proposition 3.1, $H \in (k_L{+}1)$-LIRSS et $H \in (k_I{+}1)$-IRSS. Soient $k = \max\{\mathrm{LIRS}(G), k_L{+}1\}$ et $k' = \max\{\mathrm{LIRS}(G), k_I{+}1\}$. Nous avons tout d'abord $G \in k$-LIRS et $H \in k$-LIRSS, et, d'autre part, $G \in k'$-LIRS et $H \in k'$-IRSS. Par application du théorème 3.7, nous montrons que $G\times H \in k$-LIRS et que $G\times H \in k'$-IRS. □

Le résultat suivant montre que, malheureusement, il n'existe pas de borne de la valeur maximale de la compacité d'un graphe.

Théorème 3.13 *Pour tout $k \geqslant 1$, il existe un graphe G tel que $G \notin k$-LIRS.*

Preuve. Pour $k = 1$ nous utilisons l'exemple de la figure 3.6(b) page 25.

Soit $k \geqslant 2$, et considérons le graphe G composé de trois composantes G_1, G_2 et G_3, toutes connectées à un même sommet u (voir figure 3.8). Chaque G_i a trois « niveaux ». Le premier niveau est composé de $2k-1$ sommets indépendants $x_1, \ldots, x_{2k-1}$, chaque x_i étant connectés à u. Le second niveau est composé de $\binom{2k-1}{k-1}$ sommets indépendant que l'on nomme $(i_1, \ldots, i_{k-1})$ pour $1 \leqslant i_1 < \ldots < i_{k-1} \leqslant 2k-1$. Le sommet $(i_1, \ldots, i_{k-1})$ est connecté aux $k-1$ sommets $x_{i_1}, \ldots, x_{i_{k-1}}$. Le troisième niveau est un graphe complet de même ordre que le second niveau. Il y a une connexion bijective entre le second et le troisième niveau.

Soit n l'ordre de G. Supposons $G \in k$-LIRS et soit $R = (\mathcal{L}, \mathcal{I})$ la fonction de routage par intervalles correspondante. Considérons le sous-graphe G_i tel qu'aucun de ses sommets ne soit numéroté 1 ou n (le graphe G_1 sur la figure 3.8). Puisque les sommets x_i jouent le même rôle, supposons que $\mathcal{L}(x_1) < \ldots < \mathcal{L}(x_{2k-1})$. Considérons alors l'arc e entre le sommet y et son homologue du second niveau $z = (2, 4, 6, \ldots, 2k-2)$. D'après la structure de G, nous avons d'une part 1, n et $\mathcal{L}(x_{2i}) \in I_e$, pour tout $i \in \{1, \ldots, k-1\}$, et d'autre part $\mathcal{L}(x_{2i-1}) \notin I_e$, pour tout $i \in \{1, \ldots, k\}$. I_e est donc composé d'au moins $k+1$ intervalles linéaires. Donc $G \notin k$-LIRS. □

Par conséquent en considérant l'exemple de la figure 3.6 page 25 et puisque k-IRS $\subset$ $(k{+}1)$-IRS, nous avons :

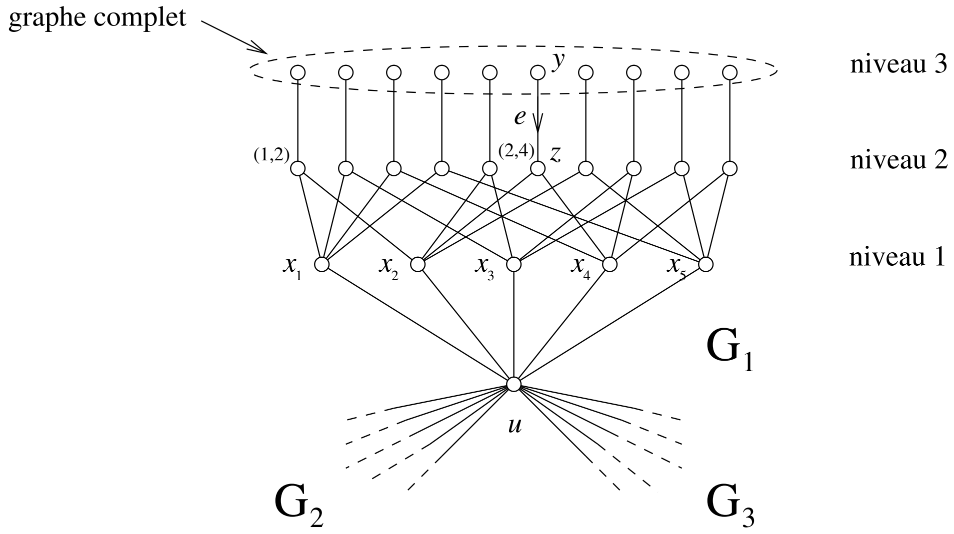


FIG. 3.8 – *Un graphe n'appartenant pas à* 3-LIRS.

Corollaire 3.6 *Pour tout $k \geqslant 1$, il existe un graphe G tel que $G \notin k$-*IRS.

Remarque. L'ordre du graphe précédent, en fonction de k, vaut $3(2k-1+2\binom{2k-1}{k-1})+1 = \Theta(4^k/\sqrt{k})$ (en utilisant l'approximation de Stirling). Sa compacité, en comparaison avec son ordre n, vaut donc au moins $\Omega(\log n)$. Son diamètre est 6 et son degré est $\binom{2k-1}{k-1}-1 = \Theta(n)$.

Nous allons maintenant étudier les propriétés que satisfont les graphes considérés comme réseaux d'interconnexions d'ordinateurs parallèles, ou comme candidats possibles à une telle utilisation.

3.6 Réseaux usuels

Dans cette section, nous décrivons les constructions de fonction de routage par intervalles (linéaires) pour les réseaux d'interconnexions usuels. Nous indiquons également, pour ces réseaux, plusieurs paramètres particuliers développés précédemment.

3.6.1 Chemins, cycles et graphes complets

Ces réseaux ont été étudiés tout au long de ce chapitre (voir corollaire 3.5, par exemple). Le cycle C_n d'ordre n, est un graphe d'arcs circulaires unitaires ; le chemin P_n d'ordre n, et le graphe complet K_n sont des graphes d'intervalles unitaires. Donc $C_n \in$ 1-IRSS, et $P_n, K_n \in$ 1-LIRSS par application des théorèmes 3.5 et 3.4.

3.6.2 Grilles

Dans [5], les auteurs ont prouvé que les grilles de n dimensions appartiennent à 1-LIRS. La proposition suivante simplifie leur preuve :

Proposition 3.3 *La grille G de n dimensions appartient à* 1-LIRSS.

Preuve. $G = P_{d_1} \times (P_{d_2} \cdots \times P_{d_n}) \in$ 1-LIRSS, par le théorème 3.7. □

3.6.3 Hypercubes généralisés

Dans [5], les auteurs ont prouvé que l'hypercube binaire de dimension n appartient à 1-LIRS. On peut généraliser facilement ce résultat :

Proposition 3.4 *L'hypercube généralisé H_n^d de dimension n sur l'alphabet $\{1, \ldots, d\}$, pour $d \geqslant 2$, appartient à* 1-LIRSS.

Preuve. L'hypercube généralisé est défini par $H_1^d = K_d$ et $H_n^d = H_{n-1}^d \times K_d$, *i.e.* $H_n^d = K_d \times \cdots \times K_d$, n fois. □

3.6.4 Tores

Dans [5], il est montré que le tore à n dimensions $T_n = C_{d_1} \times \cdots \times C_{d_n} \in$ 1-LIRS si et seulement si $d_i < 5$, pour tout $i \in \{1, \ldots, n\}$. Nous généralisons ce résultat dans le théorème suivant.

Théorème 3.14 *Soit $T_n = C_{d_1} \times \cdots \times C_{d_n}$ un tore de n dimensions, telles que $d_1 \leqslant \ldots \leqslant d_n$. Nous avons :*

i. $T_n \in$ 2-LIRSS
ii. $T_n \in$ 1-LIRS $\Leftrightarrow d_n < 5 \Leftrightarrow T_n \in$ 1-LIRSS
iii. $T_n \in$ 1-IRS $\Leftrightarrow d_{n-1} < 5 \Leftrightarrow T_n \in$ 1-IRSS

Preuve. Pour tout n, $C_n \in$ 1-IRSS, donc $C_n \in$ 2-LIRSS. Par le théorème 3.7, nous obtenons (i). Le point (ii) est obtenu par simple application du corollaire 3.5 et du théorème 3.7. Si $d_{n-1} < 5$ alors $C_{d_1} \times \cdots \times C_{d_{n-1}} = T_{n-1} \in$ 1-LIRSS (d'après (ii)). Puisque $C_{d_n} \in$ 1-IRSS, nous pouvons appliquer le théorème 3.7, donc $T_n \in$ 1-IRSS. La réciproque du point (iii) est établie dans [73], *i.e.* $C_{d_1} \times C_{d_2} \notin$ 1-IRS pour $d_1 \geqslant 5$. □

3.6.5 Une liste de réseaux usuels n'appartenant pas à 1-LIRS

a) En utilisant les sous-graphes des plus courts chemins

Proposition 3.5 *Les graphes suivants n'appartiennent pas à* 1-LIRS *:*

- *le* Shuffle-Exchange [62, 76] SE_n, $\forall n \geqslant 3$

– *le* Cube-Connected-Cycle [70] CCC_n, $\forall n \geqslant 2$
– *le* Star-Graph [1] S_n, $\forall n \geqslant 1$

Preuve. La preuve repose sur le fait que tous ces graphes contiennent un cycle d'au moins 5 sommets comme sous-graphe des plus courts chemins.

Pour tout $n \geqslant 5$, SE_n contient C_5 comme sous-graphe des plus courts chemins. En effet, dans la représentation binaire standard des sommets, $C_5 = \{01x10, 01x11, 1x110, x1101, x1100\}$ où $x = 1^{n-4}$. Il est facile de vérifier qu'il est un sous-graphe des plus courts chemins. SE_4 contient C_7 comme sous-graphe des plus courts chemins, et $SE_3 \notin$ 1-LIRS comme nous allons de voir grâce à la proposition 3.6.

Pour tout $n \geqslant 2$, CCC_n contient C_8 comme sous-graphe des plus courts chemins et, pour tout $n \geqslant 1$, S_n contient C_6 comme sous-graphe des plus courts chemins. □

b) Un nouvel outil pour tester la compacité

Soit $R = (\mathcal{L}, \mathcal{I})$ une fonction de routage par intervalles sur $G = (V, E)$ d'ordre n. Nous notons $T_{(x,z)}$ l'ensemble des numéros $\mathcal{L}(y)$ tels que tous les plus courts chemins de x à y traversent l'arc $(x, z) \in E$. Pour tout sous-ensemble I de $\{1, \dots, n\}$, nous notons $[\,I\,]$ le plus petit ensemble tel que $I \subset [\,I\,]$ et $\beta([\,I\,]) = 1$, c'est-à-dire $[\,I\,]$ est le plus petit intervalle linéaire contenant tous les entiers de I. Bien sûr, si LIRS$(G) = 1$ alors pour tout arc $(x, u) \neq (x, v) \in E$, $T_{(x,u)} \cap T_{(x,v)} = \emptyset$. En fait nous avons un résultat un peu plus fort.

Lemme 3.4 *Pour toute fonction de routage par intervalles de plus courts chemins R sur un graphe $G = (V, E)$, si* Lirs$(R) = 1$ *alors*

$$\forall\, (x, u) \neq (x, v) \in E,\ [\,T_{(x,u)}\,] \cap [\,T_{(x,v)}\,] = \emptyset$$

Preuve. Soit $R = (\mathcal{L}, \mathcal{I})$ une fonction de routage par intervalles de compacité linéaire 1 sur G. Clairement, pour tout intervalle $I_e \in \mathcal{I}$, $[\,T_e\,] \subset I_e$. □

Pour tout ensemble A et B de $\{1, \dots, n\}$, nous dirons que A et B sont « séparables » si et seulement si $[\,A\,] \cap [\,B\,] = \emptyset$ (c'est-à-dire soit $\forall (a, b) \in A \times B, a < b$, ou bien $\forall (a, b) \in A \times B, a > b$). Nous notons $A|B$ la propriété « A et B sont séparables ». Il découle de la séparabilité une méthode simple pour prouver qu'un graphe $G = (V, E) \notin$ 1-LIRS. Il suffit de trouver un sous-ensemble de sommets V' de V tel que le système d'équations $\{\,T_{(x,y)} | T_{(x,z)}$, $x \in V'$ et $(x, y) \neq (x, z) \in E\,\}$, mène à une contradiction. Par exemple :

Proposition 3.6 *Les graphes suivants n'appartiennent pas à* 1-LIRS *:*

– *le graphe SE_3*
– *la grille à* 6 *directions (voir figure 3.9(b))*
– *la grille à* 8 *directions (voir figure 3.9(c))*
– *le* Butterfly [62] BF_n, $\forall n \geqslant 2$ *(voir figure 3.9(d))*
– *tous les autres graphes de la figure 3.9*

Preuve. Pour le graphe SE_3, il suffit de montrer que le graphe dessiné sur la figure 3.9(a) n'appartient pas à 1-LIRS. En effet ce graphe est un sous-graphe des plus courts chemins de SE_3. Considérons le systèmes d'équations induits par les sommets grisés de la figure 3.9(a). On suppose que les sommets du graphe ont été numérotés arbitrairement $a, b, \ldots, f$ par une fonction de routage par intervalles de plus courts chemins. On obtient :

$$\begin{array}{llll} \text{en } a: & bc\,|\,de & \text{en } d: & a\,|\,ef \\ \text{en } b: & a\,|\,cf & \text{en } e: & c\,|\,ad \\ \text{en } c: & e\,|\,ab & & \end{array}$$

Supposons $b < e$: $b < e \overset{(a)}{\Rightarrow} c < d \overset{(e)}{\Rightarrow} c < a \overset{(b)}{\Rightarrow} f < a \overset{(d)}{\Rightarrow} e < a \overset{(c)}{\Rightarrow} e < b$: contradiction. Donc, d'après le théorème 3.6, $SE_3 \notin$ 1-LIRS.

Pour tout $n \geqslant 3$, BF_n contient toujours BF_2 comme sous-graphe des plus courts chemins, et $BF_2 \notin$ 1-LIRS en examinant le système induit par les trois sommets grisés de la figure 3.9(d). Pour tous les autres graphes, il suffit également de noter les équations induites par les sommets grisés de la figure 3.9 pour aboutir, dans tous les cas, à une contradiction à l'appartenance à 1-LIRS. □

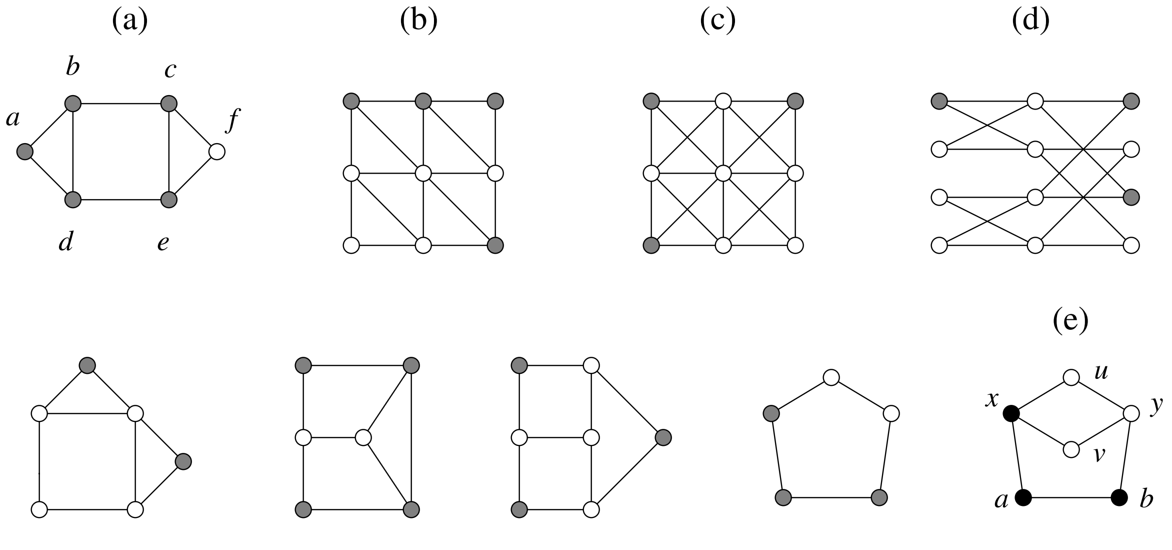


FIG. 3.9 – *Quelques graphes n'appartenant pas à* 1-LIRS.

Remarque. Une façon rapide d'obtenir une solution d'un système $\{(A_i|B_i)_i\}$, obtenu sur un graphe $G = (V, E)$, consiste à considérer le graphe orienté $R_G = (V_R, E_R)$ associé au système vérifiant :

$$V_R = \bigcup_i (A_i \cup B_i) \quad \text{et} \quad E_R = \bigcup_i \{(a, b) \in A_i \times B_i \mid a < b\}.$$

Considérons la fermeture transitive $\overline{R_G}$ de R_G. Si $\overline{R_G}$ possède un cycle, alors $G \notin$ 1-LIRS. Si $\overline{R_G}$ est acyclique alors cela ne prouve pas que $G \in$ 1-LIRS, mais un tri des sommets de $\overline{R_G}$ par degré sortant peut donner des indications sur une numérotation possible des sommets de G. Avec cette technique nous avons trouvé des numérotations des sommets pour les graphes 1-LIRS représentés sur la figure 3.10 (l'affectation des intervalles n'est pas difficile une fois qu'est déterminée la numérotation des sommets). Notez qu'il est dit dans [5] que le graphe dessiné le plus à gauche sur la figure [5] n'appartient pas à 1-LIRS, ce qui n'est manifestement pas vrai, tout comme l'affirmation des mêmes auteurs qui affirment que toute combinaison de plus d'un carré et d'un triangle partageant une face commune n'appartient pas à 1-LIRS.

Même si le lemme 3.4 se révèle très utile pour déterminer si $\mathrm{LIRS}(G) = 1$, on ne peut pas espérer caractériser complètement la classe 1-LIRS à l'aide de cette technique. En effet on ne peut pas déterminer si le graphe (e) de la figure 3.9 appartient à 1-LIRS avec le lemme 3.4 (le lecteur peut vérifier que tout système d'équations a une solution). Pourtant, il n'appartient pas à 1-LIRS. En effet, du sommet x, deux équations sont possibles, $x : ab|uy$ ou $x : ab|vy$, selon le choix de la fonction de routage. Or, dans un cas comme dans l'autre, pour une fonction de routage fixée, le système d'équations induit par les sommets noirs sera le même que celui induit par les sommets grisés du graphe C_5, qui lui n'a pas de solution. La caractérisation de la classe 1-LIRS reste donc difficile (voir également la remarque de la page 41 concernant la NP-complétude).

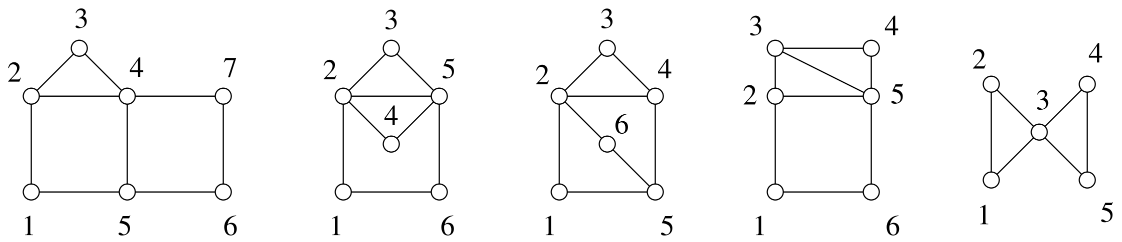


FIG. 3.10 – *Quelques graphes appartenant à* 1-LIRS.

La notion de système d'équations induit peut être généralisée au cas des fonctions de routage par intervalles cycliques. Dans ce cas, pour un ensemble d'entiers A, $[\,A\,]$ est le plus petit intervalle (éventuellement cyclique) contenant tous les entiers de A. Par exemple :

Proposition 3.7 *Pour tout $n \geqslant 3$, $CCC_n \notin$ 1-IRS.*

Preuve. Soit $R = (\mathcal{L}, \mathcal{I})$ une fonction de routage par intervalles sur CCC_n, $n \geqslant 3$. On note $d(x, y)$ la distance entre les sommets x et y. Soient x, y, z les numéros des sommets d'un cycle tel que $d(x, y) < d(x, z) + d(z, y)$ et $d(z, y) < d(z, x) + d(x, y)$. Cette situation se produit toujours dès que $n \geqslant 5$ ou $n = 3$. Nous considérerons plus tard le cas $n = 4$. Soient A, B et C les ensembles des numéros des sommets des cycles A, B et C de la figure 3.11 connectés à x, y et z respectively. Nous avons $A \cup \{x\} \subset T_{(y,x)}$, $B \cup \{y\} \subset T_{(x,y)}$ et $C \cup \{z\} \subset T_{(y,z)}$. Donc $Ax|By$, $Ax|Cz$ et $By|Cz$. Considérons maintenant les numéros a et a' de deux sommets de A (voir figure 3.11). Nécessairement, un intervalle $[\,x, y, z\,]$ doit contenir un des intervalles $[\,A\,]$, $[\,B\,]$ ou $[\,C\,]$ (considérez par exemple l'ensemble $T_{(a,x)}$). Sans perte de généralité, supposons que $[\,A\,] \subset [\,x, y, z\,]$. Clairement l'ensemble $T_{(a,x)}$ doit contenir les numéros x, y et z, mais pas a'. Par conséquent, $[\,T_{(a,x)}\,]$ contient $[\,A\,]$ mais pas a' : contradiction.

Pour le cas particulier $n = 4$, nous utilisons un quatrième cycle D connecté à un sommet numéroté t sur le premier cycle considéré en début de preuve. Comme précédemment, nous pouvons déduire un intervalle $[\,x, y, z, t\,]$ qui doit contenir un des ensembles A, B, C ou D, ce qui aboutit à une contradiction. $\square$

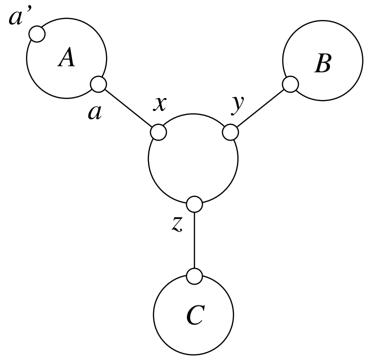


FIG. 3.11 – $CCC_n \notin$ 1-IRS.

CHAPITRE 4

Compacité maximale du routage par intervalles

4.1 Introduction

Comme nous l'avons vu au chapitre précédent, la compacité du routage par intervalles sur un réseau a un impact direct sur la mémoire des routeurs (dans la mesure où ils mettent en œuvre le routage par intervalles). En effet, on a déjà vu que la mémoire de routage d'un routeur d'un réseau d'ordre n, de degré maximum d et de compacité k, est $O(kd \cdot \log n)$ bits. Ce chapitre s'attache précisément à la valeur maximale (pire cas) de la compacité pour les réseaux de n nœuds.

La plupart des travaux sur le routage par intervalles de plus courts chemins décrits dans la littérature ont porté sur le calcul de la compacité de réseaux spécifiques : les anneaux avec cordes dans [18], les arbres, les hypercubes, les grilles et les tores à d dimensions, les graphes r-parties complets dans [5, 25, 58] (voir aussi le chapitre 3), les graphes d'intervalles unitaires et les graphes circulaires unitaires dans [25] (voir aussi le chapitre 3 de cette thèse), *etc.* Dans ce chapitre, nous exhibons pour tout ordre un graphe G de grande compacité relativement à cet ordre. Nous notons par $\mathrm{IRS}(n)$ (respectively par $\mathrm{IRS}(n,d)$) la compacité « maximale » d'un graphe d'ordre n (respectively d'ordre n et de degré borné par d). La compacité maximale indique donc que tout graphe d'ordre n et de degré maximum d a une compacité $\leqslant \mathrm{IRS}(n,d)$ et, de plus, qu'il existe un graphe d'ordre n et de degré maximum d'au plus d possédant une compacité $\mathrm{IRS}(n,d)$. La compacité linéaire maximale n'a pas lieu d'être dans le cadre d'une étude en ordre de grandeur puisque, pour tout G, $\mathrm{IRS}(G) \leqslant \mathrm{LIRS}(G) \leqslant \mathrm{IRS}(G)+1$.

Nous avons vu d'après le lemme 3.13 du chapitre 3, que la compacité d'un graphe n'est pas, en général, une constante, et qu'il est même possible que $\mathrm{IRS}(n) = \Omega(\log n)$ (voir remarque suite au corollaire 3.6). Plus précisément, il est construit dans [58] un graphe planaire d'ordre n et de compacité $\Omega(\sqrt[3]{n})$. Dans [20], il a été prouvé que $\mathrm{IRS}(n) = \Omega(n/\log n)$. Il a été également montré qu'étant donné un graphe aléatoire $G \in \mathcal{G}_{n,p}$ (p étant la probabilité d'avoir une arête entre deux sommets du graphe), alors, avec haute probabilité[1] $\mathrm{IRS}(G) \geqslant \frac{1}{10} n^{1-6/\ln(np) - \ln(np)/\ln n}$. Les auteurs de [20] déduisent qu'il existe $p > 0$ tel qu'avec haute probabilité, $\mathrm{IRS}(G) \geqslant \frac{1}{20} n^{1-2\sqrt{6}/\sqrt{\ln n}} = \Omega(n/2^{\sqrt{\log n}})$ pour $G \in \mathcal{G}_{n,p}$. Dans ce chapitre, nous construisons pour tout n un graphe G d'ordre n tel que $\mathrm{IRS}(G) \geqslant n/12$, ce qui, en ordre de grandeur, est une borne inférieure optimale. De plus, nous étendons notre construction aux réseaux de degré maximum fixé pour prouver $\mathrm{IRS}(n,d) = \Omega(n/\log^2 n)$, pour tout $d \geqslant 3$.

1. Haute probabilité voulant dire avec probabilité $1 - o(1)$.

Dans la section suivante nous introduisons un outil qui va nous permettre de donner une borne inférieure de la compacité de graphe quelconque : la « compacité des matrices de contraintes ». Nous construisons ensuite des graphes appelés « graphes de contraintes » à partir de n'importe quelle matrice de contraintes. Dans la section 4.3, nous appliquons cette technique pour construire un graphe G d'ordre n, tel que $\mathrm{IRS}(G) \geqslant n/12$. Dans la section 4.4, nous montrons comment construire un graphe de degré borné à partir d'une matrice booléenne quelconque possédant une large compacité. Enfin, dans la section 4.5, nous étudions le cas des graphes planaires.

Une partie des résultats présentés dans ce chapitre ont été établis en collaboration avec Éric Guévremont et ont donné lieu à plusieurs publications internationales et rapports de recherches soumis à des revues internationales (voir [38, 39]).

4.2 Matrices de contraintes

Étant donné un graphe quelconque G, le calcul de sa compacité est généralement difficile, et à l'heure actuelle aucun algorithme en temps polynômial n'est connu. Le problème n'a pas été montré être NP-complet, cependant le problème de trouver la compacité de la fonction de routage par intervalles de chemins de coût minimum est NP-dur [19] pour les graphes dont les arêtes sont valuées par des coûts non uniformes. Dans cette section, nous décrivons un outil qui donne une borne inférieure de la compacité d'un graphe. Cet outil est basé sur la notion de *matrice de contraintes*, indépendamment introduite dans [20] et dans [39].

Considérons le sommet b du graphe G dessiné sur la partie gauche de la figure 3.1 page 15. Notez tout d'abord que les plus courts chemins de b aux sommets a, c, d, e et g sont uniques. Les plus courts chemins de b à a, c et d doivent utiliser l'arc (b,a), et les plus courts chemins de b à e et g doivent utiliser l'arc (b,e). Il y a un plus court chemin de b à f qui utilise l'arc (b,a), et un autre qui utilise (b,e). L'un ou l'autre de ces chemins peut être utilisé selon la fonction de routage retenue.

Définition 4.1 (matrice de contraintes) *Soit $G=(V,E)$ un graphe quelconque. Soient $\{v_i \mid 1 \leqslant i \leqslant p\} \subset V$ et $\{e_j=(x_j,y_j) \mid 1 \leqslant j \leqslant q\} \subset E$. Une* matrice de contraintes *de G est une matrice booléenne $p \times q$ $M=(m_{i,j})$ telle que, pour tout $i \in \{1,\ldots,p\}$, et pour tout $j \in \{1,\ldots,q\}$:*

$$\begin{cases} m_{i,j}=1 \Leftrightarrow \text{ tous les plus courts chemins de } x_j \text{ à } v_i \text{ dans } G \text{ utilise l'arc } e_j. \\ m_{i,j}=0 \Leftrightarrow \text{ aucun plus court chemin de } x_j \text{ à } v_i \text{ dans } G \text{ utilise l'arc } e_j. \end{cases}$$

Considérons une matrice de contraintes M d'un graphe G et supposons que les sommets de G ont été numérotés par une fonction de routage par intervalles de plus courts chemins $R=(\mathcal{L},\mathcal{I})$. S'il y a un 1 à l'intersection de la colonne correspondant à un arc (u,v) avec la ligne correspondant à un sommet w, alors $\mathcal{L}(w) \in I_{(u,v)}$. Au contraire s'il y a un 0 alors $\mathcal{L}(w) \notin I_{(u,v)}$ quelque soit R. Si les sommets v_i associés aux lignes de M, $1 \leqslant i \leqslant p$, sont ordonnés selon leurs numéros $\mathcal{L}(v_i)$, alors le nombre de 1 consécutifs dans une colonne de M est clairement une borne inférieure de la compacité de R.

La figure 4.1 montre un exemple de matrice de contraintes pour le graphe de Petersen. Bien entendu, il existe d'autre matrices de contraintes pour ce graphe, et certaines sont de plus grandes dimensions. La figure 4.2 montre un exemple où la matrice M_p de la figure 4.1 est une matrice de contraintes d'un autre graphe.

$$M_p = \begin{array}{c} \begin{array}{ccc} e_1 & e_2 & e_3 \end{array} \\ \boxed{\begin{array}{ccc} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}} \end{array} \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array}$$

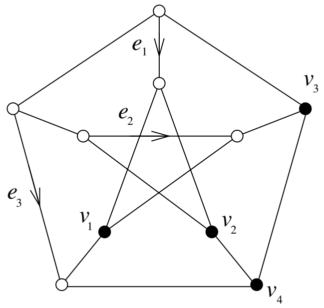


FIG. 4.1 – *Une matrice de contraintes pour le graphe de Petersen.*

Notez qu'une matrice obtenue par permutation des lignes et des colonnes d'une matrice de contraintes est aussi une matrice de contraintes pour le même graphe et correspond simplement à une renumérotation des sommets. Si nous pouvons montrer que, pour toute permutation des lignes et des colonnes d'une matrice de contraintes d'un graphe G, il existe au moins une colonne de la matrice possédant un nombre k de blocs de 1 consécutifs, alors toute fonction de routage par intervalles de plus courts chemins sur G sera de compacité au moins k. Puisque, par définition, la compacité d'un graphe est la plus petite compacité des fonctions de routage par intervalles sur G, on en déduit facilement que ce même k est une borne inférieure pour la compacité de G, et donc que $\mathrm{IRS}(G) \geqslant k$. Nous formalisons ci-dessous cette notion.

Définition 4.2 (compacité d'une matrice booléenne) *Soit M une matrice booléenne. La* compacité de M, *notée* $\mathrm{I}(M)$, *est le minimum pris sur toutes les permutations des lignes de M du maximum pris sur toutes les colonnes de la matrice permutée du nombre de blocs cycliques de* 1 *consécutifs.*

Un bloc de 1 consécutifs est dit *cyclique* si le premier et le dernier bit d'un mot formé par cette colonne sont tout égaux à 1. La figure 4.1 donne l'exemple d'une matrice M_p de compacité 2. En effet, quelque soit la permutation des lignes de M_p, il existe toujours une colonne comportant deux blocs de 1 consécutifs. Nous avons le résultat suivant dû à Flammini [16] et prouvé dans [20] et dans [39].

Lemme 4.1 *Pour toute matrice de contraintes M d'un graphe G,* $\mathrm{IRS}(G) \geqslant \mathrm{I}(M)$.

Malheureusement nous ne connaissons pas de meilleure relation entre la compacité d'un graphe et celle de sa matrice de contraintes la « moins compacte ». Par exemple il est montré en annexe 4.A que le graphe de la figure 3.1 page 15 a une compacité 2 alors que toutes les matrices de contraintes de ce graphe ont une compacité au plus 1.

Remarques. Dans [19], le problème de calculer la compacité d'une matrice booléenne quelconque a été montré NP-dur, — résultat dérivé du problème CONSECUTIVE ONES SUBMATRIX prouvé NP-complet dans [34] — tout comme le problème de calculer la compacité des graphes avec des poids non uniformes sur les arêtes. Le problème reste cependant non résolu pour le cas des poids uniformes. De même le problème de savoir si un graphe supporte une fonction de routage par intervalles de plus courts chemins de compacité k fixée est toujours ouvert. Il est intéressant de remarquer que Booth et Lueker [8] ont donné un algorithme en temps $O(pq)$ pour déterminer si une matrice booléenne $p \times q$ a une compacité 1 (ce problème est référencé dans la littérature sous le nom de « *circular ones property* » dans [33, 74]). Le problème de la caractérisation des graphes de compacité 1 reste cependant un problème non résolu.

Dans la suite, nous allons représenter une matrice booléenne M comme un ensemble de mots binaires, chaque ligne ou colonne représentant un mot binaire. Ainsi la distance de Hamming entre deux lignes quelconques de M est le nombre de bits sur lesquelles elles diffèrent.

Définition 4.3 (rayon d'une matrice booléenne) *Soit M une matrice booléenne. Pour toutes lignes l_i et l_j de M on note* $\mathrm{Ham}(l_i, l_j)$ *la distance de Hamming entre les mots binaires représentés par l_i et l_j. Le* rayon de M *est l'entier r vérifiant :*

$$r = \min_{i \neq j} \mathrm{Ham}(l_i, l_j)$$

Par exemple la matrice M_p de la figure 4.1 a un rayon 2. Si toutes les lignes de M sont distinctes deux à deux (c'est-à-dire M a un rayon non nul), alors M peut être vue comme la représentation d'un *code* binaire c'est-à-dire une suite ordonnée de mots binaires. Le rayon d'une matrice est donc la plus petite distance entre deux mots d'un code de représentation M. Par exemple les codes de Gray de longueur q peuvent être représentés par une matrice booléenne $2^q \times q$ de rayon 1.

Lemme 4.2 *Pour toute matrice $p \times q$ booléenne M de rayon r,* $\mathrm{I}(M) \geqslant \dfrac{pr}{2q}$

Preuve. Considérons une matrice $p \times q$ booléenne M de rayon r. Pour tout $i \in \{1, \ldots, p\}$, soit d_i la distance de Hamming entre deux lignes consécutives de M, i et $i+1$ (modulo p, la dernière et la première ligne sont consécutives). Pour tout $j \in \{1, \ldots, q\}$, soit k_j le nombre de blocs de 1 consécutifs de la colonne j de M. Soit $D = \sum_{i=1}^{p} d_i$ la distance Hamming totale sur toutes les lignes de M, et soit $K = \sum_{j=1}^{q} k_j$ le nombre total de blocs de 1 consécutifs de M. Il est facile de voir que chaque bloc de 1 consécutifs accroît de 2 la somme des distances de Hamming sur toutes les lignes de M, c'est-à-dire $2K = D$. Puisque $d_i \geqslant r$ pour tout i, nous avons $2 \sum_{j=1}^{q} k_j = \sum_{i=1}^{p} d_i \geqslant pr$. Par le principe des pigeons, puisque M a q colonnes, il y doit avoir une colonne j, $1 \leqslant j \leqslant q$, telle que $k_j \geqslant pr/(2q)$. □

Remarque. Si $r = \Theta(q)$ alors $\mathrm{I}(M) = \Theta(p)$ et, de plus, chaque colonne de M nécessite $\Theta(p)$ blocs de 1 consécutifs, car chaque colonne ne peut posséder qu'au plus p blocs de 1 consécutifs. Le lemme 4.2 établit donc dans ce cas que $\mathrm{I}(M) = \Omega(p)$.

Définition 4.4 *Pour tout entier $q \geqslant r \geqslant 1$, on note $A(q,r)$ le nombre maximum de mots binaires de longueur q deux à deux distants d'une distance de Hamming au moins r.*

$A(q,r)$ est une fonction que l'on rencontre souvent en théorie des codes. La valeur $A(q,r)$ peut être vue comme le plus grand nombre de lignes d'une matrice booléenne de q colonnes et de rayon r. De manière générale, la valeur exacte de $A(q,r)$ est inconnue et reste un problème ouvert fondamental de la théorie des codes. Cependant, il existe de nombreuses tables encadrant $A(q,r)$ (voir [64] par exemple).

Corollaire 4.1 *Pour tout entier $k \geqslant 0$, s'il existe des entiers p, q et r tels que $A(q,r) \geqslant p > \dfrac{2q(k-1)}{r}$ alors il existe une matrice booléenne M telle que $\mathrm{I}(M) \geqslant k$.*

Preuve. Supposons que $A(q,r) \geqslant p > 2q(k-1)/r$ pour des entiers k, p, q et r. Puisque $p \leqslant A(q,r)$ il existe, par définition de $A(q,r)$, une matrice $p \times q$ booléenne M de rayon r. En appliquant le lemme 4.2, on obtient $\mathrm{I}(M) \geqslant pr/(2q) \geqslant k$. $\square$

Lemme 4.3 *Pour tout $\delta \geqslant 1$, il existe une matrice $2^{\delta+2} \times 2^{\delta+1}$ booléenne M_δ, telle que $\mathrm{I}(M_\delta) \geqslant 2^\delta$.*

Preuve. Nous construisons une suite de matrices M_δ par récurrence sur δ. Pour toute matrice booléenne M, on note $\overline{M}$ la matrice obtenue par complémentation de tous les bits de M. De plus si $M = [\,X\,Y\,]$, où X et Y sont deux matrices de mêmes dimensions, on note $\chi(M) = [\,Y\,X\,]$, c'est-à-dire la matrice obtenue à partir de M en échangeant les colonnes de X avec celles de Y. Nous définissons $M_{\delta+1}$ comme une matrice par blocs composée des matrices M_δ, $\chi(M_\delta)$ et $\chi(\overline{M_\delta})$. M_1 et $M_{\delta+1}$, pour tout $\delta \geqslant 1$, sont décrites par l'équation 4.1.

$$M_1 = \begin{bmatrix} 0&0&0&0\\ 0&0&1&1\\ 1&1&1&1\\ 1&1&0&0\\ 0&1&0&1\\ 1&0&0&1\\ 1&0&1&0\\ 0&1&1&0 \end{bmatrix} \qquad \chi(M_1) = \begin{bmatrix} 0&0&0&0\\ 1&1&0&0\\ 1&1&1&1\\ 0&0&1&1\\ 0&1&0&1\\ 0&1&1&0\\ 1&0&1&0\\ 1&0&0&0 \end{bmatrix} \qquad M_{\delta+1} = \begin{bmatrix} M_\delta & M_\delta \\ \chi(M_\delta) & \chi(\overline{M_\delta}) \end{bmatrix} \tag{4.1}$$

Prouvons par récurrence que le rayon de M_δ est au moins 2^δ. On peut tout d'abord vérifier que le rayon de M_1 est 2. Aussi, pour toute matrice M de rayon r, le rayon de $\chi(M)$ et de $\chi(\overline{M})$ sont égaux. Par hypothèse, le rayon de M_δ de dimensions $2^{\delta+2} \times 2^{\delta+1}$ vaut $r = 2^\delta$. Le rayon de $A = [\,M_\delta\,M_\delta\,]$ est $2r$ tout comme celui de $B = [\,\chi(M_\delta)\,\chi(\overline{M_\delta})\,]$, *i.e.* toute paire de lignes des matrices A ou B diffèrent sur $2r$ bits. Vérifions également que toute ligne u de A diffère sur au moins $2r$ bits de toute ligne v de B. Les lignes u et v peuvent être vues comme des mots binaires de longueur $2^{\delta+2} = 4r$. Posons $u = x\,y\,x\,y$, où x et y sont des mots binaires de longueur r. Par construction $v = y\,x\,\overline{y}\,\overline{x}$. Si x et y diffèrent sur d bits, alors u et v diffèrent sur $d + d + (r-d) + (r-d) = 2r$ bits (x et $\overline{y}$ ainsi que y et $\overline{x}$ diffèrent sur $r-d$ bits). Donc, toute paire de lignes de $M_{\delta+1}$ diffèrent par au moins $2r$ bits, donc $M_{\delta+1}$ a un rayon sur au moins $2r = 2^{\delta+1}$. Par application du lemme 4.2, $\mathrm{I}(M_{\delta+1}) \geqslant 2^{\delta+3} 2r/(2 \cdot 2^{\delta+2}) = 2^{\delta+1}$. $\square$

Remarque. Il n'est pas difficile d'établir que la compacité de M_δ vaut exactement 2^δ en considérant la construction récurrente de l'équation 4.1 dans la preuve du lemme 4.3. En fait, la matrice M_δ telle que décrite dans l'équation 4.1 représente un code binaire avec un nombre « équilibré » de changements de dimensions entre chaque position de bit. Dans chaque colonne de M_δ, il y a exactement $2^{\delta+1}$ changements de bits. C'est une propriété importante de certains codes binaires utilisés dans les tests des circuits booléens. Par exemple, il est décrit dans [72] des codes de Gray (donc de rayon 1) équilibrés qui n'admettent pas une construction immédiate — par exemple il existe un seul code de Gray équilibré de 4 bits. Les matrices M_δ fournissent donc des codes binaires équilibrés de longueur $2^{\delta+1}$ bits et de rayon 2^δ. Ces codes sont de cardinalité maximale puisque le nombre de mots de ces codes est $2^{\delta+2} = A(2^{\delta+1}, 2^\delta)$.

Quoiqu'il en soit, le lemme 4.3 prouve qu'il existe des matrices $\Theta(p) \times \Theta(p)$ de compacité $\Theta(p)$, ce qui va nous permettre de construire des graphes de « grande » compacité relativement à leur ordre.

4.3 Un pire cas pour le routage par intervalles

L'idée des matrices de contraintes à été indépendamment introduite dans un rapport de Flammini, van Leeuwen et Spaccamela [20], où les auteurs emploient le terme de « matrice unique » dans une représentation des plus courts chemins par matrices booléennes. Avec ce concept, ils ont construit un graphe d'ordre n de compacité $\Omega(n/\log n)$. Dans la suite, nous étendons ce concept à l'idée des *graphes de contraintes*, un outil plus puissant, pour améliorer cette borne inférieure, c'est-à-dire trouver un graphe ayant une compacité supérieure.

4.3.1 Graphes de contraintes

Nous présentons ci-dessous comment construire un graphe G à partir d'une matrice booléenne M quelconque de façon à ce que M soit une matrice de contraintes de G. Un classe de graphes ayant la propriété que toute matrice booléenne $p \times q$ est une matrice de contraintes d'au moins un des graphes de cette classe est appelée classe de *graphes de contraintes*.

Lemme 4.4 *Pour toute matrice $p{\times}q$ booléenne M, il existe un graphe d'ordre $p+2q$ possédant M comme matrice de contraintes.*

Preuve. Soit $M = (m_{i,j})$ une matrice $p \times q$ booléenne. On construit un graphe G composé de deux niveaux. Le premier niveau est composé de p sommets isolés $\{v_1, \ldots, v_p\}$. Le second niveau consiste en q copies de K_2. On note a_j et b_j les deux sommets de la j-ième copie de K_2, $1 \leqslant j \leqslant q$. L'ensemble des sommets de G est donc $\{v_1, \ldots, v_p, a_1, b_1, \ldots, a_q, b_q\}$. G a $p + 2q$ sommets.

Les deux niveaux sont connectés comme suit : si $m_{i,j} = 1$, on ajoute l'arête $\{b_j, v_i\}$ et sinon l'arête $\{a_j, v_i\}$. G a donc en tout $pq + q$ arêtes. G est connexe et a un diamètre au plus 6 si une des colonnes de M est composé uniquement de 0 ou uniquement de 1, et de diamètre 4 sinon.

Prouvons que M est une matrice de contraintes pour G. Soit $e_j = (a_j, b_j)$ pour tout $j \in \{1, \ldots, q\}$. Pour tout $i \in \{1, \ldots, p\}$ et pour tout $j \in \{1, \ldots, q\}$, tous les plus courts chemins de a_j à v_i utilise e_j si et seulement si l'arête $\{b_j, v_i\}$ existe, *i.e.* $m_{i,j} = 1$. Inversement, aucun plus court chemin de a_j à v_i utilise e_j si et seulement si l'arête $\{a_j, v_i\}$ existe, *i.e.* $m_{i,j} = 0$. Par conséquent M est une matrice de contraintes de G. □

Notation. Pour tout graphe G nous posons $\Delta(G)$ le degré maximum G.

Le lemme 4.4, appliqué à la matrice 4×3 M_p de la figure 4.1, permet d'obtenir un graphe de $4+2\cdot 3=10$ sommets qui est précisément le graphe de Petersen.

Remarque. Soit M une matrice booléenne $p\times q$, et soit G le graphe de contraintes de M construit comme dans la preuve du lemme 4.4. Il est aisé de voir que, dans ce cas, $\Delta(G)\geqslant \max\{z/q, p-z/q\}$, où z est le nombre total de 0 de la matrice M (considérer une arête quelconque $\{a_j,b_j\}$ de la construction).

4.3.2 Borne inférieure de $\mathrm{IRS}(n)$

Lemme 4.5 *Soit $N(k)$ l'ordre du plus petit graphe de compacité au moins k. Pour tout $k\geqslant 2$,*

$$N(k)\leqslant 2^{\lceil \log_2 k\rceil+2}+4k-3=O(k).$$

La valeur $N(k)$ peut être un paramètre important à considérer en pratique. En effet, on peut garantir que tous les réseaux d'ordre strictement inférieur à $N(k)$ utilisent, sur tous leurs nœuds, un routeur permettant au plus k intervalles par lien.

Preuve. Soit $k\geqslant 2$. D'après le corollaire 4.1, il existe une matrice booléenne $p\times q$ de rayon r dès que $A(q,r)\geqslant p>2q(k-1)/r$. Soit M cette matrice. Le lemme 4.4 garantit l'existence d'un graphe de contraintes $G=(V,E)$ de la matrice M et, par application du lemme 4.1, tel que $\mathrm{IRS}(G)\geqslant \mathrm{I}(M)\geqslant k$. Donc, l'ordre de G est $p+2q$. C'est une borne supérieure pour $N(k)$. Par conséquent

$$\forall k\geqslant 2,\quad N(k)\leqslant \min_{p,q,r}(p+2q),\qquad \text{où } (p,q,r) \text{ vérifient } A(q,r)\geqslant p>\frac{2q(k-1)}{r}.$$

Soit $\delta=\lceil \log_2 k\rceil\geqslant 1$. En utilisant la matrice M_δ du lemme 4.3, de $p=2^{\delta+2}$ lignes, $q=2^{\delta+1}$ colonnes et de rayon $r=2^\delta$, nous avons $A(q,r)\geqslant p>2q(k-1)/r$ si et seulement si $2^{\delta+2}\geqslant p>4(k-1)$. En effet, le lemme 4.3 montre que $A(2^{\delta+1},2^\delta)\geqslant 2^{\delta+2}$ et, en appliquant la borne de Plotkin [64] qui établi que $A(2i,i)\leqslant 4i$ pour i pair, nous obtenons finalement que $A(2^{\delta+1},2^\delta)=2^{\delta+2}$. Comme p doit être le plus petit entier tel que $2^{\delta+2}\geqslant p>4(k-1)$, nous pouvons choisir $p=4k-3$ (et nous pouvons donc supprimer au moins $3\leqslant 2^{\delta+2}-(4k-3)$ lignes de la matrice M_δ, pour n'en garder que $4k-3$ avant la construction du graphe de contraintes). D'où $N(k)\leqslant p+2q=4k-3+2^{\lceil \log_2 k\rceil+2}$. □

Remarque. L'exactitude de la borne supérieure de $N(k)$ est amoindrie pour les valeurs de k qui ne sont pas des puissances entières de 2. Cela est dû au lemme 4.3 qui impose que les dimensions des matrices de grand rayon soient des puissances entières de 2. En réalité, MacWilliams et Sloane ont prouvé dans [64] que des matrices booléennes $8\delta\times 4\delta$ de rayon 2δ existent s'il existe des matrices d'*Hadamard* de dimensions $8\delta\times 4\delta$. D'après eux, les matrices d'Hadamard étaient déjà toutes connues en 1977, pour tout $\delta\leqslant 70$.

Le lemme 4.5 est suffisant pour prouver que $N(k)=\Theta(k)$ pour tout $k\geqslant 2$, car $N(k)\geqslant 2k+1$. En effet nous avons vu précédemment dans le théorème 3.11 que $\mathrm{IRS}(n)\leqslant (n-1)/2$. Le théorème suivant est une conséquence directe du lemme 4.5.

Théorème 4.1 *Pour tout entier $n\geqslant 2$, $\mathrm{IRS}(n)\geqslant \dfrac{n}{12}$.*

C'est le résultat fondamental de ce chapitre.

Preuve. Puisque pour tout $k \geqslant 2$, $2^{\lceil \log_2 k \rceil} \leqslant 2(k-1)$, nous avons donc d'après le lemme 4.5 $N(k) \leqslant 12k - 11$. Par définition de $N(k)$, nous déduisons que, pour tout entier $n \geqslant 2$, il existe un graphe G d'ordre n (pour $2 \leqslant n < 13$ on choisit pour G n'importe quelque graphe d'ordre n) tel que $\mathrm{IRS}(G) \geqslant k \geqslant \lfloor (n+11)/12 \rfloor$. Par conséquent $\mathrm{IRS}(G) \geqslant n/12$. □

Remarques.

- Nous pouvons remarquer que le graphe G construit dans la preuve du lemme 4.5 n'a pas un degré maximum borné. G a été obtenu par la construction générale du lemme 4.4, en utilisant une matrice booléenne M_δ qui contient autant de 0 que de 1. Donc, en appliquant la remarque de la section 4.3.1, nous pouvons déduire que $\Delta(G) \geqslant 2^\delta + 1/2$. En effet, nous effaçons certaines lignes de la matrice M_δ avant la construction de G, mais en en conservant au moins $2^{\delta+2}/2+1$. Il reste donc au moins $2^{\delta+1}(2^{\delta+2}/2+1)/2$ 0 ou 1. Notons z ce nombre. Puisque l'ordre de G est $n = 2^{\delta+2} + 4k - 3$ et que k satisfait $2^{\delta-1} < k \leqslant 2^\delta$, nous obtenons $\Delta(G) \geqslant z/q \geqslant n/8$.
- Il est bien évident, d'après la remarque du lemme 4.2, que p routeurs du graphe de contraintes construit à partir des matrices $p \times q$ $M_\delta = M_{\Theta(k)}$ nécessitent $\Theta(k)$ intervalles sur un de leurs liens. Autrement dit, la compacité maximale $n/12$ est atteinte sur $\Theta(n)$ liens différents de $\Theta(n)$ routeurs. Nous verrons aux chapitres suivants que cela peut être important pour la recherche de la mémoire de routage globale d'un réseau.

Montrons maintenant qu'il est encore possible d'affiner la borne supérieure de $N(k)$, ce qui revient à affiner la borne inférieure de la compacité maximale d'un graphe d'ordre n. Pour k puissance entière de 2, nous obtenons $N(k) \leqslant 8k-3$ d'après le lemme 4.5. Si nous considérons le cas $k = 2$ par exemple, la borne supérieure de $N(2)$ du lemme 4.5 est obtenue en utilisant une matrice 5×4 de rayon 2. En effet $r = 2^{\lceil \log_2 2 \rceil} = 2$, $q = 2r = 4$ et $p = 4k - 3 = 5$. Donc, la borne du lemme 4.5 établit que $N(2) \leqslant p + 2q = 5 + 2 \cdot 4 = 13$. Cependant, si à la place d'une telle matrice nous avions utilisé la matrice M_p 4×3 de rayon 2 de la figure 4.1, nous aurions alors trouvé que $N(2) \leqslant p + 2q = 4 + 2 \cdot 3 = 10$ en appliquant le lemme 4.1. Un autre exemple d'amélioration peut être mis en évidence pour $k = 5$. La borne du lemme 4.5 donne $N(5) \leqslant 49$ par utilisation d'une matrice 17×6 de rayon 3. Cependant, avec une autre matrice 25×6 de rayon 2 (le lecteur est invité à se convaincre lui-même qu'une telle matrice existe) nous aurions pu trouver $N(5) \leqslant p + 2q = 25 + 2 \cdot 6 = 37$. En conséquence, le lemme 4.5 ne fournit pas un résultat exact (bien qu'en ordre de grandeur il soit optimal) et peut être amélioré.

Nous pouvons calculer une meilleure borne supérieure de $N(k)$ en utilisant la construction du graphe de contraintes du lemme 4.5. Cette construction est basée sur le fait que $N(k) \leqslant \min_{p,q,r}(p + 2q)$ tel que $A(q,r) \geqslant p > 2q(k-1)/r$. En tenant compte des meilleures bornes inférieures connues de $A(q,r)$ [64], nous calculons par une recherches systématiques les différentes bornes supérieures de $N(k)$, pour $2 \leqslant k \leqslant 21$ (voir la table 4.1). Cette table donne également les valeurs p, q et r des matrices $p \times q$ de rayon r utilisées pour la construction du graphe de contraintes. Le graphe correspondant à la première ligne de la table, pour $k = 2$, utilisant une matrice 4×3 de rayon 2 et donnant $N(2) \leqslant 10$, illustré sur dans la figure 4.1 : il s'agit en fait du graphe de Petersen construit à partir de la matrice M_p.

Nous avons montré, par utilisation de techniques issues de la théorie des codes, que nous pouvons obtenir une approximation exacte de l'ordre minimum d'un graphe nécessitant au moins k intervalles pour router avec une fonction de routage par intervalles de plus courts

k	d	q	p	borne sup. de $N(k)$
2	2	3	4	**10**
3	2	5	11	**21**
4	2	5	16	**26**
5	2	6	25	**37**
6	2	6	31	**43**
7	4	10	31	**51**
8	4	10	36	**56**
9	6	14	38	**66**
10	6	14	43	**71**
11	6	14	47	**75**

k	d	q	p	borne sup. de $N(k)$
12	6	14	52	**80**
13	6	14	57	**85**
14	6	14	61	**89**
15	8	18	64	**100**
16	6	15	76	**106**
17	6	15	81	**111**
18	6	15	86	**116**
19	6	15	91	**121**
20	6	15	96	**126**
21	6	15	101	**131**

TAB. 4.1 – *Bornes supérieures de $N(k)$ pour les petites valeurs de k.*

chemins. Néanmoins, nous pensons que cette borne peut être améliorée si l'on s'intéresse aux valeurs des constantes. Par exemple, considérons encore une fois le cas $k = 2$. L'application du résultat général du lemme 4.5 a montré que $N(2) \leqslant 10$. Cependant il a été montré dans une analyse de cas qu'il existe un graphe d'ordre 7 de compacité 2 (voir le graphe de la figure 4.2 possédant M_p de compacité 2 comme matrice de contraintes). En annexe 4.B nous prouvons en fait que $N(2) = 7$. Bien qu'asymptotiquement optimale, il existe encore un petit écart entre notre borne supérieure et la valeur exacte de $N(k)$.

$$M_p = \begin{array}{c} \\ \end{array} \begin{array}{ccc} (b,g) & (d,g) & (f,g) \\ \end{array}$$
$$M_p = \left[\begin{array}{ccc} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right] \begin{array}{c} g \\ e \\ a \\ c \end{array}$$

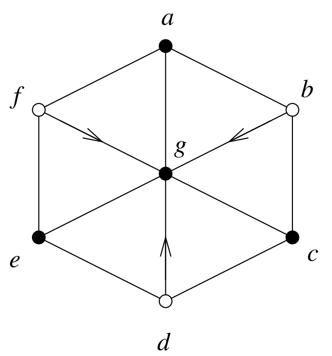


FIG. 4.2 – *La matrice de contraintes M_p d'un graphe G avec* $\mathrm{IRS}(G) = \mathrm{I}(M_p) = 2$.

4.4 Compacité maximale pour les réseaux de degré borné

À la vue de l'étude de la section précédente, il semble que la borne inférieure de la compacité maximale d'un graphe d'ordre n soit atteinte pour un réseau de degré non borné, plus exactement de degré au moins $n/8$ (voir la remarque du théorème 4.1). Cependant, dans la pratique, il n'est pas raisonnable de supposer un nombre de liens par routeur aussi important.

4.4.1 Graphes de contraintes de degré borné

Dans cette section, nous réutilisons les outils développés dans la section 4.2. Nous construisons un graphe de contraintes de degré borné à partir d'une matrice booléenne M quelconque. Notez que le problème de calculer la compacité des graphes de degré au plus 2 est direct puisque de tels graphes ont tous une compacité 1 (voir la section 3.6 du chapitre 3). Donc $\mathrm{IRS}(n,2)=1$ pour tout $n \geqslant 2$.

Lemme 4.6 *Pour toute matrice $p \times q$ booléenne M, $p \geqslant 4$ et $q \geqslant 3$, et tout entier $d \geqslant 3$, il existe un graphe de degré maximum d, d'ordre au plus $6pq-4(p+q)$ et possédant M comme matrice de contraintes.*

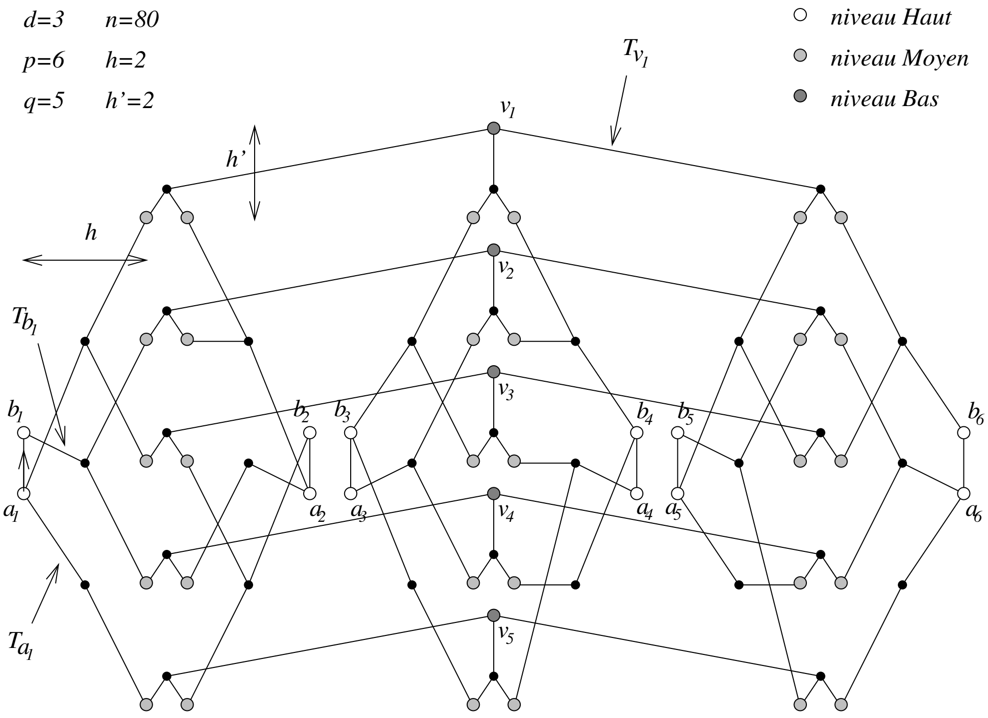
Construction. Nous nous référons à la figure 4.3 pour une illustration de la construction. Soit une matrice $p\times q$ booléenne $M=(m_{i,j})$ et un entier $d \geqslant 3$. Le graphe G que nous construisons afin de satisfaire le lemme 4.6 est composé de trois niveaux de sommets notés Bas, Moyen et Haut, et de sommets intermédiaires (dessinés avec des petits points noirs sur la figure 4.3). Le niveau Bas est un ensemble de p sommets indépendants que nous notons v_i, $1 \leqslant i \leqslant p$. Le niveau Haut est constitué de q copies de K_2 dont les sommets sont notés a_j, b_j, $1 \leqslant j \leqslant q$. Le niveau Moyen est composé de pq sommets indépendants désignés par $w_{i,j}$, $1 \leqslant i \leqslant p$ et $1 \leqslant j \leqslant q$.

Ces trois niveaux sont interconnectés par des arbres de degré maximum d utilisant des sommets intermédiaires. Décrivons tous d'abord comment sont connectés les niveaux Bas et Moyen. À tout sommet v_i du niveau Bas, nous enracinons un arbre T_{v_i} dont les feuilles sont les sommets $w_{i,j}$, $j \in \{1,\ldots,q\}$. Tous ces arbres sont isomorphes. Ils sont non orientés, de degré maximum d et possèdent exactement q feuilles. Dans chaque T_{v_i}, $i \in \{1,\ldots,p\}$, les feuilles $w_{i,j}$, $j \in \{1,\ldots,q\}$, sont toutes à distance $\Theta(\log q)$ de leur racine v_i.

Pour connecter les niveaux Moyen et Haut, nous utilisons un second type d'arbre. Aux sommets a_j (respectively b_j), $j \in \{1,\ldots,q\}$, nous enracinons un arbre $(d-1)$-aire T_{a_j} (respectively T_{b_j}). Dans de tels arbres, la racine à un degré au plus $d-1$ tandis que les autres sommets ont un degré au plus d. Dans T_{a_j} (respectively T_{b_j}), $j \in \{1,\ldots,q\}$, les feuilles sont les sommets $w_{i,j}$, pour tout i tel que $m_{i,j}=0$ (respectively $m_{i,j}=1$). Ces arbres ont donc au plus p feuilles. Nous imposons de plus que chaque feuille de l'arbre T_{a_j} (respectively T_{b_j}) soit à une même distance $\Theta(\log p)$ de a_j (respectively b_j). T_{a_j} et T_{b_j} possèdent le plus petit nombre de sommets nécessitant toutes ces contraintes.

En fait, les sommets $w_{i,j}$ peuvent être vu comme formant une grille où ceux de la ligne i sont connectés par un arbre T_{v_i}, certains sommets des colonnes sont connectés par un arbre T_{a_j} et d'autres sont connectés par un arbre T_{b_j}.

Les arbres T_{a_j} et T_{b_j} peuvent être construit de la manière suivante : partant des feuilles (au plus p), construire un arbre (d–1)-aire, en ajoutant, si nécessaire et à chaque niveau, des nœuds intermédiaires. De la racine de cet arbre, construire un chemin jusqu'à a_j (ou b_j), de manière à ce que les arbres T_{a_j} et T_{b_j}, soient de hauteur $\lceil \log_{d-1} \max_j(\max\{z_j, p-z_j\}) \rceil$ où z_j est le nombre de zéros de la j-ième colonne de M. Pour construire T_{v_i}, nous commençons à de la racine v_i avec d fils (ou q fils si $q \leqslant d$) et nous fixons à chaque fils un arbre (d–1)-aire complets de hauteurs $\lceil \log_{d-1}(q/d) \rceil$ chacun. Ensuite, nous supprimons les $d(d-1)^{\lceil \log_{d-1}(q/d)\rceil - 1} - q$ feuilles éventuellement en trop du dernier étage. Le graphe obtenu est connexe et de degré maximum d.



$$
\begin{array}{c|cccccc|c}
 & (a_1,b_1) & (a_2,b_2) & (a_3,b_3) & (a_4,b_4) & (a_5,b_5) & (a_6,b_6) & \\
\hline
 & 0 & 0 & 1 & 1 & 0 & 1 & v_1 \\
 & 1 & 0 & 0 & 1 & 1 & 0 & v_2 \\
M = & 0 & 1 & 1 & 0 & 0 & 1 & v_3 \\
 & 1 & 0 & 0 & 1 & 0 & 0 & v_4 \\
 & 0 & 1 & 1 & 0 & 1 & 0 & v_5 \\
\hline
\end{array}
$$

FIG. 4.3 – *Un graphe de contraintes de matrice M avec 80 sommets et un degré maximum 3.*

Preuve. Considérons la construction précédente du graphe G à partir de M. Soient $i \in \{1,\ldots,p\}$ et $j \in \{1,\ldots,q\}$. Prouvons que M est une matrice de contraintes de G. Soit $e_j = (a_j, b_j)$, soit A_j (respectively B_j) l'ensemble des sommets v_k tels que $w_{k,j}$ soit une feuille de l'arbre T_{a_j} (respectively T_{b_j}), pour tout $k \in \{1,\ldots,p\}$. Clairement les arbres T_{a_j} et T_{b_j} étant disjoints, les ensembles A_j et B_j partitionnent les sommets du niveau Bas.

Supposons que $v_i \in A_j$. Le plus court chemin de a_j à v_i doit utiliser un arc de T_{a_j}, et donc ne doit pas utiliser l'arc e_j. En effet, il est facile de constater qu'il y a un chemin de a_j à v_i de longueur $h + h'$ avec $h = \lceil \log_{d-1} \max_j(\max\{z_j, p - z_j\}) \rceil$ où z_j est le nombre de zéros de la j-ième colonne de M et $h' = 1 + \lceil \log_{d-1}(q/d) \rceil$: cela prend h étapes pour descendre l'arbre T_{a_j} vers le sommet $w_{i,j}$, puis h' étapes pour monter dans l'arbre T_{v_i} jusqu'à v_i. En appliquant un argument similaire, on peut vérifier que tout chemin de a_j à v_i utilisant l'arc e_j doit avoir une longueur au moins $1 + h + 2 + 2 + h'$. Donc, si $v_i \in A_j$ alors aucun plus court chemin de a_j vers v_i n'utilise e_j. Par le même type de raisonnement, on établit facilement que si $v_i \in B_j$, tout plus court chemin de a_j vers v_i utilise l'arc e_j. Or si $v_i \in B_j$ alors, par définition de l'ensemble B_j, le sommet $w_{i,j}$ est une feuille de T_{b_j}. Par construction de G, ceci est vrai si et seulement si $m_{i,j} = 1$. De manière similaire, $v_i \in A_j$ si et seulement si $m_{i,j} = 0$. Ceci prouve que M est une matrice de contraintes de G.

Comptons le nombre de sommets de G. Considérons d'abord T_{v_i}, le plus petit arbre avec q feuilles à égale distance de la racine et ayant un degré maximum d (T_{v_i} diffère d'un arbre (d–1)-aire par le fait que sa racine a un degré d, et non $d-1$). On note par $|T_{v_i}|$ son ordre. Cet arbre a au plus $d(d-1)^{k-1}$ sommets à distance k de la racine, et donc $|T_{v_i}| \leqslant 1+d\sum_{k=0}^{h'-2}(d-1)^k+q$. En prenant $h' = 1 + \lceil \log_{d-1}(q/d) \rceil$ et $d = 3$, alors $|T_{v_i}| \leqslant 3{\cdot}2^{h'-1}+q-2$ (l'ordre est maximum pour $d = 3$). Notez que pour obtenir une matrice de compacité $\geqslant 2$ il faut avoir $p \geqslant 4$ et $q \geqslant 3$. Le lecteur pourra vérifier le fait qu'il n'y a pas de matrice booléenne plus petite de compacité $\geqslant 2$. Dans ce cas, puisque $3 \leqslant q \leqslant 3{\cdot}2^{\lceil \log_2(q/3) \rceil} \leqslant 2(q-1)$, il découle que $|T_{v_i}| \leqslant 3q-4$.

Nous avons ensuite maintenant de borner la valeur de $|T_{a_j}| + |T_{b_j}|$, c'est-à-dire le nombre total de sommets des arbres T_{a_j} et T_{b_j}. T_{a_j} est le plus petit arbre $(d-1)$-aire de hauteur $h = \lceil \log_{d-1} p \rceil$ avec z feuilles, chacune à distance h de la racine, où $z = \max_j z_j$ ($z \leqslant p$) z_j étant le nombre de zéros de la j-ième colonne de M. Dans un arbre $(d-1)$-aire il y a au plus $(d-1)^k$ sommets au niveau k (k étant la distance du niveau à la racine) et donc $p \leqslant (d-1)^h$ (on peut supprimer des feuilles au dernier niveau). La hauteur des arbres T_{a_j} est $h_{a_j} = \lceil \log_{d-1} z_j \rceil$ ou 1 si $z_j = 0$. La hauteur de l'arbre T_{b_j} est $h_{b_j} = \lceil \log_{d-1}(p-z_j) \rceil$ ou 1 si $z_j = p$. Notez que $h_{a_j} + h_{b_j} \geqslant 2$. Ainsi $|T_{a_j}| \leqslant \sum_{k=0}^{h_{a_j}-1} (d-1)^k + z_j + h - h_{a_j}$ où $h - h_{a_j}$ est la longueur du chemin ajouté de la racine de l'arbre complet à a_j. De manière similaire, $|T_{b_j}| \leqslant \sum_{k=0}^{h_{b_j}-1} (d-1)^k + p - z_j + h - h_{b_j}$. Ces sommes sont maximum pour d minimum. Soit $d = 3$. Si $1 < z_j < p$, nous avons $|T_{a_j}|+|T_{b_j}| \leqslant 2^{\lceil \log_2 z_j \rceil} + 2^{\lceil \log_2(p-z_j) \rceil} - 2 + p + 2\lceil \log_2 p \rceil - 2$. Puisque $\log_2 p \leqslant p/2$, $|T_{a_j}| + |T_{b_j}| \leqslant 2(z_j-1) + 2(p-z_j-1) + p - 4 + 2\lceil p/2 \rceil$. Puisque $2\lceil p/2 \rceil \leqslant p+1$, il suit que $|T_{a_j}| + |T_{b_j}| \leqslant 4p-7$. Si $z_j \leqslant 1$ ou si $z_j = p$, alors $2^{h_{a_j}} + 2^{h_{b_j}} \leqslant 1 + 2(p-1)$ et donc $|T_{a_j}| + |T_{b_j}| \leqslant 4p-4$.

Dans tous les cas, $|T_{a_j}| + |T_{b_j}| \leqslant 4p-4$ et $|T_{v_i}| \leqslant 3q-4$. Par conséquent l'ordre de G est $|G| \leqslant \sum_{j=1}^{q}(|T_{a_j}| + |T_{b_j}|) + p|T_{v_i}| - pq \leqslant q(4p-4) + p(3q-4) - pq = 6pq - 4p - 4q$. □

4.4.2 Borne inférieure de $\mathrm{IRS}(n, d)$

La théorème suivant établit une borne inférieure de $\Omega(n/\log^2 n)$ sur la compacité de tout graphe de degré borné d'ordre n.

Théorème 4.2 *Pour tout entier suffisamment grand n et pour toute constante $d \geqslant 3$,*

$$\mathrm{IRS}(n,d) \geqslant \frac{n + 4\log_2 n + 5}{4\log_2 n(3\log_2 n + 1)} = \Omega\left(\frac{n}{\log^2 n}\right)$$

Preuve. Soit n un entier suffisamment grand, soit $p = 2^{q-1}$ et soit q tel que $6pq - 4(p+q) = 3q2^q - 2^{q+1} - 4q \leqslant n < 3(q+1)2^{q+1} - 2^{q+2} - 4(q+1)$. En appliquant le lemme 4.6, pour toute matrice $p{\times}q$ booléenne M', il existe un graphe de contraintes G' de M' de degré maximum d et d'ordre n', $n' \leqslant n$. Nous construisons un graphe G avec un degré maximum d et exactement n sommets à partir de G' en ajoutant un chemin de $n - n'$ sommets connecté à l'un des sommets $w_{i,j}$ de G' — les sommets $w_{i,j}$ sont tous de degrés 2. Clairement $\mathrm{IRS}(G) \geqslant \mathrm{IRS}(G')$ puisque M', matrice de contraintes de G', est aussi une matrice de contraintes pour G. Donc, pour tout $d \geqslant 3$, $\mathrm{IRS}(n,d) \geqslant \mathrm{IRS}(G)$. En appliquant le lemme 4.1, pour toute matrice $2^{q-1} \times q$ booléenne M', $\mathrm{IRS}(n,d) \geqslant \mathrm{I}(M')$. Soit $P = (v_1, \ldots, v_{2^q})$ un chemin hamiltonien de l'hypercube de dimension q. Soit M la matrice booléenne $2^{q-1} \times q$ telle que la ligne r_i de M, $i \in \{1, \dots, 2^{q-1}\}$, soit la représentation binaire standard du sommet v_{2i-1} de P. Par construction, deux lignes r_i et r_j de M, $i \neq j$, diffèrent sur au moins deux bits, puisque deux sommets adjacents dans l'hypercube sont à une distance impaire dans P (ce qui est équivalent à dire que deux sommets à distance pair dans P ne sont pas adjacent, donc à distance au moins 2 dans l'hypercube). Par conséquent le rayon de M est 2 et, d'après le lemme 4.2, nous avons $\mathrm{I}(M) \geqslant 2^q/(2q)$. Ainsi

$$\begin{aligned} n < 3(q+1)2^{q+1} - 2^{q+2} - 4(q+1) \quad &\Rightarrow \quad n + 4q + 5 \leqslant \frac{2^q}{2q} 4q(3q+1) \\ &\Rightarrow \quad \frac{n+5}{4q(3q+1)} + \frac{1}{3q+1} \leqslant \mathrm{I}(M) \end{aligned}$$

Or, $n \geqslant 3q2^q - 2^{q+1} - 4q \geqslant 2^q$, pour tout $q \geqslant 3$ et tout $n \geqslant 44$. Par conséquent, $\log_2 n \geqslant q$ et finalement,

$$\mathrm{IRS}(n,d) \geqslant \frac{n+5}{4\log_2 n(3\log_2 n + 1)} + \frac{1}{3\log_2 n + 1}$$

□

Remarque. La borne inférieure établie par le théorème 4.2 peut être améliorée puisque la construction résulte du lemme 4.6 qui ne fournit qu'une construction générale pour toute matrice booléenne $p \times q$, alors que la preuve ci-dessus utilise une matrice très particulière. Cependant, l'ordre de grandeur de la borne inférieure de la compacité maximale, utilisant un graphe de contraintes d'ordre $O(pq)$ et un degré maximum 3, reste en $\Omega(n/\log^2 n)$.

Nous allons voir au chapitre 6, que cette borne inférieure pour les réseaux de degré borné n'est pas optimale, en utilisant une généralisation des outils présentés dans ce chapitre.

4.5 Cas des graphes planaires

Après avoir considéré la classe des réseaux de degré borné, la recherche de grandes classes de réseaux utilisables en pratique mène naturellement à la classe des réseaux planaires. L'étude du routage compact dans les réseaux planaires a été menée principalement par Frederickson et Janardan dans plusieurs articles [30, 31, 32, 50].

En relation directe avec le routage par intervalles, le résultat suivant est dû à Janardan et Siu Cheng [49, lemme 1].

Théorème 4.3 [49, lemme 1] *Tout graphe planaire G à p plans vérifie* $\mathrm{IRS}(G) \leqslant \dfrac{3p}{2}$.

Le nombre de « plans » d'un graphe planaire est défini comme le plus petit nombre de faces nécessaires pour couvrir tous les sommets du graphe, pris sur toutes les représentations possibles de G dans un plan. On obtient comme corollaire du théorème 4.3 le résultat déjà énoncé dans [30] :

Corollaire 4.2 *[30] Tout graphe planaire-externe est de compacité* 1.

En effet les graphes planaires-externes sont les graphes planaires à un plan. Dans [30] la réciproque de ce corollaire est établie dans le modèle du routage par intervalles stricts avec des coûts non uniformes sur les arêtes.

Le problème du calcul de la compacité d'un graphe planaire est ouvert, tout comme celui de la caractérisation des graphes planaires de compacité 1, ou même de compacité linéaire 1. Nous avons cependant:

Théorème 4.4 *Les graphes planaires sans isthme d-réguliers de compacité linéaire* 1 *sont les* 7 *graphes suivants :*
- K_2, K_3, K_4
- $K_2 \times K_2$, $K_2 \times K_3$, $K_2 \times K_4$
- $K_2 \times K_2 \times K_2$

L'établissement de ce résultat étant technique nous ne présentons ici que l'idée de la preuve.

Idée de la preuve. Les 7 graphes précédents sont évidemment de compacité linéaire 1 d'après le théorème 3.7. De plus, ils sont réguliers et planaires. Un graphe d-réguliers ne peut être planaire que pour $d \leqslant 4$. L'étude des graphes 1- et 2-réguliers est triviale puisqu'il s'agit du graphe K_2 et des cycles C_n. Or, d'après le théorème 3.5, seuls les cycles C_n avec $n \leqslant 4$ sont tels que $\mathrm{LIRS}(C_n) = 1$, ce qui est le cas de K_3 et de $K_2 \times K_2$. L'étude se ramène donc aux cas $d = 3$ ou $d = 4$ (notez qu'il existe une infinité de graphes planaires sans isthme 3- ou 4-réguliers). On démontre alors le lemme suivant : si $\mathrm{LIRS}(G) = 1$ alors toute arête de G est soit un isthme, soit appartient à un cycle de longueur 3 ou 4.

Par une étude de cas, il suffit donc de démontrer que tout graphe planaire sans isthme 3- ou 4-régulier qui n'est pas un produit de graphes complets possède une arête où tout cycle la traversant est de longueur au moins 5. □

Le nombre de plans d'un graphe planaire d'ordre n pouvant atteindre $\Theta(n)$, on peut conjecturer que la borne du théorème 4.3 est exacte en ordre de grandeur.

Conjecture 4.1 *Il existe un graphe planaire d'ordre n tel que* $\mathrm{IRS}(G) = \Omega(n)$.

Nous rappelons que la meilleure borne inférieure de la compacité d'un graphe planaire d'ordre n est en $\Omega(\sqrt[3]{n})$ [58].

4.A Annexe : compacité des graphes et matrices de contraintes

Proposition 4.1 *Il existe un graphe G tel que* $\mathrm{IRS}(G) > \mathrm{I}(M)$ *pour toute matrice de contraintes M de G.*

Preuve. Considérons le graphe G d'ordre 7 dessiné sur la figure 3.1 page 15. Nous avons à construire toutes les matrices de contraintes possibles de G et vérifier que chacune d'elles a une compacité au plus 1. En fait, il est suffisant de le vérifier pour les matrices $p \times q$ avec $p \geqslant 4$ et $q \geqslant 3$ car il est facile de voir que toute matrice, avec p ou q plus petit, a une compacité au plus 1. Notez que la compacité d'une matrice peut être nulle d'après la définition 4.1. De plus, il suffit seulement de vérifier pour $p \leqslant 6$ parce que nous avons supposé que tout arc (u, v) n'est pas une contrainte pour le sommet u (le sommet u peut éventuellement appartenir à un intervalles associé à l'arc (u, v)). Par conséquent, le nombre p de lignes de toute matrice de contraintes de G est au plus 6. Comme les matrices de contraintes ont la même compacité à permutation des lignes et des colonnes près, le lecteur pourra facilement vérifier que toute les matrices de contraintes de G avec $4 \leqslant p \leqslant 6$ et $3 \leqslant q$ ont une compacité au plus 1.

Nous allons prouver maintenant que $\mathrm{IRS}(G) = 2$. Supposons qu'il existe une fonction de routage par intervalles de plus courts chemins $R = (\mathcal{L}, \mathcal{I})$ sur G qui soit de compacité 1. Pour simplifier la présentation de la preuve, nous posons $x = \mathcal{L}(x)$, pour tout $x \in \{a, b, c, d, e, f, g\}$. De même, si X et Y sont deux sous-ensembles de sommets de G, nous notons $X < Y$ si pour tout $(x, y) \in X \times Y$, $x < y$. Puisque les sommets b et c sont isomorphes, on supposera sans perte de généralité que $b < c$. L'ordre $d < b < a < c$ (ordre circulaire modulo 7) est impossible car l'intervalle $I_{(g,d)}$ assigné à l'arc (g, d), doit contenir d et a mais pas b ni c. Nous avons donc $\{a, d\} < b < c$. Les arcs (d, a) et (d, g) établissent la condition $\{e, f, g\} < \{a, b, c\}$. Par conséquent $\{a, e, f, g, d\} < b < c$. $I_{(g,f)}$ doit contenir f et c mais pas d ni b, donc $f < d < b < c$. De plus $I_{(g,e)}$ doit contenir e et b mais pas d ni c, donc $d < e < b < c$. Nous avons donc finalement $f < d < e < b < c$. Or, nous devons avoir $f < \{d, g\} < e < b < c$, car $I_{(a,d)}$ doit contenir d et g mais pas f, e, b et c. De la même manière nous devons avoir $f < \{a, d, g\} < e < b < c$, car $I_{(g,d)}$ doit contenir d et a mais pas f, e, b et c. Cette dernière condition est incompatible avec la condition $\{e, f, g\} < \{a, b, c\}$. Cette contradiction montre que R ne peut pas être de compacité 1. La figure 3.1 page 15 montre un example de fonction de compacité 2, donc $\mathrm{IRS}(G) = 2$.

□

4.B Annexe : le plus « petit » graphe de compacité 2

Dans cette annexe nous montrons que le plus « petit » graphe de compacité 2 possède 7 sommets, c'est-à-dire $N(2) = 7$ (selon la notation introduite au lemme 4.5). Ainsi, tout réseau de 6 sommets est de compacité 1. Pour montrer ce résultat, nous utilisons la liste de tous les graphes d'ordre plus petit que 7 qui peut être trouvée dans [44]. Le lemme suivant va permettre de réduire le nombre de cas à considérer.

Rappelons qu'un point d'articulation d'un graphe est un sommet dont la suppression déconnecte le graphe.

Lemme 4.7 *Soit G un graphe 1-sommet-connexe. La compacité de G est le maximum de la compacité de tous les sous-graphes de G composés d'une composante 2-sommet-connexe de G et de ses points d'articulations voisins dans G.*

Preuve. Soit $G = (V, E)$ un graphe 1-sommet-connexe. Un sous-graphe de G composé d'une composante 2-sommet-connexe et de ses points d'articulations dans G est appelée dans cette preuve une 2-*composante* de G. Soit A une 2-composante de G. Clairement A est un sous-graphe des plus courts chemins de G. Soit $k = \max_S \mathrm{IRS}(S)$ pour toute 2-composante S de G. Par application du théorème 3.6, nous avons que $\mathrm{IRS}(G) \geqslant k \geqslant \mathrm{IRS}(A)$.

Prouvons que $\mathrm{IRS}(G) \leqslant k$. La preuve est constructive : 1) décomposer G en ses 2-composantes, 2) procéder par étape en fusionnant successivement les 2-composantes et leur fonctions de routage par intervalles de plus courts chemins pour obtenir une nouvelle fonction de routage par intervalles de plus courts chemins sur G. À la première étape la phase 2) fusionne deux 2-composantes. Il en résulte un sous-graphe de G qui n'est plus une 2-composante de G. En fait, dans les fusions suivantes, nous fusionnons des sous-graphes qui ne sont certes pas des 2-composantes de G mais qui ont un point d'articulation en commun. Montrons comment faire la fusion en général.

Puisque G est un graphe 1-sommet-connexe, il existe un point d'articulation x de G et nous pouvons décomposer G en deux sous-graphes, $A = (V_A, E_A)$ et $B = (V_B, E_B)$, tels que $V_A \cup V_B = V$ et $V_A \cap V_B = \{x\}$. Nous supposons par induction que A et B sont de compacité au plus k, et nous allons prouver que $\mathrm{IRS}(G) \leqslant k$.

Soient $n_A = |V_A|$ et $n_B = |V_B|$. Soient $R_A = (\mathcal{I}_A, \mathcal{L}_A)$ et $R_B = (\mathcal{I}_B, \mathcal{L}_B)$ deux fonctions de routage par intervalles et de plus courts chemins sur les graphes A et B respectivement telles que $\mathcal{L}_A(x) = n_A$ et $\mathcal{L}_B(x) = 1$. Les conditions $\mathcal{L}_A(x) = n_A$ et $\mathcal{L}_B(x) = 1$ ne sont pas restrictives puisque toute permutation circulaire de la numérotation des sommets ne change ni les chemins de routage, ni la compacité.

Nous définissons une fonction de routage par intervalles $R = (\mathcal{L}, \mathcal{I})$ sur G comme suit : $\mathcal{L}(v) = \mathcal{L}_A(v)$ pour tout sommet v du graphe A, et $\mathcal{L}(w) = \mathcal{L}_B(w) + n_A - 1$, pour tout sommet $w \neq x$ du graphe B. Nous décalons aussi tout intervalle simple $[a, b] \in \mathcal{I}_B$ pour avoir un nouvel ensemble $\mathcal{I}'_B$ d'intervalles de la forme $[a + n_A - 1, b + n_A - 1]$. Nous étendons les ensembles $\mathcal{I}_A$ et $\mathcal{I}'_B$ pour obtenir un ensemble d'intervalles $\mathcal{I}$ pour G de la manière suivante : pour tout intervalle simple $I = [a, b]$ de $\mathcal{I}_A$ ou de $\mathcal{I}'_B$ contenant les entiers n_A, soit $I' = I \cup [n_A, n_A + n_B - 1]$. I' est composé de seulement un intervalle puisque tous ses éléments sont des entiers consécutifs. Nous posons finalement $\mathcal{I}$ comme l'union des intervalles étendus de $\mathcal{I}_A$ et $\mathcal{I}'_B$. Il est facile de voir que les plus courts chemins définit par R entre deux sommets d'un même sous-graphe A ou B sont les mêmes que ceux définis par R_A ou R_B. De plus tout plus courts chemins entre un sommet u de A et un sommet w de B, doit traverser le point d'articulation x, lequel appartient à l'ensemble des sommets de A et de B.

La fonction R contient donc au plus $\max\{\mathrm{IRS}(A), \mathrm{IRS}(B)\} \leqslant k$ intervalles par arcs. Par conséquent $\mathrm{IRS}(G) = k$. □

Le lemme suivant se révèle utile pour vérifier rapidement si un graphe ayant un grand nombre d'arêtes a une compacité 1. On appelle graphe complémentaire du graphe G, le graphe obtenu en complémentant la matrice d'adjacence de G.

Lemme 4.8 *Soit G un graphe d'ordre n ayant d sommets de degré $n-1$. Soit m le nombre de composantes arête-connexes ayant au moins deux sommets dans le graphe complémentaire de G. Si $d \geqslant (n-m)/2$ alors la compacité de G est 1. De plus $G \in$ 1-LIRSS.*

Preuve. Soit G un graphe d'ordre n. Supposons que G a d sommets de degré $n-1$. On note $\overline{G}$ le graphe complémentaire de G. $\overline{G}$ est composé de m composantes connexes $A_1, \ldots, A_m$ d'ordre au moins deux et de d sommets isolés. Supposons que $d \geqslant (n-m)/2$. Sans perte de généralité, supposons que $m \geqslant 1$, car sinon G est le graphe complet, de compacité 1. Pour chaque composante connexe A_i, $i \in \{1, \ldots, m\}$, nous enracinons un arbre couvrant T_i en un sommet quelconque de A_i. Soit n_i de nombre de sommets de A_i.

Nous allons maintenant construire une fonction de routage par intervalles de plus courts chemins $R = (\mathcal{L}, \mathcal{I})$ sur G. Pour tout $i \in \{1, \dots, m\}$ et pour tout sommet x_j de A_i, $j \in \{1, \dots, n_i\}$, nous fixons $\mathcal{L}(x_j) = 2j - 1 + \sum_{k=1}^{i-1} n_k$ dans un parcours en profondeur d'abord des sommet de A_i selon l'arbre T_i. Puisque $n - d \geqslant 2m$ (en effet chacune des m composantes a au moins 2 sommets et il y a d sommets isolés dans $\overline{G}$), $m \geqslant 1$ et $d \geqslant (n-m)/2$, on a $d \geqslant n - d - m \geqslant 1$. Nous pouvons donc numéroter $n - d - m$ sommets parmi les d sommets isolés y_j de $\overline{G}$ avec $\mathcal{L}(y_j) = 2j$, $j \in \{1, \dots, n-d-m\}$. Pour les autres sommets isolés y'_j de $\overline{G}$, s'ils existent, nous posons $\mathcal{L}(y'_j) = j$, $j \in \{2(n-d-m)+2, \dots, n\}$. L'ensemble $\mathcal{I}$ est défini comme suit :

i. Pour les d sommets x de G de degré $n-1$, nous affectons $I_{(x,y)} = [\mathcal{L}(y)]$, aux $n-1$ voisins y de x.
ii. Pour les sommets x de G de degré strictement plus petit que $n-1$, nous procédons de la manière suivante : pour tout arc (x,y) de G, on pose $I_{(x,y)} = [\mathcal{L}(y)-1, \mathcal{L}(y)]$ si x et y ne sont pas adjacent dans G, et $I_{(x,y)} = [\mathcal{L}(y)]$ sinon.

Donc, tous les sommets de G sont numérotés et R à au plus un intervalle strict linéaire par arc.

Pour prouver que R est connexe et définit une fonction de routage de plus courts chemins, considérons un couple de sommets (x,y) de G. Comme $n > d \geqslant 1$, car G a au moins un sommet de degré $n-1$, et comme $G \neq K_n$, G a donc un diamètre 2. Donc, ou bien x et y sont adjacents, ou il y a un troisième sommet z de G tel que z est connecté à la fois à x et à y. Si x et y sont adjacents, alors nous avons $\mathcal{L}(y) \in I_{(x,y)}$ dans les deux cas (i) and (ii), et, par symétrie, $\mathcal{L}(x) \in I_{(y,x)}$. Si x et y ne sont pas adjacents dans G, il existe $i \in \{1, \dots, m\}$ tel que x et y appartiennent tous deux à A_i. Par conséquent $\mathcal{L}(y) \in I_{(x,y)} = [\mathcal{L}(z) = \mathcal{L}(y) - 1, \mathcal{L}(y)]$, par (ii) car ni x ni y sont de degré $n-1$. Par symétrie, nous obtenons aussi $\mathcal{L}(x) \in I_{(y,x)}$. Nous avons donc prouvé que tous les chemins construits par R sont des plus courts chemins. □

Théorème 4.5 $N(2) = 7$.

Preuve. Nous avons déjà montré que $N(2) \leqslant 7$ (figure 3.1 page 15 et proposition 4.1). De plus $N(2) \geqslant 5$ puisque $N(k) \geqslant 2k+1$ pour tout $k \geqslant 2$. Vérifions que tout graphe d'ordre 5 ou 6 supporte une fonction de routage par intervalles de plus courts chemins avec un intervalle par arc. Nous n'avons pas besoin de vérifier les graphes planaires-externes puisqu'ils sont de compacité 1 [30]. Comme tout graphe d'ordre au moins 4 à une compacité 1 et grâce aux lemme 4.7 et 4.8, deux graphes seulement de 5 sommets doivent être vérifiés : ce sont $K_{2,3}$ et le graphe C_4 avec $K_{1,3}$ connecté par ses trois feuilles à trois sommets de C_4. Selon la représentation de ces graphes à la pages 217 de [44], une numérotation circulaire des sommets et une affectation triviale des intervalles donnent une fonction de routage par intervalles de plus courts chemins avec un intervalle par arc.

De même, puisque tout graphe d'ordre 5 a une compacité 1, alors, en appliquant le lemme 4.7 et le lemme 4.8, seulement 43 graphes d'ordre 6 (graphes planaires-externes exclus), représentés sur les pages 220–224 de [44], doivent être vérifiés. Une numérotation circulaire des sommets et une affectation triviale des intervalles donne une fonction de routage par intervalles de plus courts chemins avec un intervalle par arc, à l'exception d'un seul graphe : un C_5 dont deux sommets non adjacents sont reliés par un chemin de longueur 2. Pour ce graphe, nous pouvons numéroter ses sommets de manière circulaire par la suite $(1,2,6,5,4,3)$, toujours selon la représentation donnée page 220 [44] de ce graphe. Par conséquent $N(2) \geqslant 7$. $\square$

CHAPITRE 5

Méthodes de routage universel

5.1 Introduction

Nous avons vu que certains graphes d'ordre n construits à partir des matrices M_δ (définies par l'équation 4.1 page 43) ont une compacité d'au moins $n/12$ intervalles. Ceci implique de devoir stocker un grand nombre de bits sur certains routeurs (en supposant que chaque intervalles est stocké sur $O(\log n)$ bits). On peut se demander si une autre méthode de routage sur ce graphe aurait généré un routage plus compact.

Proposition 5.1 *Soit G le graphe d'ordre n construit dans la preuve du théorème 4.1 et ayant une compacité d'au moins $n/12$ intervalles. La mémoire locale de routage de G est au plus $2\log_2 n + O(1)$ bits.*

Preuve. La preuve du théorème 4.1 est constructive, le graphe G peut donc être construit entièrement à partir de la valeur de n. Autrement dit, il existe un algorithme $A(n)$ dépendant uniquement de n qui construit la matrice d'adjacence de G. La construction de G par $A(n)$ induit une numérotation canonique des sommets et des arcs de G (par exemple en numérotant les sommets et les arcs de manière croissante par ordre de création dans l'algorithme $A(n)$). Par conséquent, il suffit que chaque routeur de G possède un copie de $A(n)$ et le numéro de son propre sommet pour pouvoir router vers n'importe quelle destination. Il pourra par exemple utiliser un algorithme de calcul des plus courts chemins sur la représentation de G (voir Dijkstra et Floyd dans [14, 22]). Dans chaque routeur x, il suffit de stocker $\lceil \log_2 n \rceil$ bits pour décrire la valeur n, $\lceil \log_2 n \rceil$ autres bits pour décrire le numéro de x et un nombre constant de bits pour définir les différents algorithmes, ce qui fait au plus $2\lceil \log_2 n \rceil + O(1) = 2\log_2 n + O(1)$ bits de mémoire de routage. □

Cet exemple montre que l'information de routage peut être calculée plutôt que stockée dans une table ou codée par un ensemble d'intervalles. Avant de tenter d'imaginer une autre méthode de routage plus compacte que le routage par intervalles nous allons déterminer dans ce chapitre une limite inférieure de la compressibilité de l'information de routage.

Plusieurs méthodes de routage et stratégies de codage des fonctions de routage on été proposées pour réduire l'espace mémoire de chacun des routeurs. Généralement, l'efficacité des fonctions de routage est mesurée par le « facteur d'élongation » qui est le rapport maximum entre la longueur des chemins construits par la fonction de routage et de la distance entre les sommets considérés. Plusieurs méthodes de routage ont été proposées dans [31] pour les graphes planaires, chacune nécessitant sur tous les routeurs un total de $O(n^{4/3}\log n)$ bits pour un facteur d'élongation de 3. Les mêmes auteurs ont proposé dans [32] des méthodes de routage pour des graphes c-décomposables, c'est-à-dire qui peuvent être décomposés récursivement par un *séparateur* de taille au plus c, où $c \geqslant 2$. Leurs méthodes utilisent un total de $O(c^2 \log c \cdot n \log^2 n)$ bits pour router avec un facteur d'élongation au plus $1 + 2/\alpha$ où $1 < \alpha \leqslant 2$ est la racine positive de l'équation $\alpha^{\lceil (c+1)/2 \rceil} - \alpha - 2 = 0$. Plus généralement Peleg et Upfal [68] ont établi des méthodes de routage valables pour tous les réseaux et pour tout facteur d'élongation fixé. Ces méthodes génèrent des fonctions de routage de facteur d'élongation au plus $12k + 3$, pour tout entier $k \geqslant 1$ fixé, et utilisent un total de $O(k^3 n^{1+1/k} \log n)$ bits mémoire, améliorant ainsi de nombreux travaux antérieurs [6, 53, 54, 69, 77]. Ils ont également donné dans [67] une borne inférieure de la mémoire globale de routage de tout graphe d'ordre n en $\Omega(n^{1+1/(2s+4)})$ bits pour tout facteur d'élongation s. Concernant le routage de plus courts chemins, nous déduisons de cette borne inférieure qu'il existe des graphes d'ordre n possédant une mémoire globale de routage d'au moins $\Omega(n^{7/6})$ bits et une mémoire locale de routage d'au moins $\Omega(\sqrt[6]{n})$ bits. Nous allons montrer dans ce chapitre que certains graphes possèdent en fait une mémoire locale de routage d'au moins $\Omega(n^2)$ bits pour tout facteur d'élongation au plus 2.

La section suivante donne les définitions de plusieurs notions nécessaires à l'établissement des résultats de ce chapitre. La section 5.3 généralise les notions de matrice et de graphe de contraintes introduites au chapitre 4. Enfin, la section 5.4 établit une borne inférieure de la mémoire globale de routage des réseaux de degré maximum non borné et borné.

Les principaux résultats présentés dans ce chapitre ont été établis en collaboration avec Pierre Fraigniaud et ont donné lieu à une publication internationale et une soumission à une revue internationale (voir [27, 28, 29]).

5.2 Méthode de routage universel

5.2.1 Facteur d'élongation

Le « facteur d'élongation », introduit dans [67] permet de donner une mesure de l'efficacité d'une fonction de routage sur un graphe donné. Il est définit de la façon suivante.

Définition 5.1 (facteur d'élongation) *Soit R une fonction de routage sur un graphe G. Pour tout couple (x,y) de sommets de G, on note $d_G(x,y)$ la distance de x à y dans G, et $d_R(x,y)$ la longueur du chemin construit par R de x à y. Le* facteur d'élongation *de R, noté $s(R,G)$, est défini par :*

$$s(R,G) \ = \ \max_{x \neq y} \frac{d_R(x,y)}{d_G(x,y)}$$

Le facteur d'élongation a au moins deux intérêts. Premièrement, il mesure l'efficacité du routage sans uniquement prendre en considération les cas extrêmes : plus courts chemins ou diamètre. Dans le cas des réseaux de diamètre non borné par exemple, la longueur des chemins de routage d'une fonction de routage non de plus courts chemins entre sommets voisins sera bornée par le facteur d'élongation et non pas par une expression dépendante du diamètre. Deuxièmement, le facteur d'élongation permet d'établir un compromis entre l'espace mémoire (information collectée par voisinage) et le temps (la longueur de chemins).

5.2.2 Borne supérieure de la mémoire de routage

Nous allons généraliser les définitions introduites au chapitre 2. La mémoire de routage est liée à deux éléments : la stratégie de codage des fonctions de routage et la stratégie de routage sur les graphes, c'est-à-dire le choix des chemins de routage. Une méthode de routage est définie par une stratégie de codage et de routage.

Dans la suite nous parlerons de *classe de fonctions de routage* pour décrire, par exemple, l'ensemble des fonctions de routage de plus courts chemins ou bien l'ensemble des fonctions de routage de facteur d'élongation $< s$. Pour toute classe de graphes $\mathcal{G}$, on note $|\mathcal{G}|$ le nombre de graphes non isomorphes de $\mathcal{G}$ (deux graphes sont isomorphes si et seulement si ils sont identiques à une numérotation près de leurs sommets). Rappelons que $\mathrm{MEM}(G,R,x)$, représente le nombre minimum de bits pour coder la fonction de routage locale R_x sur G (voir définition 2.3 du chapitre 2).

Définition 5.2 *Soient $\mathcal{G}$ une classe de graphes et $\mathcal{R}$ une classe de fonctions de routage sur $\mathcal{G}$. On définit :*

$$\mathrm{MEM}_{globale}(\mathcal{G},\mathcal{R}) = \max_{G\in\mathcal{G}} \min_{R\in\mathcal{R}} \sum_{x\in V} \mathrm{MEM}(G,R,x)$$

$$\mathrm{MEM}_{locale}(\mathcal{G},\mathcal{R}) = \max_{G\in\mathcal{G}} \min_{R\in\mathcal{R}} \max_{x\in V} \mathrm{MEM}(G,R,x)$$

où la minimisation sur $R \in \mathcal{R}$ est faite pour toute fonction de routage de $\mathcal{R}$ sur G, d'ensemble de sommets V.

Notations. Pour tout entier n et tout réel $s \geqslant 1$, on note $\mathrm{MEM}_{globale}(n,s) = \mathrm{MEM}_{globale}(\mathcal{G}_n,\mathcal{R}_s)$, où $\mathcal{G}_n$ est l'ensemble des graphes d'ordre n et $\mathcal{R}_s$ la classe des fonctions de routage de facteur d'élongation au plus s. De manière similaire on définit $\mathrm{MEM}_{locale}(n,s) = \mathrm{MEM}_{locale}(\mathcal{G}_n,\mathcal{R}_s)$.

Une borne inférieure de $\mathrm{MEM}_{locale}(\mathcal{G},\mathcal{R})$ est donc une borne inférieure de toute stratégie de codage des fonctions de routage de $\mathcal{R}$ pour la classe de graphes $\mathcal{G}$. Il est bien évident que nous sommes en fait intéressés par l'obtention de bornes supérieures pour une stratégie de codage fixée, indépendante de $\mathcal{G}$ et de $\mathcal{R}$. Nous avons par exemple :

Proposition 5.2 *Soit $\mathcal{P}_n$ la classe des graphes planaires d'ordre n. Alors, pour toute classe de fonctions de routage $\mathcal{R}$ sur $\mathcal{P}_n$, $\mathrm{MEM}_{locale}(\mathcal{P}_n,\mathcal{R}) \leqslant 12n + 2\log_2 n + O(1)$.*

Preuve. Il est bien connu que tout graphe planaires d'ordre n peut être décrit par au plus $12n$ bits, en utilisant le codage le Turán [79]. On applique donc la même idée que la proposition 5.1 en stockant localement une description du graphe, son ordre et l'adresse locale du sommet. □

Remarque. Nous avons vu que la valeur de la mémoire de routage est relative à un langage de programmation (voir la section 2.2 du chapitre 2). Par conséquent, pour toute classe de graphes $\mathcal{G}$ et pour toute classe de fonctions de routage $\mathcal{R}$ sur $\mathcal{G}$, il existe un langage ℓ pour lequel $\mathrm{MEM}_{locale}(\mathcal{G},\mathcal{R}) \leqslant \log_2|\mathcal{G}| + 2\log_2 n + O(1)$. En effet, puisque $|\mathcal{G}|$ est fini, ℓ peut très bien comporter chacun des graphes de $\mathcal{G}$ comme symboles du langage. Ainsi, il suffit de représenter un graphe $G_i \in \mathcal{G}$, $i \in \{1,\ldots,|\mathcal{G}|\}$, par l'indice i codé sur $\lceil\log_2|\mathcal{G}|\rceil$ bits ($2\lceil\log_2 n\rceil$ bits supplémentaires sont suffisant pour coder l'ordre du graphe et l'adresse locale).

La proposition 5.2 exprime un résultat plus fort : pour *tout* langage, $\mathrm{MEM}_{locale}(\mathcal{P}_n, \mathcal{R}) \leqslant 12n + 2\log_2 n + O(1)$.

Nous avons par conséquent pour la classe $\mathcal{A}_n$ des arbres d'ordre n et pour toute classe de fonctions de routage $\mathcal{R}$ sur $\mathcal{A}_n$, $\mathrm{MEM}_{locale}(\mathcal{A}_n, \mathcal{R}) = O(n)$ bits. Des techniques de représentation compacts des classes usuelles de graphes (forêts, fermeture transitive de forêts, graphes d'arboricité bornée, graphes d'intervalles, graphes arc-circulaires, graphes de cercles, graphes de permutations, graphes avec un nombre d'intervalles borné, graphes de chemins, graphes c-décomposable, *etc*) peuvent être trouvées dans [52] et [48].

En appliquant le même principe, nous avons pour la classe $\mathcal{G}_n$ des graphes d'ordre n et pour toute classe de fonctions de routage $\mathcal{R}$ sur $\mathcal{G}_n$, $\mathrm{MEM}_{locale}(\mathcal{G}_n, \mathcal{R}) = O(n^2)$. En effet, on peut toujours stocker la description complète d'un graphe donnée par sa matrice d'adjacence (éventuellement compressée) dans chaque routeur. Cependant, dans le cas des graphes d'ordre n, la matrice d'adjacence ne peut pas être compressée en général. En effet, d'après le théorème de Wright [82], le nombre de graphes non isomorphes d'ordre n est $|\mathcal{G}_n| \sim 2^{\binom{n}{2}}/n! = 2^{\Theta(n^2)}$. Cependant, le stockage de la description complète de G dans chaque routeur n'est pas la meilleure solution, puisque nous avons vu que les tables de routage utilisent seulement $O(n \log n)$ bits, donc $\mathrm{MEM}_{locale}(\mathcal{G}_n, \mathcal{R}) = O(n \log n)$.

Il peut paraître naturel de conjecturer que toute classe de graphes $\mathcal{G}$ nécessite au moins $|\mathcal{G}|$ fonctions de routage de plus courts chemins distinctes, une pour chaque graphe de $\mathcal{G}$. Par conséquent, quelque soit la stratégie de codage, certains graphes doivent avoir une mémoire globale de routage d'au moins $\log_2 |\mathcal{G}|$ bits. Nous allons cependant voir dans ce chapitre que cet argument est faux.

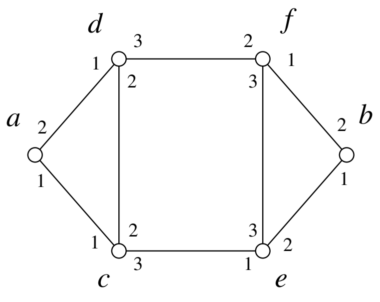
5.2.3 Isomorphisme de fonction de routage

Comme il a été remarqué dans [68], la même fonction de routage peut être utilisée sur plusieurs réseaux non isomorphes, car les voisins d'un sommet sont spécifiés par un numéro de port local indépendant de la numérotation globale des sommets.

L'exemple de la figure 5.1 présente deux graphes G et G' non isomorphes pour lesquels la même fonction de routage R peut être utilisée. Dans cet exemple, R est une fonction de routage représentée par une table. R est de facteur d'élongation $3/2$ sur les deux graphes. Sur G, $d_R(x,y)$ est minimum, excepté pour $d_R(c,b) = 3$ alors que $d_G(c,b) = 2$. Sur G', $d_R(x,y)$ est minimum, excepté pour $d_R(a,b) = d_R(e,b) = d_R(c,d) = 3$ alors que $d_{G'}(a,b) = d_{G'}(e,b) = d_{G'}(c,d) = 2$. Notez qu'une fonction de routage supportée par plusieurs graphes n'a pas nécessairement le même facteur d'élongation sur chacun des graphes. Peleg et Upfal ont montré dans [67] un exemple de graphes non isomorphes supportant une même fonction de routage de plus courts chemins.

Définissons plus formellement ce que signifie : « deux fonctions de routage sont isomorphes ».

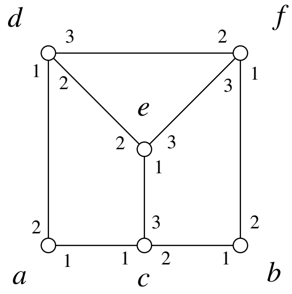
Définition 5.3 (isomorphisme de fonction de routage) *Soit $G = (V, E)$ un graphe et soit $R = \{(P_x, H_x) \mid x \in V\}$ une fonction de routage sur G. Soit $G' = (V', E')$ un graphe et soit $R' = \{(P'_x, H'_x) \mid x \in V'\}$ une fonction de routage sur G'. Soient $\mathcal{L}$ et $\mathcal{L}'$ les numérotations respectives des sommets de G et de G'. R est* isomorphe *à R' si et seulement si il existe une bijection $\varphi : V \to V'$, des bijections $\psi_x : \mathbb{N} \to \mathbb{N}$, $x \in V$, vérifiant chacune $\psi_x(0) = 0$, et une*



$s(R,G) = 3/2$

$\nearrow$	a	b	c	d	e	f
a	0	2	1	2	1	2
b	1	0	1	2	1	2
c	1	2	0	2	3	3
d	1	3	2	0	2	3
e	1	2	1	2	0	3
f	2	1	3	2	3	0

R



$s(R,G') = 3/2$

FIG. 5.1 – *Deux graphes G et G' non isomorphes mais supportant la même fonction de routage R avec, pour les deux, un facteur d'élongation $3/2$.*

bijection $\gamma : \mathbb{N} \to \mathbb{N}$ vérifiant $\gamma \circ \mathcal{L}' \circ \varphi(x) = \mathcal{L}(x)$ pour tout $x \in V$, telles que pour tout $x \in V$ et tout $(p,h) \in \mathbb{N}^2$:

i. $P'_{\varphi(x)}(\psi_x(p), \gamma(h)) = \psi_x \circ P_x(p,h)$

ii. $H'_{\varphi(x)}(\psi_x(p), \gamma(h)) = \gamma \circ H_x(p,h)$

Cette définition peut être exprimée sous une forme implicite : R est isomorphe à R' si et seulement si, à une numérotation près des sommets, des ports et des entêtes, R est une fonction de routage sur G' et R' est une fonction de routage sur G.

Intuitivement, le nombre de graphes supportant une même fonction de routage de facteur d'élongation au plus s croît avec s. Plus précisément, Peleg et Upfal on montré dans [67] que le nombre de fonctions de routage non isomorphes de facteur d'élongation au plus s est au moins $2^{\Omega(n^{1+1/(2s+4)})}$. Donc, quelque soit la stratégie de codage des fonctions de routage, $\mathrm{MEM}_{globale}(n,s) = \Omega(n^{1+1/(2s+4)})$.

Le résultat principal de ce chapitre est qu'il existe $2^{\Theta(n^2)}$ fonctions de routage de facteur d'élongation au plus 2. D'où le résultat :

Théorème 5.1 *Pour tout $s < 2$, $\mathrm{MEM}_{globale}(n,s) = \Omega(n^2)$.*

Notons que le théorème 5.1 implique que pour tout $s < 2$, $\mathrm{MEM}_{locale}(n,s) = \Omega(n)$. Nous allons cependant voir au chapitre suivant un meilleur résultat que celui-ci pour la mémoire locale de routage.

5.2.4 Stratégie de routage

Le théorème 5.1 est basé sur le nombre de fonctions de routage non isomorphes. La formalisation du comptage des fonctions de routage non isomorphes est elle-même basée sur la notion de *stratégie de routage* :

Définition 5.4 (stratégie de routage) *Soit $\mathcal{G}$ un ensemble de graphes et $\mathcal{R}$ une classe de fonction de routage sur $\mathcal{G}$. Une* stratégie de routage *est une fonction $RS : \mathcal{G} \to \mathcal{R}$ telle que,* *pour tout graphe $G \in \mathcal{G}$, $RS(G)$ est une fonction de routage de $\mathcal{R}$ sur G. De plus RS est une* stratégie de routage universel *pour $\mathcal{R}$ si $\mathcal{G}$ est l'ensemble de tous les graphes d'ordre n.*

Une méthode de routage universel est donc définie par une stratégie de routage universel et d'une stratégie de codage des fonctions de routage. Les tables de routages sont des méthodes de routage universel puisque tout graphe supporte au moins une fonction de routage codée par une table de routage. À l'inverse, pour le routage de plus courts chemins, le routage par intervalles avec un intervalle au plus par arc n'est pas une méthode de routage universel, car certains graphes ne supportent pas de fonction de routage qui puisse être codée avec au plus un intervalle par arc. La définition d'une stratégie de routage nous permet de donner une borne inférieure de le nombre de fonctions de routage. En effet la cardinalité minimum de l'ensemble image de RS, minimisée sur toutes les stratégies de routage RS de $\mathcal{G}$ sur $\mathcal{R}$, est précisément le nombre de fonctions de routage non isomorphes de la classe $\mathcal{R}$.

Nous allons construire une classe particulière de graphes telle que deux graphes différents de cette classe ne supportent pas deux fonctions de routage non isomorphes de facteur d'élongation au plus 2. Cette classe $\mathcal{G}$ est similaire à la classe des graphes de contraintes définies au chapitre 4 et dont la construction est basée sur une matrice booléenne. Nous allons étendre la définition des matrices de contraintes introduite à la section 4.2 pour tenir compte du facteur d'élongation.

5.3 Matrices et graphes de contraintes

5.3.1 Matrices de contraintes et facteur d'élongation

Définition 5.5 (matrice de contraintes de facteur d'élongation s) *Soit s un réel, $s \geqslant 1$. Soit $G = (V, E)$ un graphe quelconque. Soient $\{v_i \mid 1 \leqslant i \leqslant p\} \subset V$ et $\{e_j = (x_j, y_j) \mid 1 \leqslant j \leqslant q\} \subset E$. Une* matrice de contraintes *de G et* de facteur d'élongation *s est une matrice $p \times q$ booléenne $M = (m_{i,j})$ telle que, pour tout $i \in \{1, \dots, p\}$, et pour tout $j \in \{1, \dots, q\}$:*

$$\begin{cases} m_{i,j} = 1 \Leftrightarrow \textit{tous les chemins de longueur au plus } sd_G(x_j, v_i) \textit{ de } x_j \textit{ à } v_i \textit{ dans } G \textit{ utilisent l'arc } e_j. \\ m_{i,j} = 0 \Leftrightarrow \textit{aucun chemin de longueur au plus } sd_G(x_j, v_i) \textit{ de } x_j \textit{ à } v_i \textit{ dans } G \textit{ utilise l'arc } e_j. \end{cases}$$

La définition d'une matrice de contraintes de facteur d'élongation 1 coïncide avec la définition 4.1 des matrices de contraintes utilisée au chapitre précédent. Comme nous l'avons déjà vu, toute matrice de contraintes d'un graphe est définie à une permutation près des lignes et des colonnes.

Définition 5.6 (indice et représentant canonique) *soit $M = (m_{i,j})$ une matrice booléenne $p \times q$. L'*indice *de M est l'entier positif défini par :*

$$\mu(M) = \sum_{1 \leqslant i \leqslant p, 1 \leqslant j \leqslant q} m_{i,j} 2^{(p-i)+p(q-j)}$$

Pour toute classe C de matrices $p \times q$, le représentant canonique *de C est l'unique matrice $M_0 \in C$ telle que*

$$\mu(M_0) = \max_{M \in C} \mu(M)$$

L'indice $\mu(M)$ d'une matrice $p{\times}q$ peut être vu comme la valeur de l'expression binaire de M obtenue en concaténant les q mots de p bits formés par les colonnes de M.

Notation. Pour tout couple d'entier (p,q), considérons l'ensemble des matrices $p{\times}q$ booléennes M telle que M est le représentant canonique de sa classe d'équivalence de matrices définie à permutation des lignes et des colonnes près, et à une complémentation des bits des colonnes près. On note $\mathcal{M}_{p,q}$ cet ensemble privée des matrices ayant une colonne contenant exactement un 1 ou un 0. Par exemple :

$$\mathcal{M}_{4,2} = \left\{ \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \right\} \tag{5.1}$$

Lemme 5.1 *Pour tout couple d'entiers (p,q),*

$$\frac{(2^p-p)^q}{p!\,q!\,2^q} \leqslant |\mathcal{M}_{p,q}| \;=\; 2^{\Omega(pq-p\log p-q\log q)}$$

Preuve. Il y a $(2^p-p)^q$ matrices sans colonne ayant un seul 1 ou un seul 0. Par conséquent il y a au moins $(2^p-p)^q/(p!q!)$ classes différentes de matrices sans un unique 1 ou 0 dans une colonne et à une permutation des lignes et des colonnes près. D'autre part, il y a au plus 2^q manières distinctes de complémenter tous les bits des q colonnes d'une matrice booléenne $p{\times}q$. $\square$

Pour plus de détails concernant le nombre de classes des matrices booléennes définies par une permutation des lignes et des colonnes, avec ou sans complémentations des colonnes, se reporter à [45].

Soit $\sim$ la relation d'équivalence définie dans l'ensemble des matrices booléennes telle que $M \sim M'$ si et seulement si $M = M'$ à une permutations des lignes et des colonnes près. Le lemme suivant va être utile pour le dénombrement des fonctions de routage non isomorphes.

Nous employons les notations matricielles déjà utilisées dans la preuve du lemme 4.3.

Lemme 5.2 *Pour toutes matrices $M, M' \in \mathcal{M}_{p,q}$, si $M \neq M'$ alors $[\,M\,\overline{M}\,] \not\sim [\,M'\,\overline{M'}\,]$.*

Preuve. Soient $M, M' \in \mathcal{M}_{p,q}$ telles que $M \neq M'$. Par définition de l'ensemble $\mathcal{M}_{p,q}$, $M \neq M' \Rightarrow M \not\sim M'$. Montrons que :

$$\max_{X\sim[M\overline{M}]} \mu(X) = \mu([MY]) \tag{5.2}$$

pour une matrice Y particulière. En effet, supposons par contradiction que $\max_{X\sim[M\overline{M}]}\mu(X) = \mu([NZ])$ avec $N \neq M$. Il existe alors une permutation r des lignes de $[M\overline{M}]$ et une permutation c des colonnes de $[M\overline{M}]$, telles que $NZ = r(c([M\overline{M}]))$ et $\mu([NZ]) > \mu([MY])$ pour toute matrice Y obtenue d'après $\overline{M}$ par permutations des colonnes. Notons m_i les colonnes de M, $i \in \{1,\dots,q\}$, et $\overline{m_i}$ les colonnes correspondantes de la matrice $\overline{M}$. Puisque $N \neq M$, N contient une colonne $\overline{m_j}$ à une permutation des lignes près. De plus la colonne m_j ne peut pas appartenir à la matrice Z parce que $M \in \mathcal{M}_{p,q}$ et la matrice obtenue d'après M

par complémentation de la colonne j et en appliquant la permutation r des lignes aurait un indice plus grand que $\mu(M)$. Donc N contient les deux colonnes m_j et $\overline{m_j}$. Par conséquent, la matrice Z contient une colonne m_k de M. En considérant chaque colonne comme une matrice $p\times 1$, nous avons $\mu(m_j) > \mu(m_k)$, car sinon la matrice obtenue à partir de $[NZ]$ en échangeant les colonnes m_j et m_k aurait un indice plus grand que $\mu(Z)$. De manière similaire $\mu(\overline{m_j}) > \mu(m_k)$. Puisque $\mu(M)$ est l'indice maximum des matrices obtenues à partir de M par permutations de lignes et de colonnes, et par complémentations des colonnes, il suit que $\mu(m_k) > \mu(\overline{m_k})$. Donc $\mu(\overline{m_j}) > \mu(m_k) > \mu(\overline{m_k})$, *i.e.* $\mu(\overline{m_j}) > \mu(\overline{m_k})$. Ce qui est une contradiction avec $\mu(m_j) > \mu(m_k)$. Nous concluons ainsi que toutes les colonnes de N sont les colonnes de M. Pour finir il est facile de constater que r est nécessairement la fonction identité car sinon $\mu(M)$ ne serait pas maximum. Donc $N = M$, ce qui prouve l'équation 5.2.

De manière similaire, $\max_{X\sim[M'\overline{M'}]}\mu(X) = \mu([M'Y'])$ pour une matrice Y' particulière. Puisque $\mu(M) \neq \mu(M')$ implique $\mu([MY]) \neq \mu([M'Y'])$, alors $\max_{X\sim[M'\overline{M'}]}\mu(X) \neq \max_{X\sim[M\overline{M}]}\mu(X)$, *i.e.* $[M\overline{M}] \not\sim [M'\overline{M'}]$. □

5.3.2 Graphes de contraintes pour un facteur d'élongation au plus 2

Nous allons établir une correspondance bijective entre l'ensemble des matrices de contraintes et un ensemble particulier de graphes. Nous rappelons que, pour tout couple de sommets (a,b), on note $\{a,b\}$ la paire d'arcs (a,b) et (b,a).

Définition 5.7 *Soit $M = (m_{i,j})$ une matrice booléenne $p\times q$, $i \in \{1,\dots,p\}$ et $j \in \{1,\dots,q\}$. Le* graphe de contraintes *de M est le graphe $G = (V,E)$ tel que :*
- $V = \{a_1,\dots,a_q\} \cup \{b_1,\dots,b_q\} \cup \{c_1,\dots,c_q\} \cup \{v_1,\dots,v_p\}$
- *pour tout $j \in \{1,\dots,q\}$, $\{c_j,a_j\}$ et $\{c_j,b_j\} \in E$*
- *pour tout $i \in \{1,\dots,p\}$ et $j \in \{1,\dots,q\}$, $\{a_j,v_i\} \in E$ si et seulement si $m_{i,j} = 0$*
- *pour tout $i \in \{1,\dots,p\}$ et $j \in \{1,\dots,q\}$, $\{b_j,v_i\} \in E$ si et seulement si $m_{i,j} = 1$*
- *aucun autre arc appartient à E*

L'équation 5.3 ci-dessous montre trois exemples de graphes de contraintes (voir l'ensemble noté $\mathcal{G}_{4,2}$), un pour chaque matrice de l'ensemble $\mathcal{M}_{4,2}$ de l'équation 5.1. On peut vérifier que ces graphes ne supportent pas de fonction de routage de facteur d'élongation < 2 isomorphes deux à deux. Nous allons voir que ce résultat est général pour tous les graphes obtenus à partir des matrices de $\mathcal{M}_{p,q}$. Le lemme suivant va également être utile pour la preuve du théorème 5.1.

Lemme 5.3 *Soient M une matrice booléenne $p\times q$ quelconque et $G = (V,E)$ son graphe de contraintes associé. Alors M est une matrice de contraintes de facteur d'élongation < 2 pour G. De plus :*
- $|V| = p + 3q$
- *pour tout $i \in \{1,\dots,p\}$, $\deg(v_i) = q$*
- *pour tout $j \in \{1,\dots,q\}$, $\deg(a_j) + \deg(b_j) = p + 2$ et $\deg(c_j) = 2$*

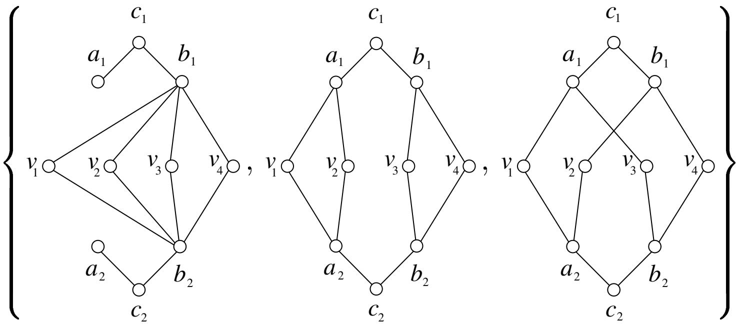
Preuve. Considérons le graphe de contraintes G d'une matrice $p\times q$ booléenne M. Soit A l'ensemble des arcs (c_j,b_j) pour $j \in \{1,\dots,q\}$, et soit $B = \{v_1,\dots,v_p\}$. Clairement, par définition de G, M est une matrice de contraintes de facteur d'élongation < 2 pour les arcs de

A et les sommets de B. En effet, tout arc $e \in A$ est soit un isthme, soit appartiennent à un cycle sans corde de longueur 6. De plus, il existe un unique arc $e \in A$ tel que $d_G(c_j, v_i) = 2$. Donc, si une fonction de routage R sur G utilise e au premier pas du routage alors qu'elle ne devrait pas, ou ne l'utilise pas quand elle le devrait, alors $d_R(c_j, v_i) \geqslant d_G(c_j, v_i) + 2$. R n'étant pas forcément simple, on ne peut pas garantir que $d_R(c_j, v_i) \geqslant d_G(c_j, v_i) + 6$ car un message émis de c_j sur le « mauvais » arc e peut très bien revenir en c_j après deux étapes pour être ensuite routé selon un plus court chemin. Par conséquent, le facteur d'élongation d'une telle fonction de routage est nécessairement au moins $(2+2)/2 \geqslant 2$. $\square$

Remarque. Ces graphes ont un diamètre borné par 6 et donc un degré maximum non borné.

Pour tout couple d'entier (p, q), nous notons $\mathcal{G}_{p,q}$ l'ensemble des graphes de contraintes de toutes les matrices de $\mathcal{M}_{p,q}$. L'équation 5.3 donne l'ensemble $\mathcal{G}_{4,2}$ dont les graphes sont représentés non orientés.

$\mathcal{G}_{4,2} =$



(5.3)

L'ensemble $\mathcal{G}_{p,q}$ contient des graphes qui sont deux à deux non isomorphes. Pour la nécessité de notre preuve, nous avons besoin d'une propriété plus forte qui est qu'ils ne supportent pas deux à deux la même fonction de routage de facteur d'élongation < 2. Nous démontrons cette propriété dans la section suivante.

5.4 Nombre de fonctions de routage

5.4.1 Énumération de l'ensemble image d'une stratégie de routage universel

Pour toute fonction $f : A \to B$, nous notons $f(A)$ l'ensemble image de f, c'est-à-dire $f(A) = \{y \in B \mid \exists x \in A, y = f(x)\}$. Le théorème 5.1 découle du résultat suivant.

Théorème 5.2 *Soit $\mathcal{R}$ un ensemble de fonctions de routage de facteur d'élongation < 2 sur la classe $\mathcal{G}_n$ des graphes d'ordre n. Pour toute stratégie de routage universel $RS : \mathcal{G}_n \to \mathcal{R}$,*

$$|RS(\mathcal{G}_n)| = 2^{\Theta(n^2)}$$

Preuve. Soit RS une stratégie de routage universel de $\mathcal{G}_n$ sur $\mathcal{R}$. $\mathcal{R}$ est composé de la classe des fonctions de routage de facteur d'élongation < 2. Considérons, pour tout entier $n \geqslant 20$, le sous-ensemble $\mathcal{G} = \mathcal{G}_{p,q}$ où $p = \lfloor n/4 \rfloor - 3 + ((n \bmod 4) \bmod 3)$ et $q = p + 4\lfloor (n+5)/4 \rfloor - n$. Comme l'ordre des graphes de $\mathcal{G}_{p,q}$ est $p + 3q$ (lemme 5.3), l'ordre des graphes de l'ensemble

$\mathcal{G}$ est exactement n, pour tout $n \geqslant 20$. Soient $M, M' \in \mathcal{M}_{p,q}$, $M \neq M'$. Soient $G = (V, E)$ et $G' = (V', E')$ les deux graphes de contraintes correspondants de $\mathcal{G}$. Pour toute fonction de routage $R = (P, H)$ sur G et pour toute fonction de routage $R' = (P', H')$ sur G', nous allons montrer que si $s(R, G) < 2$ et $s(R', G') < 2$, alors R n'est pas isomorphe à R'. Nous partionnons l'ensemble des sommets V de G en quatre sous-ensembles A, B, C et D, regroupant respectively les sommets a_j, b_j, c_j et v_i de V, pour $j \in \{1, \ldots, q\}$ et $i \in \{1, \ldots, p\}$ (voir définition 5.7). Nous partionnons V' de la même manière pour obtenir les ensembles de sommets A', B', C' et D'.

Supposons que R et R' soient isomorphes. Il existerait des bijections φ, ψ_x et γ telles que pour tout $x \in V$, pour tout $(p, h) \in \mathbb{N}^2$, les équations (i) et (ii) de la définition 5.3 soient satisfaites. On note $\mathcal{L}$ (respectively $\mathcal{L}'$) la numérotation des sommets de G (respectively G'). Soit $x \in V$, montrons que x et $\varphi(x)$ sont de même degré : supposons sans perte de généralité que $\deg(x) > \deg(\varphi(x))$. Puisque ψ_x est une bijection, il existe donc un arc (x, y) de E qui n'est pas utilisé par R sur G. Or les graphes de contraintes sont sans boucle ni multi-arcs, par conséquent $d_R(x, y) \geqslant 2$: une contradiction avec le fait que $s(R, G) < 2$. Donc, si R et R' sont isomorphes et de facteur d'élongation < 2, alors x et $\varphi(x)$ sont de même degré, pour tout $x \in V$.

D'après le lemme 5.3 et par définition des paramètres p et q comme fonctions de n, puisque $\lfloor n/4 \rfloor - 3 + ((n \bmod 4) \bmod 3) \geqslant 2$ et $4\lfloor (n+5)/4 \rfloor - n \geqslant 2$, pour tout $n \geqslant 20$, nous avons donc $\deg(c) < \deg(t) < \deg(v)$, pour tous sommets $c \in C \cup C'$, $v \in D \cup D'$ et $t \in A \cup B \cup A' \cup B'$. Par conséquent, $\varphi(C) = C'$, $\varphi(D) = D'$ and $\varphi(A \cup B) = A' \cup B'$.

Soit $N = [M\overline{M}]$ la matrice $p \times 2q$ composée de M et de sa matrice complément $\overline{M}$, et soit $N' = [M'\overline{M'}]$. Les matrices N et N' sont respectively matrice de contraintes de G (des arcs $(c, t) \in E$, pour $c \in C$ et $t \in A \cup B$, vers les sommets de D) et matrice de contraintes de G' (des arcs $(c', t') \in E'$, pour $c' \in C'$ et $t' \in A' \cup B'$, vers les sommets de D'). Il est facile de vérifier que N et N' sont équivalentes à permutation près des lignes et des colonnes, *i.e.* $N \sim N'$. Ces permutations dépendent des bijections φ, ψ_x et γ. D'autre part, nous savons que $M \neq M'$ (elles sont deux représentants canoniques différents de deux classes d'équivalences différentes de $\mathcal{M}_{p,q}$). Par conséquent, en appliquant le lemme 5.2, $[M\overline{M}] \not\sim [M'\overline{M'}]$. C'est une contradiction avec $N \sim N'$. Donc R et R' ne peuvent être isomorphes.

Par conséquent $|\mathcal{M}_{p,q}| = |\mathcal{G}|$. D'autre part, $|RS(\mathcal{G})| \leqslant |RS(\mathcal{G}_n)| \leqslant 2^{n^2}$. D'après le lemme 5.1, $|\mathcal{M}_{p,q}| = 2^{\Omega(pq - p\log p - q\log q)}$. Or $p = \Theta(n)$ et $q = \Theta(n)$, donc $|RS(\mathcal{G}_n)| = 2^{\Theta(n^2)}$. □

5.4.2 Preuve du théorème 5.1

Soient n un entier, $n \geqslant 20$, s un réel, $s < 2$, et RS^* la « meilleure » stratégie de routage universel pour les fonctions de routage de facteur d'élongation au plus s, c'est-à-dire une stratégie de routage qui, pour tout graphe $G = (V, E)$ d'ordre n, vérifie $\sum_{x \in v} \mathrm{MEM}(G, RS^*(G), x) = \min_{R \in \mathcal{R}_2} \sum_{x \in v} \mathrm{MEM}(G, R, x)$, où $\mathcal{R}_2$ est la classe des fonctions de routage de facteur d'élongation < 2. Par application du théorème 5.2, nous avons $|RS^*(\mathcal{G}_n)| = 2^{\Theta(n^2)}$, où $\mathcal{G}_n$ est l'ensemble de tous les graphes d'ordre n. Donc, toute stratégie de codage des fonctions de routage $RS^*(G)$ nécessite une mémoire globale de routage d'au moins $\log_2 |RS^*(\mathcal{G}_n)|$ bits pour au moins un graphe. Par conséquent, pour tout $n \geqslant 20$, il existe un graphe G d'ordre n tel que $\mathrm{MEM}_{globale}(G, s) = \Omega(n^2)$, car RS^* est la fonction de routage qui nécessite le moins de bits mémorise pour le graphe G.

Nous allons préciser maintenant le théorème 5.1 dans le cas des graphes de degré borné.

5.4.3 Nombre de fonctions de routage sur les graphes de degré borné

Dans ce paragraphe nous considérons la classe des graphes de degré maximum d. Nous allons calculer, comme précédemment, la cardinalité de l'ensemble image de toute stratégie de routage de l'ensemble des graphes de degré borné vers la classe des fonctions de routage de plus courts chemins.

Théorème 5.3 *Soit $\mathcal{G}_{n,d}$ l'ensemble des graphes d'ordre n et de degré maximum $d \geqslant 3$. Soit $\mathcal{R}$ la classe des fonctions de routage de plus courts chemins. Pour tout stratégie de routage $RS : \mathcal{G}_{n,d} \to \mathcal{R}$,*

$$|RS(\mathcal{G}_{n,d})| = 2^{\Omega(n)}$$

La preuve est basée sur la construction d'une classe de graphes de contraintes de degré au plus d. Nous allons montrer plus tard que deux graphes de cette classe ne peuvent supporter de fonctions de routage de plus courts chemins isomorphes. Dans la suite on appelle les graphes de cette classe *d-graphes de contraintes.*

Lemme 5.4 *Pour tout couple d'entiers (p,q), pour toute matrice $p \times q$ booléenne M et pour tout $d \geqslant 3$, il existe un graphe $G = (V,E)$ tel que :*

i. $|V| \leqslant 24pq + 8p + 4q$
ii. *G a un degré au plus d*
iii. *$[\,M\ \overline{M}\,]$ est une matrice de contraintes de G*
iv. *$V = B \cup H \cup W$, avec $|B| = p$ et $|H| = q$ et avec $\deg(u) = 1$, $\deg(v) = 2$ et $3 \leqslant \deg(w) \leqslant d$, pour tout $(u,v,w) \in B \times H \times W$.*

Preuve. Soient (p,q) un couple d'entiers quelconques, d un entier $\geqslant 3$, et M une matrice booléenne $p \times q$. Décrivons la construction d'un graphe $G = (V,E)$ satisfaisant les propriétés (i), (ii), (iii) et (iv) du lemme 5.4. Le graphe G est composé de trois niveaux de sommets appelés respectively Haut, Moyen et Bas. On note $H = \{c_1, \dots, c_q\}$ l'ensemble des sommets du niveau haut, $B = \{v_1, \dots, v_p\}$ l'ensemble des sommets Bas, et W l'ensemble des sommets du niveau Moyen. Le graphe G est composé de p graphes notés G_{v_i}, $i \in \{1, \dots, p\}$, et de q graphes notés G_{c_j}, $j \in \{1, \dots, q\}$. Ce graphe a une forme générale telle qu'indiquée sur la partie gauche de la figure 5.2. La construction des graphes G_{v_i} et G_{c_j} est basée sur un arbre que nous allons décrire.

Soit c_j un sommet de H, $j \in \{1, \dots, q\}$. Ce sommet est la racine d'un arbre $(d-1)$-aire, noté T_{c_j}. Les p feuilles de T_{c_j} sont des sommets de W, notés $d_{i,j}$, $i \in \{1, \dots, p\}$. Les deux fils de c_j sont les sommets de W notés a_j et b_j. Ces sommets sont les racines des sous-arbres de T_{c_j} notés T_{a_j} et T_{b_j} tels que $d_{i,j}$ est une feuille de T_{a_j} si $m_{i,j} = 0$ et $d_{i,j}$ est une feuille de T_{b_j} si $m_{i,j} = 1$, $i \in \{1, \dots, p\}$. La partie droite de la figure 5.2 montre un exemple d'arbre T_{c_j}. Les feuilles de T_{c_j} sont toutes à distance $\lceil \log_{d-1} p \rceil$ de la racine c_j. Le nombre de sommets d'un tel arbre est $|T_{c_j}| = |T_{a_j}| + |T_{b_j}| + 1$. Soit p_0 le nombre de zéros de la colonne j de M. Si $p_0 = 0$ alors $|T_{a_j}| = 1$, *i.e.* $\deg(a_j) = 1$. Sinon, $|T_{a_j}| \leqslant 2p_0 + \lceil \log_{d-1} p \rceil - \lceil \log_{d-1} p_0 \rceil$. Donc $|T_{a_j}| \leqslant 2p$ car T_{a_j} a p_0 feuilles, $1 \leqslant p_0 \leqslant p$, et une hauteur d'au moins $\lceil \log_{d-1} p_0 \rceil$. De manière similaire

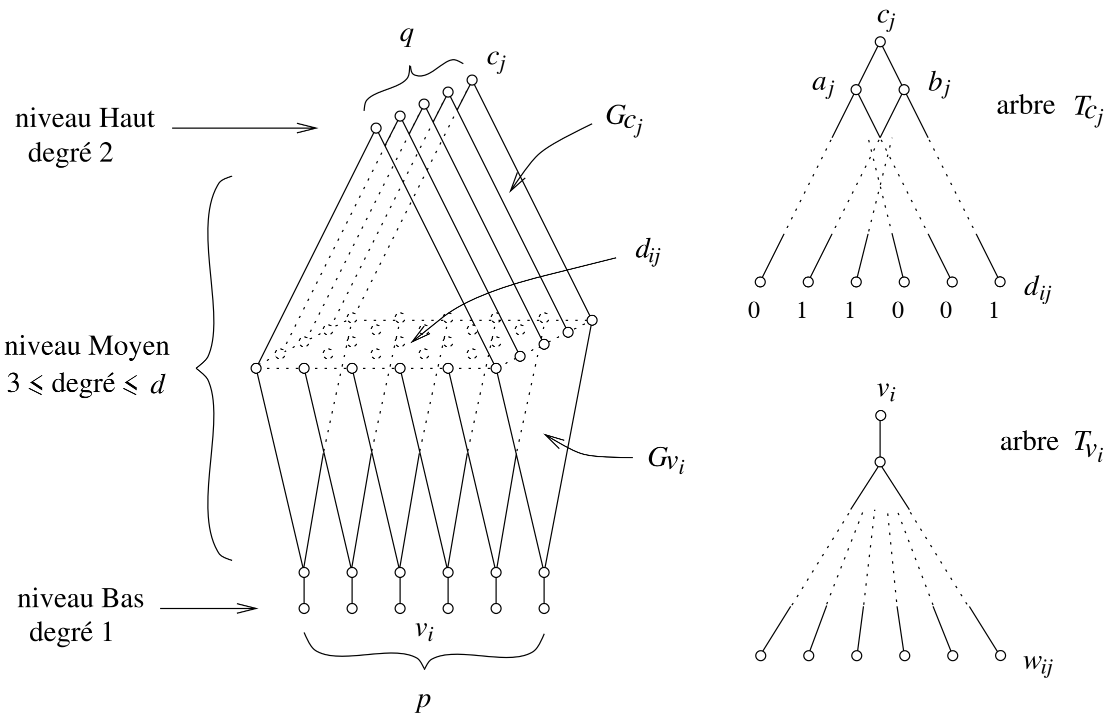


FIG. 5.2 – *Un d-graphe de contraintes*

$|T_{b_j}| \leqslant 2p$. Par conséquent $|T_{c_j}| \leqslant 4p+1$. Notons que certains sommets de T_{c_j}, en dehors de la racine c_j, sont de degré 1 ou 2. Le graphe G_{c_j} est obtenu par transformation de l'arbre T_{c_j} de sorte que tout sommet x de T_{c_j} qui ne soit ni la racine ni une feuille, soit tel que $3 \leqslant x \leqslant d$ et que toutes les feuilles de T_{c_j} soient toujours à égale distance de c_j. Soit x un tel sommet. La figure 5.3 montre comment, en échangeant l'arête connectée au père de x dans T_{c_j} on obtient G_{c_j}. Le nombre de sommets de G_{c_j} est $|G_{c_j}| \leqslant |T_{c_j}| + 5(|T_{c_j}| - p - 2) + 4 \leqslant 19p + 4$ car au plus $|T_{c_j}| - p - 2$ arêtes doivent être échangées par la structure d'au plus 5 sommets (voir figure 5.3). De plus, seuls a_j ou b_j peuvent être de degré 1, ce qui ajoute au plus 4 sommets.

Soit v_i un sommet du niveau Bas B, $i \in \{1, \ldots, p\}$. v_i est de degré 1 et est la racine d'un arbre $(d-1)$-aire, noté T_{v_i}, possédant q feuilles toutes situées à la même distance de la racine. Les feuilles sont les sommets notés $w_{i,j}$, $j \in \{1, \ldots, q\}$. La partie droite de la figure 5.2 montre un arbre T_{v_i}. Le nombre de sommets de T_{v_i} est $|T_{v_i}| \leqslant 2q + 1$. T_{v_i} peut contenir certains sommets de degré 2. Comme précédemment, nous transformons T_{v_i} en un graphe G_{v_i} en remplaçant tout sommet x de T_{v_i} de degré 2, en dehors de la racine, de manière à ce que $3 \leqslant \deg(x) \leqslant d$, et tel que toutes les feuilles soient encore à la même distance de v_i. On utilise pour cela la transformation donnée par la figure 5.3. Le nombre de sommet de G_{v_i} est $|G_{v_i}| \leqslant |T_{v_i}| + 5(\lceil \log_{d-1} q \rceil - 1) \leqslant 3q + 8$, car il y a au plus un sommet de degré 2 à chaque hauteur de l'arbre. De plus $\lceil \log_{d-1} q \rceil \leqslant (q + 12)/5$, pour tout $q \geqslant 1$ et pour tout $d \geqslant 3$.

Pour tout $i \in \{1, \ldots, p\}$ et $j \in \{1, \ldots, q\}$, le graphe G_{v_i} est connecté au graphe G_{c_j} pour former G. La connexion entre les sommets $d_{i,j}$ et $w_{i,j}$ est encore telle que tous les sommets sont de degré au moins 3 et au plus d (voir la partie droite de la figure 5.3). Par conséquent, tous les sommets de H sont de degré 2, tous les sommets de B de degré 1, et tous les sommets

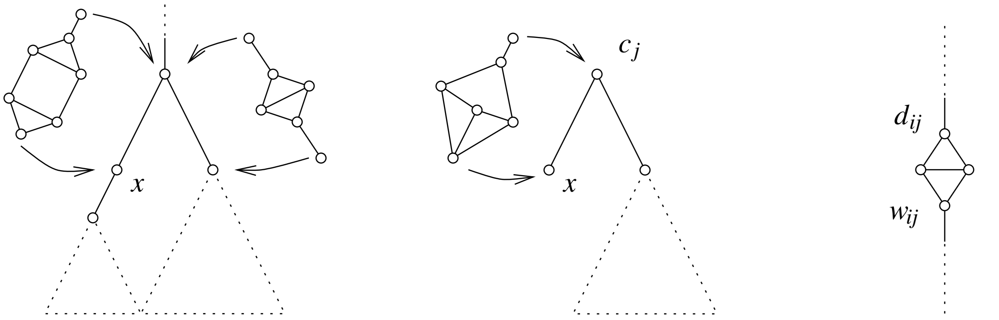


FIG. 5.3 – *Transformation de certaines arêtes pour la preuve du lemme 5.4.*

x de W sont tels que $3 \leqslant \deg(x) \leqslant d$, ce qui prouve (iv) et donc aussi (ii). Le nombre de sommets de G est $|V| \leqslant q(19p+4) + p(3q+8) + 2pq = 24pq + 8p + 4q$, ce qui prouve (i).

Pour tout $j \in \{1, \ldots, q\}$, soit $e_j = (c_j, a_j)$. Il est facile de vérifier que l'unique plus court chemin de c_j à un quelconque v_i, $i \in \{1, \ldots, p\}$ utilise l'arc e_j si et seulement $m_{i,j} = 1$. Les contraintes de v_i sur e_j, pour tout $i \in \{1, \ldots, p\}$, représente donc exactement la j-ième colonne de M. De manière similaire, l'arc (c_j, b_j) représente exactement le complément de la j-ième colonne de M pour le sommet v_i, pour tout $i \in \{1, \ldots, p\}$. Par conséquent $[M\overline{M}]$ est une matrice de contraintes de G, prouvant (iii). □

Preuve du théorème 5.3. Soit $d \geqslant 3$ et n un entier suffisamment grand. Soit RS une stratégie de routage de $\mathcal{G}_{n,d}$ sur l'ensemble $\mathcal{R}$ des fonctions de routage de plus courts chemins, où $\mathcal{G}_{n,d}$ est l'ensemble des graphes d'ordre n et de degré maximum d. Considérons le sous-ensemble $\mathcal{G} \subset \mathcal{G}_{n,d}$ des d-graphes de contraintes d'ordre n. Soient $p = q = \lfloor \sqrt{n}/6 \rfloor$. D'après le lemme 5.4, il existe un graphe G' pour toute matrice $M \in \mathcal{M}_{p,q}$ de degré maximum $\leqslant d$ et d'ordre $n' \leqslant 24pq + 8p + 4q = 24p^2 + 12p \leqslant (6p)^2 \leqslant n$. On peut transformer G' en un graphe G d'ordre exactement n en ajoutant une chemin A de $n - n'$ sommets au sommet v_1. On appelle v_1 le nouveau sommet de degré 1 dans ce chemin. Le graphe G satisfait les propriétés (i), (ii), (iii) et (iv) du lemme 5.4.

Soient $M, M' \in \mathcal{M}_{p,q}$, $M \neq M'$. Soit $G = (V, E)$ et $G' = (V', E')$ les deux d-graphes de contraintes correspondants de $\mathcal{G}$. Soient $R = (P, H)$ une fonction de routage de plus courts chemins sur G et $R' = (P', H')$ une fonction de routage de plus courts chemins sur G'. Nous allons montrer que R ne peut pas être isomorphe à R'. En accord avec les notations du lemme 5.4, on décompose $V = B \cup H \cup W \cup A$ et $V = B' \cup H' \cup W' \cup A'$ (A et A' est l'ensemble des sommets du chemins A, respectively A'). On pose $Z = \{(c_j, a_j), (c_j, b_j) \mid 1 \leqslant j \leqslant q\}$ et $Z' = \{(c'_j, a'_j), (c'_j, b'_j) \mid 1 \leqslant j \leqslant q\}$.

Supposons que R et R' sont isomorphes. Alors il existe des bijections φ, ψ_x, $x \in v$, et γ satisfaisant les équations (i), (ii) de la définition 5.3. Comme il est montré dans la preuve du théorème 5.2, pour tout $x \in V$, $\deg(x) = \deg(\varphi(x))$. Ce qui implique $\varphi(B) = B'$, $\varphi(W) = W'$ et $\varphi(H) \subset H' \cup A'$ (les sommets de A et de A' sont de degré 2). Prouvons que $\varphi(H) =$

H'. Supposons qu'il existe un sommet $x \in H$ tel que $\varphi(x) \in A'$. Considérons toutes les communications entre $\varphi(x)$ et les sommets de B utilisant l'arc $(\varphi(x), y')$, pour un voisin quelconque y' de $\varphi(x)$. Puisque $\varphi(B) = B'$, les contraintes de l'arc $(\varphi(x), y')$ sur B doivent être une colonne de M ou de $\overline{M}$. Or cette colonne doit contenir un seul 1 ou un seul 0 pour la communication du sommet $\varphi(x)$ à v_1. Ce qui est une contradiction avec la définition $\mathcal{M}_{p,q}$ (voir page 63). par conséquent $\varphi(H) = H'$ et donc $\psi(Z) = Z'$.

La matrice $[M\overline{M}]$ est une matrice de contraintes de Z sur B. C'est aussi une matrice de contraintes de Z' sur B'. Par application du lemme 5.2, nous obtenons une contradiction avec le fait que $M \neq M'$. Donc R et R' ne peuvent être isomorphes. Par conséquent $|\mathcal{M}_{p,q}| = |\mathcal{G}| \leqslant |RS(\mathcal{G}_{n,d})|$ et d'après le lemme 5.1, $|RS(\mathcal{G}_{n,d})| = 2^{\Omega(pq - p\log p - q\log q)} = 2^{\Omega(n)}$. □

Nous déduisons du théorème 5.3 le résultat suivant :

Théorème 5.4 *Soient $\mathcal{G}_{n,d}$ la classe des graphes d'ordre n et de degré maximum d, et $\mathcal{R}$ la classe des fonctions de routage de plus courts chemins. Pour tout $d \geqslant 3$, $\mathrm{MEM}_{\mathit{globale}}(\mathcal{G}_{n,d}, \mathcal{R}) = \Omega(n)$.*

5.4.4 Limitations des matrices de contraintes

Les preuves des théorèmes 5.1 et 5.3 sont basées sur le fait que pour un facteur d'élongation s fixé, tous les chemins de longueur inférieure à $sd_G(x, y)$ entre les sommets x et y d'un graphe G doivent *commencer* par un certain arc e incident à x. On peut se demander si une autre construction des graphes de contraintes aurait permis de déterminer une borne inférieure de le nombre de fonctions de routage non isomorphes de facteur d'élongation s, pour s quelconque. Il est simple de voir, d'après la définition 5.5, qu'il n'est pas possible de faire le même raisonnement pour $s \geqslant 3$. En effet, quelque soient x, y deux sommets d'un graphe G (x de degré au moins 2) et e un arc incident à x, il existe toujours un chemin de x à y de longueur au plus $3d_G(x, y)$ qui ne commence pas par e. Notons que cette limitation est due en partie au fait que nous n'avons pas restreint la classe des fonctions de routage à celles des fonctions de routage simples, autorisant pas conséquent des chemins de routage contenant des cycles.

La technique qui consiste à énumérer le nombre de fonctions de routage pour une classe de graphes $\mathcal{G}$ particulière et sur une classe de fonctions de routage $\mathcal{R}$ particulière, est une technique générale pour obtenir une borne inférieure de $\mathrm{MEM}_{\mathit{globale}}(\mathcal{G}, \mathcal{R})$. Cependant elle ne permet pas de donner des bornes inférieures optimales sur la mémoire globale de routage pour les classes de graphes de degré maximum borné. En effet, une limite intrinsèque à cette méthode est le nombre de graphes non isomorphes de la classe considérée. Nous allons voir dans le chapitre suivant comment « dépasser » cette limite, notamment en prouvant qu'il existe un graphe d'ordre n pour lequel la mémoire globale de routage requise est plus grande que $\Omega(n^2)$ bits.

CHAPITRE 6

Incompressibilité des tables de routage

6.1 Introduction

Plusieurs compromis entre la mémoire de routage et la longueur des chemins ont été étudiés dans [3, 4, 67]. Cependant, dans ces articles, les bornes inférieures données ne sont pas exactes pour les méthodes de routage de plus courts chemins et ne prennent en particulier pas en compte le degré maximum du réseau, ce qui implique que ces bornes inférieures n'ont que peu de chance d'être optimale pour des réseaux de degré maximum borné (voir [29] pour plus de précisions à propos des bornes inférieures et supérieures de la mémoire de routage).

Dans ce chapitre, nous considérons le problème général de la mémoire de routage requise pour les méthodes de routage de plus courts chemins sur les réseaux de degré maximum d. Nous allons montrer que pour toute méthode de routage et pour toute constante ε, $0 < \varepsilon < 1$, et pour tout entier d tel que $3 \leqslant d \leqslant \varepsilon n$, il existe un réseau de n nœuds de degré maximum d où toute fonction de routage de plus courts chemins nécessite $\Theta(n \log d)$ bits de mémoire locale sur $\Theta(n)$ routeurs. Ceci prouve qu'en général on ne peut pas faire mieux que les tables de routage sur $\Theta(n)$ nœuds du réseau. En d'autres termes, il n'y a pas d'espoir de trouver une méthode de routage universel plus compact que les tables de routage pour les réseaux de degré maximum d. Ce résultat améliore les précédents travaux sur le problème du routage compact présentés dans [27, 57, 67]. Une conséquence immédiate de ce résultat est qu'il existe un réseau de n nœuds de degré maximum 3 qui nécessite $\Omega(n/\log n)$ intervalles sur une fraction constante de ses liens, ce qui améliore d'un facteur $\log n$ les bornes inférieures établies dans [9, 38]. Plus fondamentalement, nous donnons une borne inférieure exacte en $\Theta(n)$ intervalles sur une fraction constante des liens d'un réseau de n nœuds de degré 3, montrant que, dans le pire cas, le routage par intervalles est pire que les tables de routage, puisque les $\Theta(n)$ intervalles sont stockés en utilisant $\Theta(n \log n)$ bits sur chaque routeur.

La section 6.2 présente formellement les principaux résultats de ce chapitre. Dans la section 6.3 nous présentons une généralisation des graphes et matrices de contraintes vus dans les chapitres précédents. Dans la section 6.4, nous effectuons la démonstration du résultat principal de ce chapitre. La section 6.5 présente un résultat général qui permet d'accroître le nombre de nœuds qui requièrent la quantité d'information de routage critique permettant d'établir une borne de la mémoire globale de routage. La section 6.6 étend les résultats précédents pour des facteurs d'élongation < 2. En section 6.7, nous étudions le cas particulier du routage par intervalles, puis dans la dernière section nous présentons l'état de l'art actuel sur les bornes de la mémoire de routage.

Les résultats présentés dans ce chapitre ont été établis en collaboration avec Stéphane Pérénnès, et ont donné lieu aux rapports [40] et [41].

6.2 Borne inférieure de la mémoire de routage

Cette section présente brièvement les principaux résultats de ce chapitre et décrit quelques techniques et définitions utilisées pour prouver les bornes inférieures de la mémoire de routage. Nous allons en particulier montrer le théorème suivant en section 6.4.

Théorème 6.1 *Soit ε, $0 < \varepsilon < 1$, une constante fixée. Pour tout entier d, $3 \leqslant d \leqslant \varepsilon n$, il existe un graphe d'ordre n et de degré maximum d qui requiert une mémoire de routage de $\Theta(n \log d)$ bits sur $\Theta(\log_d n)$ sommets.*

Pour prouver ce résultat, nous proposons la construction d'un réseau dont une petite partie des sommets, nommément $\Theta(\log_d n)$, nécessitent la connaissance d'une grande quantité d'information, nommément $\Omega(n \log n)$ bits, pour router selon les plus courts chemins. Ce graphe d'ordre n et de degré maximum d est noté $G_{n,d}$ dans la suite.

Dans un second temps nous appliquons une transformation générale de graphe, expliquée en section 6.5 qui permet d'accroître le nombre de routeurs ayant besoin de connaître $\Theta(n \log d)$ bits d'information de routage. Cette transformation est appliquée au graphe $G_{\Theta(n),d}$ pour obtenir un nouveau pire cas pour le routage compact, c'est-à-dire un graphe d'ordre n et de degré maximum d noté Split-$G_{n,d}$. Le théorème suivant est démontré en section 6.5.

Théorème 6.2 *Soit ε, $0 < \varepsilon < 1$, une constante fixée. Pour tout entier d, $3 \leqslant d \leqslant \varepsilon n$, il existe un graphe d'ordre n et de maximum degré d qui requiert une mémoire de routage de $\Theta(n \log d)$ bits sur $\Theta(n)$ sommets.*

Ce théorème prouve qu'il existe un cas où la mémoire globale de routage peut être de $\Theta(n^2 \log d)$ bits, impliquant que les tables de routage sont *incompressibles* asymptotiquement dans le cas général, ou encore qu'il n'existe pas d'algorithme distribué de routage *compact* sur ce réseau. Ce théorème peut être considéré comme le point central de ce chapitre et même de ma thèse.

Nous présentons ci-dessous une table résumant les bornes de la mémoire de routage et le nombre de routeurs utilisant une telle mémoire en fonction du degré d et de l'ordre du graphe. La dernière colonne représente la somme de la mémoire de routage sur tous les routeurs du graphe. Il est bien connu que tout graphe connexe de degré maximum 2 (les chemins P_n et les cycles C_n) possède une mémoire de routage de $O(\log n)$ bits sur tous ses routeurs. En effet la méthode du routage par intervalles avec un intervalle par lien est applicable (voir [80]). Ci-dessous ε est une constante arbitraire telle que $0 < \varepsilon < 1$.

degré maximum	bits mémoire	# routeurs	graphes	total des bits mémoire
2	$O(\log n)$	n	P_n, C_n	$O(n \log n)$
$3 \leqslant d \leqslant \varepsilon n$	$\Theta(n \log d)$	$\Theta(\log_d n)$	$G_{n,d}$	$\Omega(n \log n)$
$3 \leqslant d \leqslant \varepsilon n$	$\Theta(n \log d)$	$\Theta(n)$	Split-$G_{n,d}$	$\Theta(n^2 \log d)$

6.3 Matrices et graphes de contraintes généralisés

Nous reprenons ici les notions de matrices et de graphes de contraintes introduites au chapitre 4 afin de les généralisées.

L'idée principale de notre borne inférieure est basée sur le fait que dans certain graphe G, certains sommets peuvent être connectés deux à deux par un unique plus courts chemin. D'une certaine façon, les communications entre ces sommets sont *contraintes*, au sens où toute fonction de routage sur G doit sélectionner le même arc pour des communications entre ces paires de sommets. Un ensemble de contraintes est une quantité d'information intrinsèquement nécessaire au routage sur G, à une numérotation des sommets et des arcs près. Partant de cette constatation, nous tentons de construire des graphes à partir d'ensemble de contraintes. Plus précisemment, pour chaque bloc de contraintes, appelés *matrice de contraintes*, nous construisons un graphe qui possède un ensemble unique de plus courts chemins satisfaisant les contraintes décrites par la matrice. Nous montrons que certains ensembles de contraintes possèdent une grande quantité d'informations (due à une complexité inhérente à la matrice) par rapport à la taille du graphe construit. Nous allons montrer que toute fonction de routage sur un tel graphe nécessite de stocker cette quantité d'information.

Définition 6.1 (matrice de contraintes) *Soit $G = (V, E)$ un graphe. Une* matrice de contraintes *de G est une matrice $p \times q$ entière $M = (m_{i,j})$, $i \in I = \{1, \dots, p\}$ et $j \in J = \{1, \dots, q\}$, telle qu'il existe deux ensembles de sommets $A = \{a_i \,|\, i \in I\} \subset V$ et $B = \{b_j \,|\, j \in J\} \subset V$, vérifiant que, pour toute fonction de routage de plus courts chemins $R = \{(P_x, H_x) \,|\, x \in V\}$ sur G, il existe une numérotation $\mathcal{L}$ des sommets de B et une numérotation des arcs des sommets de A telle que :*

$$\forall (i,j) \in I \times J, \quad m_{i,j} = P_{a_i}(0, \mathcal{L}(b_j))$$

De manière informelle, si $M = (m_{i,j})$ est une matrice de contraintes d'un graphe G, la définition 6.1 signifie qu'il existe deux sous-ensembles de sommets A et B tels que tout plus court chemin de $a_i \in A$ à $b_j \in B$ dans G commence par l'arc $e_{i,j}$, lequel peut être numéroté par le port $m_{i,j}$. Dans la suite, les sommets de A sont appelés les *sommets contraints* et les sommets de B les *sommets cibles* pour M. Les permutations ψ_i et les composantes $m_{i,j}$ de M peuvent être vues comme les numérotations des ports des sommets contraints.

Considérons le graphe de Petersen de la figure 6.1, où les ports sont numérotés dans $\{1, 2, 3\}$. La matrice 5×5 $M = (m_{i,j})$ est une matrice de contraintes des sommets contraints $\{a_1, \dots, a_5\}$ vers les sommets cibles $\{b_1, \dots, b_5\}$. Cela signifie donc que quel que soit la fonction de routage de plus courts chemins sur le graphe de Petersen, il est toujours possible de numéroter les arcs des sommets contraints et les sommets cibles, tels que le routage d'un message de a_i vers b_j utilise le port $m_{i,j}$. Dans l'exemple de la figure 6.1 l'unique plus court chemin entre a_3 et b_2 utilise le port 1.

Définition 6.2 ($\sim$ Relation) *Soit $M = (m_{i,j})$ et $M' = (m'_{i,j})$ deux matrices $p \times q$ entières, $i \in \{1, \dots, p\}$ et $j \in \{1, \dots, q\}$. On note $M \sim M'$ si et seulement si il existe deux permutations r de $\{1, \dots, p\}$ et c de $\{1, \dots, q\}$, et p permutations ψ_i de $\{1, \dots, |\cup_{j \in \{1, \dots, q\}} \{m_{i,j}\}|\}$, $i \in \{1, \dots, p\}$, telles que pour tout $i \in \{1, \dots, p\}$ et pour tout $j \in \{1, \dots, q\}$:*

$$m'_{i,j} \;=\; \psi_i(m_{r(i),c(j)})$$

$$M = \begin{array}{c} \begin{array}{ccccc} b_1 & b_2 & b_3 & b_4 & b_5 \end{array} \\ \left|\begin{array}{ccccc} 1 & 1 & 2 & 2 & 3 \\ 1 & 1 & 2 & 2 & 3 \\ 1 & 1 & 2 & 2 & 3 \\ 1 & 1 & 2 & 2 & 3 \\ 1 & 2 & 2 & 1 & 3 \end{array}\right| \end{array} \begin{array}{c} \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{array}$$

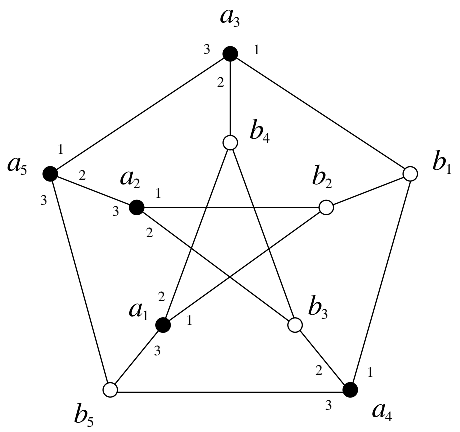


FIG. 6.1 – *Une matrice de contraintes généralisée pour le graphe de Petersen.*

La relation $\sim$ est une relation d'équivalence puisque r, c et ψ_i, $i \in \{1,\ldots,p\}$, sont des bijections. $M' \sim M$ signifie que M' est obtenue à partir de M par permutations de lignes et des colonnes, et par permutations de l'ensemble des composantes de chaque ligne de M.

D'après les définitions ci-dessus, M et M' sont deux matrices de contraintes de G pour les sommets contraints A et les sommets cibles B, si et seulement si $M \sim M'$. Les permutations ψ_i peuvent être vues comme une numérotation des ports alors que r et c peuvent être vues comme une numérotation des sommets de G (précisément la numérotation des sommets contraints et cibles).

Notation. Pour tout ensemble $\mathcal{M}$ de matrices $p{\times}q$ entières, nous notons par $\mathcal{M}_{/\sim}$ l'ensemble des classes d'équivalence de $\mathcal{M}$ sous la relation d'équivalence $\sim$.

Définition 6.3 (graphe de contraintes) *Soit $\mathcal{M}$ un ensemble de matrices $p \times q$ entières. Une classe $\mathcal{G}$ de* graphes de contraintes *de $\mathcal{M}$ est un ensemble de graphes telle que pour tout $M \in \mathcal{M}$ il existe un graphe $G \in \mathcal{G}$ qui possède M comme matrice de contraintes.*

Le lemme suivant établit la propriété fondamentale des graphes de contraintes pour la mémoire de routage. Il exprime le fait que certains graphes de contraintes de $\mathcal{M}$ doivent posséder une mémoire de routage dépendante de $|\mathcal{M}_{/\sim}|$ et partagée par les sommets contraints.

Lemme 6.1 *Soit $\mathcal{M}$ un ensemble de matrices $p \times q$ entières. Pour toute classe $\mathcal{G}$ de graphes de contraintes de $\mathcal{M}$ et d'ordre n, il existe un graphe $G \in \mathcal{G}$ tel que la somme de la mémoire de routage sur p sommets de G est au moins*

$$\log_2 \frac{|\mathcal{M}_{/\sim}|}{\binom{n}{q}} - O(\log n)$$

Preuve. Soit (p,q) une paire d'entiers et soit $\mathcal{M}$ un ensemble de matrices $p{\times}q$ entières. Soit $\mathcal{G}$ une classe de graphes de contraintes de $\mathcal{M}$ et d'ordre n. De manière similaire à la définition 5.6 du chapitre 5, on définit l'indice d'une matrice $M \in \mathcal{M}$ comme l'entier obtenu en concaténant toutes les colonnes de M, à la différence que l'on choisit ici l'entier q comme base, au lieu de 2. Soit C_μ la fonction qui à tout $M \in \mathcal{M}$ associe le représentant canonique de M de $\mathcal{M}_{/\sim}$

défini d'après la définition 5.6. Pour tout $M \in \mathcal{M}$, il existe $G = (V, E) \in \mathcal{G}$ et il existe deux ensembles de sommets $A, B \subset V$ de tailles respectives p et q, tels que, pour toute fonction de routage sur G, A soit l'ensemble des sommets contraints et B l'ensemble des sommets cibles pour M. Soit $R = \{R_x = (P_x, H_x) \mid x \in V\}$ une fonction de routage de plus courts chemins sur G, soit $R_A = \{R_a \mid a \in A\}$ la collection des fonctions locales de routage des sommets contraints, et f_B la fonction $f_B : \{1, \ldots, n\} \to \{0, 1\}$ telle que $f_B(x) = 1$ si et seulement si il existe un sommet $b \in B$ numéroté x. Montrons que la connaissance de C_μ, R_A, f_B et de $O(\log n)$ bits est suffisante pour calculer $C_\mu(M)$. Supposons qu'un observateur connaisse seulement les fonctions C_μ, R_A, f_B et la procédure suivante :

« *Ordonner l'ensemble des fonctions de ports de* R_A *d'une manière arbitraire* $\{P_1, \ldots, P_p\}$.
Poser $j \leftarrow 0$.
Pour i *de* 1 *à* p *faire*
Pour b *de* 1 *à* n *faire*
si $f_B(b) = 1$ *alors poser* $j \leftarrow j + 1$ *et*
poser $M'_{i,j} \leftarrow P_i(0, b)$.
Poser $M'' \leftarrow C_\mu(M')$. »

L'observateur possède suffisamment d'informations pour calculer une matrice $M' \sim M$. En effet, la procédure précédente teste les réponses des fonctions locales de ports des sommets contraints sur les numéros des sommets cibles de G. Donc $M'' = C_\mu(M)$. La procédure, exceptant C_μ, f_B et les fonctions P_i, peut être décrite avec au plus $O(\log n)$ bits (la procédure est décrite connaissant les valeurs de n, p et q, que l'on peut représenter chacune avec au plus $O(\log n)$ bits). $O(\log n)$ bits sont aussi suffisants pour décrire C_μ. En effet C_μ ne dépend que des dimensions de M, p et $q \leqslant n$. Il découle qu'un simple algorithme, paramétré par p et q, permet de calculer l'indice μ (voir définition 5.6) de toutes les matrices équivalentes à M sous $\sim$, et donc de trouver la matrice qui le maximise. De plus, toute fonction caractéristique f_Y d'un sous-ensemble $Y \subset X$ peut être décrite avec au plus $\log_2 \binom{|X|}{|Y|} + O(\log |X|)$ bits, pour tout ensemble fini X. Autrement dit, la fonction f_B peut être décrite par au plus $\log_2 \binom{n}{q} + O(\log n)$ bits. Soit $m = \sum_{a \in A} \mathrm{MEM}(G, R, a)$. Puisque le calcul de $C_\mu(M)$ ci-dessus peut être fait pour toute matrice $M \in \mathcal{M}$, nous concluons, par un argument de comptage, qu'il existe une matrice canonique $C_\mu(M_0) \in \mathcal{M}$ nécessitant au moins $\log_2 |\mathcal{M}_{/\sim}|$ bits pour être stockée. Par conséquent $m + \log_2 \binom{n}{q} + O(\log n) \geqslant \log_2 |\mathcal{M}_{/\sim}|$ car sinon la procédure précédente suffirait à calculer $C_\mu(M_0)$ avec moins de $\log_2 |\mathcal{M}_{/\sim}|$ bits. Cela signifie donc que la somme de la mémoire de routage sur p routeurs est d'au moins m bits où m vérifie $m \geqslant \log_2(|\mathcal{M}_{/\sim}| / \binom{n}{q}) - O(\log n)$. □

6.4 Constructions de graphes de contraintes

Nous allons construire des graphes de contraintes d'ordre n et de degré maximum d pour une classe de matrices de contraintes que nous préciserons plus loin. Le lemme 6.1 permettra de déduire une borne inférieure de la mémoire de routage d'un graphe particulier d'ordre n et de degré maximum d.

Afin de simplifier la preuve du théorème 6.1, nous donnons tout d'abord une construction pour tout degré d tel que $3 \leqslant d \leqslant n^{1/3}$ et pour $n = \alpha(d-1)^p$, où α une constante $\geqslant 10$ et où p est un entier $\geqslant 2$. Nous généralisons ensuite la construction pour tout n et tout degré d tel que $3 \leqslant d \leqslant \varepsilon n$, où ε est une constante positive fixée et < 1.

6.4.1 Construction pour $n = (d-1)^p$

Pour tout couple d'entiers (p, δ), soit $A_{p,\delta} = (a_{i,j})$ la matrice entière $p \times \delta^p$ telle que $a_{i,j} = (\lfloor (j-1)/\delta^{i-1} \rfloor \bmod \delta) + 1$, $i \in \{1, \ldots, p\}$ et $j \in \{1, \ldots, \delta^p\}$. Chaque colonne de $A_{p,\delta}$ peut être vue comme un mot de longueur p sur l'alphabet $\{1, \ldots, \delta\}$, et l'ensemble des colonnes peut être vu comme l'ensemble de tous les mots de longueur p sur l'alphabet $\{1, \ldots, \delta\}$. Par exemple

$$A_{2,3} = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 \\ 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 \end{bmatrix}$$

Lemme 6.2 *Pour tout couple d'entiers (p, δ), $p \geqslant 1$ et $\delta \geqslant 2$, il existe un graphe $H_{p,\delta}$ qui satisfait :*

i. *$H_{p,\delta}$ a au plus $5\delta^p$ sommets*
ii. *$H_{p,\delta}$ est de degré maximum $\delta + 1$*
iii. *$H_{p,\delta}$ possède $A_{p,\delta}$ comme matrice de contraintes*

Preuve. Soit (p, δ) un couple d'entiers tel que $p \geqslant 1$ and $\delta \geqslant 2$. Le graphe $H_{p,\delta}$ est défini par récurrence sur p. $H_{1,\delta}$ est isomorphique à $K_{1,\delta}$. Clairement les points (i), (ii) et (iii) sont satisfaits puisque la matrice $A_{1,\delta} = [\,1\,2 \ldots \delta\,]$ est une matrice de contraintes de $H_{p,\delta}$. Plus précisément, $A_{1,\delta}$ est une matrice de contraintes avec un sommet contraint, numéroté 1 et formant le singleton $A_1 = \{1\}$, qui forme la racine de $K_{1,\delta}$, et avec les sommets cibles formant un ensemble $B_1 = \{1, \ldots, \delta\}$ qui sont les feuilles de $K_{1,\delta}$.

Supposons maintenant que $H_{p,\delta}$ possède un ensemble de sommets contraints $A_p = \{1, \ldots, p\}$ et un ensemble de sommets cibles B_p numérotés par tous les mots de longueur p sur l'alphabet $\{1, \ldots, \delta\}$. De plus, supposons que les propriétés (i), (ii) et (iii) soient satisfaites. Dans la suite, nous notons par $T_{p,\delta}$, l'arbre δ-aire de hauteur p et de δ^p feuilles. L'ordre de $T_{p,\delta}$, noté par $|T_{p,\delta}|$, vaut $(\delta^{p+1} - 1)/(\delta - 1)$. $H_{p+1,\delta}$ est composé d'une copie de $H_{p,\delta}$ et d'une copie de $T_{p,\delta}$ connectées par $\delta^p + \delta^{p+1}$ sommets intermédiaires comme montré sur la partie gauche de la figure 6.2. Pour le cas particulier $H_{2,\delta}$, nous n'ajoutons pas les premiers $\delta^p = \delta$ sommets intermédiaires. La racine de $T_{p,\delta}$ est numérotée $p + 1$. Elle forme, avec l'ensemble A_p, le nouvel ensemble de sommets contraints $A_{p+1} = A_p \cup \{p + 1\}$. Chacun des δ^p premiers sommets intermédiaires est connecté par une arête à un unique sommet de B_p, formant un nouvel ensemble de sommets cibles notés B'_p numérotés comme B_p. Les δ^{p+1} autres sommets forment l'ensemble B_{p+1} des sommets cibles de $H_{p+1,\delta}$ numérotés par tous les mots de longueur $p + 1$ sur l'alphabet $\{1, \ldots, \delta\}$. Les δ^p feuilles de $T_{p,\delta}$ sont partionnées en δ sous-ensembles. Une feuille x est numérotée i, $i \in \{1, \ldots, \delta\}$, si le i-ième fils de la racine de $T_{p,\delta}$ est un ancêtre de x. Chaque sommet $w = ui$ de B_{p+1}, où u est un mot de longueur p et i une lettre de l'alphabet $\{1, \ldots, \delta\}$, est de degré deux et est connecté à $u \in B'_p$ et à une feuille x numérotée i telle que x soit connecté au plus à δ sommets de l'ensemble B_{p+1}. Par conséquent, tout couple (u, x), où $u \in B'_p$ et x est une feuille de $T_{p,\delta}$, est connecté avec un unique sommet $w = ui \in B_{p+1,\delta}$ où i est le numéro de x (voir la partie droite de la figure 6.2).

Notez que, sur la figure 6.2, selon la numérotation utilisée, le plus court chemin allant au sommet cible $xyz \in B_3$ doit traverser successivement le port x depuis de sommet contraint $1 \in A_3$, le port y depuis de sommet 2 et le port z depuis le sommet 3.

Le degré de $H_{p,\delta}$ est $\delta + 1$, dû aux feuilles de $T_{p,\delta}$, prouvant (ii). L'ordre de $H_{p+1,\delta}$ est[1] $|H_{p+1,\delta}| = |H_{p,\delta}| + |T_{p,\delta}| + \delta^p + \delta^{p+1}$, *i.e.* $|H_{p+1,\delta}| = |H_{1,\delta}| + \sum_{i=1}^{p}(|T_{i,\delta}| + \delta^i + \delta^{i+1}) = 1+\delta+\sum_{i=1}^{p}((\delta^{i+1}-1)/(\delta-1)+\delta^i+\delta^{i+1}) = 1+\delta+\delta(\delta^p-1)(\delta^2+\delta-1)/(\delta-1)^2-p/(\delta-1) \leqslant 5\delta^{p+1}$, pour tout $p \geqslant 1$ et $\delta \geqslant 2$, prouvant (i). En considérant les sommets contraints A_{p+1} et les sommets cibles B_{p+1}, la matrice $A_{p+1,\delta}$ est une matrice de contraintes de $H_{p+1,\delta}$ car pour tout sommet $w = ui \in B_{p+1}$ est connecté par un unique chemin à B'_p. Donc, par hypothèse, tout $w \in B_{p+1}$ est connecté par un unique plus court chemin à un sommet quelconque de A_p. De plus, w est aussi connecté par un unique plus court chemin au i-ième fils de $T_{p,\delta}$. Ceci prouve (iii) et termine la preuve. □

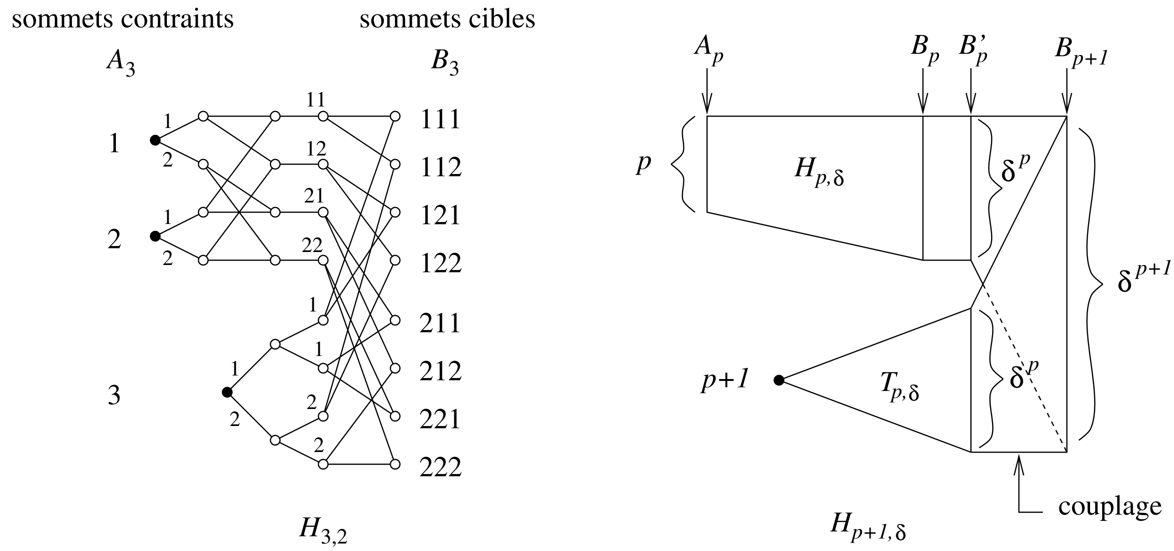


FIG. 6.2 – *Le graphe $H_{p,\delta}$ du lemme 6.2. Le plus courts chemins allant au sommet cible $xyz \in B_3$ doit traverser successivement le port x depuis de sommet contraint $1 \in A_3$, le port y depuis de sommet 2 et le port z depuis le sommet 3.*

Lemme 6.3 *Pour tout couple d'entiers (p, δ), soit $\mathcal{A}_{p,\delta}$ l'ensemble des matrices entières $2p\times\delta^p$ de la forme $\left[\begin{smallmatrix} A_{p,\delta} \\ B \end{smallmatrix}\right]$, où $B \sim A_{p,\delta}$. Alors, pour tout $p \geqslant 2$ et pour tout $\delta \geqslant 2$,*

$$\frac{(\delta^p)!}{(\delta!)^{2p}(2p)!} \leqslant |\mathcal{A}_{p,\delta}{}_{/\sim}| = 2^{\Omega(\delta^p \log \delta^p)}$$

Preuve. Soit (p, δ) une paire d'entiers $\geqslant 2$ et soit $\mathcal{A} = \{M \mid M \sim X, X \in \mathcal{A}_{p,\delta}\}$. Alors $|\mathcal{A}| \geqslant (\delta^p!)^2$, car toute matrice de la forme $M = \left[\begin{smallmatrix} \pi_1(A_{p,\delta}) \\ \pi_2(A_{p,\delta}) \end{smallmatrix}\right]$, où π_1, π_2 sont deux permutations des colonnes de chaque sous-matrice $A_{p,\delta}$, est dans $\mathcal{A}$. Pour le vérifier, il suffit d'appliquer π_1^{-1} sur les colonnes de M pour avoir $\left[\begin{smallmatrix} A_{p,\delta} \\ \pi_1^{-1}(\pi_2(A_{p,\delta})) \end{smallmatrix}\right]$. Par conséquent $|\mathcal{A}_{/\sim}| \geqslant (\delta^p!)^2/((\delta^p)!(\delta!)^{2p}(2p)!)$, où $(\delta^p)!(\delta!)^{2p}(2p)!$ est la taille maximum d'une classe d'équivalence de matrices entières $2p\times\delta^p$ à valeur dans $\{1,\ldots,\delta\}$ sous $\sim$ (clairement $\mathcal{A}$ est un ensemble de matrices à valeur dans $\{1,\ldots,\delta\}$).

1. Selon la figure 6.2, pour $p \geqslant 2$ nous pouvons effacer δ sommets intermédiaires, ce qui ne change pas le nombre de sommets de $H_{p+1,\delta}$ en ordre de grandeur.

De plus $|\mathcal{A}_{/\sim}| = |\mathcal{A}_{p,\delta_{/\sim}}|$, donc $|\mathcal{A}_{p,\delta_{/\sim}}| \geqslant (\delta^p)!/((\delta!)^{2p}(2p)!)$. Posons $L = \log_2 |\mathcal{A}_{p,\delta_{/\sim}}|$. On obtient $L = \Omega(p\delta^p \log \delta - p\delta \log \delta - p \log p) = \Omega(p((\delta^p - \delta)\log \delta - \log p))$. Puisque $\delta \geqslant 2$ et $p \geqslant 2$, on conclut $L = \Omega(p\delta^p \log \delta) = \Omega(\delta^p \log \delta^p)$. □

Proposition 6.1 *Soient α une constante réelle, $\alpha \geqslant 10$, et p une constante entière, $p \geqslant 2$. Pour tout entier d, $3 \leqslant d \leqslant n^{1/3}$, il existe un graphe d'ordre $n = \alpha(d-1)^p$ et de degré maximum d ayant une mémoire de routage de $\Theta(n \log d)$ bits par sommets sur $\Theta(\log_d n)$ sommets.*

Pour les degrés maximum de $\Theta(n^{1/3})$ cette proposition montre déjà qu'il existe un graphe qui a une mémoire de routage de $\Theta(n \log n)$ bits, bien que ce soit pour des valeurs particulières de n.

Preuve. Soit n un entier suffisamment grand de la forme $n = \alpha(d-1)^p$, où α est une constante $\geqslant 10$ et p un entier $\geqslant 2$. Soit d un entier quelconque satisfaisant $3 \leqslant d \leqslant n^{1/3}$. Nous construisons un ensemble de graphes de contraintes d'ordre n et de degré maximum d à partir de l'ensemble de matrices $\mathcal{A}_{p,d-1}$ (introduit au lemme 6.3) et avec $2p$ sommets contraints. En première approximation, en appliquant le lemme 6.1 et le lemme 6.3, la somme de la mémoire de routage sur ces $2p$ sommets est $\Omega((d-1)^p \log{(d-1)^p}) - O(n)$, car $\log_2 \binom{n}{x} \leqslant n$ pour tout x. Puisque $n = \alpha(d-1)^p$ avec α constante, il suit que $n = \Theta((d-1)^p)$. Par conséquent la somme de la mémoire de routage est $\Omega(n \log n)$ bits sur $2p = \Theta(\log_{d-1} n) = \Theta(\log_d n)$ sommets, pour $d \geqslant 3$.

Plus précisément, soit $\delta = d-1$ et $p = \log_\delta(n/\alpha)$. Soit M une matrice de $\mathcal{A}_{p,\delta}$. Soit H_1 et H_2 deux copies du graphe $H_{p,\delta}$ (défini au lemme 6.2) de sommets contraints A_1, A_2 et de sommets cibles B_1, B_2 respectively. De tels graphes existent car $\delta \geqslant 2$ et $\delta^2 \leqslant n^{2/3} \leqslant n/\alpha$ implique $2 \leqslant \log_\delta(n/\alpha) = p$. Le graphe est composé de H_1 et H_2 connectés par un couplage parfait entre les sommets B_1 et B_2 (voir la figure 6.2), plus quelques sommets supplémentaires. Plus précisément $w_1 \in B_1$ est connecté à $w_2 \in B_2$ si et seulement si le vecteur $w_1 w_2$ formé par la concaténation des mots w_1 et w_2 est une colonne de M (rappelons que les sommets de B_1 ou de B_2 sont numérotés par les mots de longueur p sur l'alphabet $\{1, \ldots, \delta\}$). Par construction de $H_{p,\delta}$, les sommets de A_1 sont des sommets contraints pour les sommets $B_1 \cup B_2$ car B_1 et B_2 sont connectés par une unique arête du couplage parfait. Par symétrie A_2 est aussi un ensemble de sommets contraints pour $B_1 \cup B_2$. En définitive, $A_1 \cup A_2$ est un ensemble de sommets contraints pour l'ensemble de sommets B_1. On peut vérifier que M est une matrice de contraintes de sommets contraints $A_1 \cup A_2$ sur les sommets cibles B_1. D'après le lemme 6.2, l'ordre de ce graphe est m avec $m \leqslant 2 \cdot 5\delta^p \leqslant \alpha\delta^p = n$. Nous ajoutons un chemin de longueur $n - m$ à un sommet quelconque n'appartenant pas à $A_1 \cup A_2$ afin d'obtenir un graphe d'ordre n exactement. Clairement, l'addition de ce chemin ne modifie par les contraintes entre les sommets contraints et les sommets cibles. Par hypothèse, $\delta^p = n/\alpha = \Theta(n)$ et $p = \Theta(\log_\delta n) = \Theta(\log_d n)$, pour $d \geqslant 3$. Puisque $p \geqslant 2$ et $\delta \geqslant 2$, nous pouvons appliquer le lemme 6.3 et le lemme 6.1. Il existe aussi un graphe noté $G_{n,d}$ tel que la somme de sa mémoire de routage sur $2p$ sommets contraints est au moins $\Omega(\delta^p \log \delta^p) - O(n) = \Omega(n \log n)$ bits. Donc, la mémoire de routage de $G_{n,d}$ est $\Omega(n \log n)/(2p) = \Omega(n \log d)$ bits sur au moins un sommet. Puisque cette mémoire de routage est au plus $O(n \log d)$ bits, en utilisant les tables de routage, il suit qu'en fait $\Theta(p) = \Theta(\log_d n)$ sommets ont une mémoire de routage de $\Theta(n \log d)$ bits chacun. □

Notez que ce résultat est également vrai pour tout degré maximum d vérifiant $3 \leqslant d \leqslant n^{1/2-\varepsilon}$ où ε est une constante vérifiant $0 < \varepsilon < 1/2$. Quoiqu'il en soit, le Théorème 6.1 va établir un résultat plus général dans le prochain paragraphe.

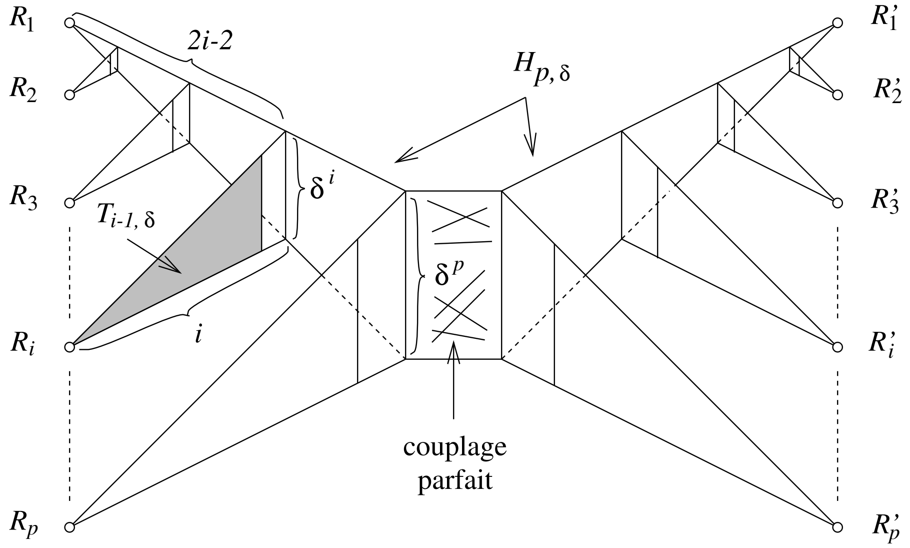


FIG. 6.3 – *Le graphe $G_{n,d}$ est un graphe de contraintes d'ordre n et de degré maximum $d = \delta+1$. Les routeurs $R_1, \ldots, R_p$ et $R'_1, \ldots, R'_p$ « calculent » le couplage parfait entre les $2\delta^p = \Theta(n)$ sommets.*

6.4.2 Construction pour tout n et pour tout d

Nous généralisons les matrices $A_{p,\delta}$ par des matrices $A_{p,\delta,k}$ de la manière suivante : $A_{p,\delta,k}$ est une matrice entière $(p+1)\times(k\delta^p)$ composée de k matrices $A_{p,\delta}$, $[\,A_{p,\delta} \mid A_{p,\delta} \mid \ldots \mid A_{p,\delta}\,]$, et d'une ligne supplémentaire $[\,1\,1\ldots1\,|\,2\,2\ldots2\,|\ldots|\,k\,k\ldots k\,]$. Chaque colonne de la matrice $A_{p,\delta,k}$ peut être vue comme un mot de longueur $p+1$ où le premier symbole appartient à l'alphabet $\{1,\ldots,k\}$ et les p derniers à l'alphabet $\{1,\ldots,\delta\}$. Notez que $A_{p,\delta,\delta} = A_{p+1,\delta}$. Par exemple

$$A_{2,3,2} = \left[\begin{array}{ccccccccc|ccccccccc} 1&1&1&1&1&1&1&1&1&2&2&2&2&2&2&2&2&2 \\ \hline 1&1&1&2&2&2&3&3&3&1&1&1&2&2&2&3&3&3 \\ 1&2&3&1&2&3&1&2&3&1&2&3&1&2&3&1&2&3 \end{array}\right]$$

Lemme 6.4 *Pour tout triplé d'entiers (p,δ,k), $p \geqslant 1$ et $\delta \geqslant k \geqslant 2$, il existe un graphe $H_{p,\delta,k}$ qui satisfait :*

i. *$H_{p,\delta,k}$ a au plus $(k+8)\delta^p$ sommets*
ii. *$H_{p,\delta,k}$ est de degré maximum $\delta+1$*
iii. *$H_{p,\delta,k}$ possède $A_{p,\delta,k}$ comme matrice de contraintes*

Preuve. Le graphe $H_{p,\delta,k}$ est similaire au graphe $H_{p+1,\delta}$ où l'arbre $T_{p,\delta}$, qui était interconnecté au graphe $H_{p+1,\delta}$, est remplacé par un nouvel arbre $Q_{p,\delta,k}$, ayant une sa racine de degré k. Plus précisément, $H_{p,\delta,k}$ est composé d'une copie de $H_{p,\delta}$ auquel on a ajouté δ^p arêtes aux sommets cibles de $H_{p,\delta}$, d'un arbre $Q_{p,\delta,k}$ et de $k\delta^p$ sommets (qui sont les sommets cibles de $H_{p,\delta,k}$). L'arbre $Q_{p,\delta,k}$ est composé de k sous-arbres δ-aires, chacun ayant δ^{p-1} feuilles toutes à égale distance de leur racine. Ces k sous-arbres sont connectés à un seul sommet de degré k formant la racine de $Q_{p,\delta,k}$. L'ordre de $Q_{p,\delta,k}$ est $k(\delta^p - 1)/(\delta - 1) + 1$.

Les connexions entre $H_{p,\delta}$ et $Q_{p,\delta,k}$ sont les suivantes: parmi les $k\delta^p$ sommets de $H_{p,\delta,k}$, chaque sommet cible est connecté exactement à un sommet cible de $H_{p,\delta}$ et à une feuille de $Q_{p,\delta,k}$. Toute feuille de $Q_{p,\delta,k}$ est connectée à δ sommets cibles de $H_{p,\delta,k}$ (voir la figure 6.4). L'ordre de $H_{p,\delta,k}$ est $|H_{p,\delta}| + |Q_{p,\delta,k}| + \delta^p + k\delta^p \leqslant 5\delta^p + k(\delta^p - 1)/(\delta - 1) + 1 + (k+1)\delta^p \leqslant \delta^p(k + 6 + k/(\delta - 1)) + 1 - k/(\delta - 1) \leqslant (k+8)\delta^p$, pour tout $\delta \geqslant k \geqslant 2$ et $p \geqslant 2$. On peut vérifier que $A_{p,\delta,k}$ est une matrice de contraintes de $H_{p,\delta,k}$ par une analyse similaire à celle de la preuve du lemme 6.2. En effet les sommets cibles de $H_{p,\delta,k}$ peuvent être vus comme les mots de longueurs $p+1$ où le premier symbole est dans $\{1, \ldots, k\}$ et des p suivants dans $\{1, \ldots, \delta\}$. La figure 6.4 montre un graphe $H_{2,3,2}$ qui possède comme matrice de contraintes la matrice $A_{2,3,2}$, représentée page 79. $\square$

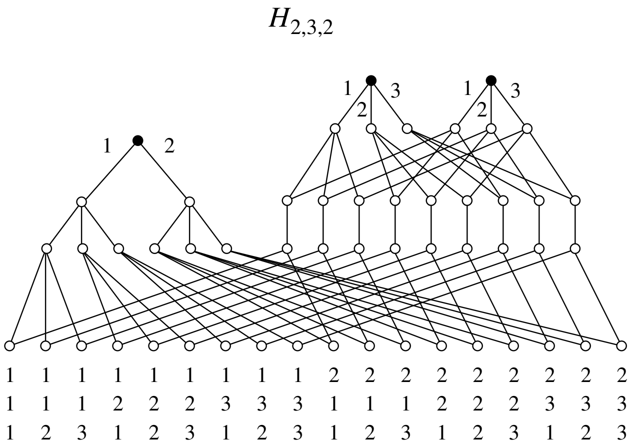


FIG. 6.4 – *Le graphe $H_{p,\delta,k}$, pour $p = 2$, $\delta = 3$ et $k = 2$.*

Lemme 6.5 *Pour tout triplé d'entiers (p, δ, k), soit $\mathcal{A}_{p,\delta,k}$ l'ensemble des matrices entières $(2p+2) \times (k\delta^p)$ de la forme $\begin{bmatrix} A_{p,\delta,k} \\ A \end{bmatrix}$ où $A \sim A_{p,\delta,k}$. Alors, pour tout $p \geqslant 2$ et pour tout $\delta \geqslant k \geqslant 2$,*

$$\frac{(k\delta^p)!}{(k!)^2(\delta!)^{2p}(2p+2)!} \leqslant |\mathcal{A}_{p,\delta,k/\sim}| = 2^{\Omega(k\delta^p \log(k\delta^p))}$$

Preuve. Similaire à la preuve du lemme 6.3. $\square$

Remarque. $|\mathcal{A}_{p,\delta,k/\sim}| = 2^{\Omega(k\delta^p \log(k\delta^p))}$ également pour $p = 1$ et pour tout $\delta \geqslant k \geqslant 3$.

Théorème 6.1 *Soit ε, $0 < \varepsilon < 1$, une constante fixée. Pour tout entier d, $3 \leqslant d \leqslant \varepsilon n$, il existe un graphe d'ordre n et de degré maximum d qui requiert une mémoire de routage de $\Theta(n \log d)$ bits sur $\Theta(\log_d n)$ sommets.*

Preuve. Soient $\varepsilon \in]\,0,1[$ une constante réelle, n un entier suffisamment grand et d un entier tel que $3 \leqslant d \leqslant \varepsilon n$. De manière similaire à la preuve de la proposition 6.1, nous construisons un graphe de contraintes d'ordre n et de degré maximum d. Nous le notons également $G_{n,d}$ car, pour les petites valeur de d, la proposition 6.1 peut être vue comme un cas particulier de ce théorème. Notre construction dépend du degré d.

Petit degré: $3 \leqslant d \leqslant \sqrt{n/10}$

Soient $\delta = d - 1$ et p le plus grand entier $\geqslant 2$ tel que $\lfloor n/\delta^p \rfloor \geqslant 10$. Soit $k = \lfloor n/\delta^p \rfloor - 8$. L'entier $p \geqslant 2$ existe, parce que pour $p = 2$ nous avons en particulier $\lfloor n/\delta^p \rfloor > n/d^2 - 1 \geqslant n/\sqrt{n/10}^2 - 1 = 9$, et $\lfloor n/\delta^p \rfloor > 9$ implique $\lfloor n/\delta^p \rfloor \geqslant 10$. En conséquence, $2 \leqslant k \leqslant \delta = d - 1$. Le graphe $G_{n,d}$ est composé de deux copies du graphe $H_{p,\delta,k}$ connectés par couplage parfait. De manière similaire à la preuve de la proposition 6.1, $G_{n,d}$ est un graphe de contraintes de $\mathcal{A}_{p,\delta,k}$ (voir lemme 6.5). Il a $2p$ sommets contraints. Par définition $(k+8)\delta^p = \lfloor n/\delta^p \rfloor \delta^p = n - r$, où $r = (n \bmod \delta^p) \in \{0, \ldots, n-1\}$ car $n \geqslant \delta^p$. Donc $(k+8)\delta^p \leqslant n < 2(k+8)\delta^p$ et par conséquent $k\delta^p = \Theta(n)$. Donc, $p = \Theta(\log_\delta(n/k)) = \Theta(\log_\delta n - \log_\delta k) = \Theta(\log_d n)$, car $2 \leqslant k \leqslant \delta$ et $\log \delta = \Theta(\log d)$. Par application du lemme 6.1 et du lemme 6.5, on obtient que la somme de la mémoire de routage sur $2p$ sommets est $\Omega(k\delta^p \log(k\delta^p)) - O(n) = \Omega(n \log n)$ bits.

Grand degré: $\sqrt{n/10} \leqslant d \leqslant \varepsilon n$

Soient $n' = \lfloor (1-\varepsilon)n \rfloor$ et $\delta = \lfloor \sqrt{n'/10} \rfloor - 1$. La construction de $G_{n,d}$ pour d grand consiste en un graphe $G_{n',\delta+1}$, où nous ajoutons un arbre $K_{1,d-1}$ connecté par sa racine avec une arête à un quelconque des sommets de $G_{n',\delta+1}$ qui n'est pas un sommet contraint. Le degré maximum de $G_{n',\delta+1}$ est au plus $\delta + 1 \leqslant \sqrt{n'/10} \leqslant d$. Clairement le degré maximum du graphe total est d. Cela est dû à la racine de $K_{1,d-1}$. Les plus courts chemins entre les $2p'$ sommets contraints de $G_{n',\delta+1}$ ne sont pas modifiés par l'addition du $K_{1,d-1}$. Nous pouvons ajouter, par exemple à une feuille de $K_{1,d-1}$, un chemin de longueur $n - (n' + d) = n - (\lfloor (1-\varepsilon)n \rfloor + \varepsilon n) \geqslant 0$ pour obtenir un graphe d'ordre n. D'après l'analyse du cas précédent, la somme de la mémoire de routage sur $2p' = \Theta(\log_\delta n')$ sommets est $\Omega(n' \log n')$ bits. Comme $n' = \Theta(n)$ and $\log \delta = \Theta(\log n) = \Theta(\log d)$, la somme de la mémoire de routage sur $\Theta(\log_d n)$ sommets est $\Omega(n \log n)$ bits.

Par conséquent, pour tout degré d tel que $3 \leqslant d \leqslant \varepsilon n$, nous obtenons que la somme de la mémoire de routage sur $\Theta(\log_d n)$ sommets du graphe $G_{n,d}$ est $\Omega(n \log n)$ bits. Soit $p'' = \Theta(\log_d n)$ le nombre de sommets contraints de $G_{n,d}$. La mémoire de routage est donc de $\Omega(n \log n)/p'' = \Omega(n \log d)$ bits sur au moins un sommet contraint parmi p''. Puisque $O(n \log d)$ bits sont suffisants pour router selon les plus courts chemins, on obtient que $\Theta(p'') = \Theta(\log_d n)$ sommets nécessitent $\Theta(n \log d)$ bits chacun. $\square$

Remarques. Le diamètre de $G_{n,d}$ est $4p - 3 = \Theta(\log_d n)$. Il est facile de montrer par récurrence que son nombre d'arêtes est plus petit que $3n$ et que son *épaisseur* est exactement 2. Rappelons que l'épaisseur d'un graphe G est le nombre minimum de graphes planaires dont l'union est G (voir Harary [44, pages 120-121]). D'une certaine façon, l'épaisseur d'un graphe est une mesure de sa « planarité », ce qui est très important pour la construction *VLSI* (voir aussi [35]). Clairement $G_{n,d}$ ne peut être planaire à cause du couplage parfait entre les deux copies des graphes $H_{p,\delta,k}$. Il est également facile de remarquer que $G_{n,3}$ peut être transformé en un graphe 3-régulier en ajoutant aux sommets de degré 2 (les sommets contraints) une structure identique à celle de la figure 5.3 (au centre) page 69. Le nombre de sommets et d'arêtes ajoutés au graphe est au plus $O(\log n)$.

6.5 Mémoire globale de routage

Dans la littérature concernant le routage compact, les bornes inférieures et supérieures sont souvent données pour la mémoire globale de routage (voir par exemple [27, 67]) alors que la mémoire locale de routage semble plus importante en pratique pour la conception réelle des routeurs. En général nous nous sommes donc intéressé à la mémoire locale de routage plutôt qu'à la mémoire globale de routage car également la valeur globale ne donne pas toujours la façon de distribuer l'information de routage sur chaque routeur. La mémoire globale de routage peut cependant donner une information théorique sur la difficulté du routage pour un réseau donné.

Le résultat suivant montre que, d'une manière générale, nous pouvons obtenir une borne inférieure de la mémoire globale de routage à partir d'une transformation des graphes de contraintes.

Lemme 6.6 *Soit $\mathcal{M}$ un ensemble de matrices $p{\times}q$ entières. Soit $\mathcal{G}$ un ensemble de graphes de contraintes de $\mathcal{M}$ tel que tout graphe $G \in \mathcal{G}$ est d'ordre au plus n, de degré maximum $d \geqslant 3$ et vérifie que tout sommet contraint x de G satisfait $2 \leqslant \deg(x) \leqslant n/p$. Alors, pour tout entier $n' \geqslant 9n$, il existe un graphe G' d'ordre n' et de degré maximum d qui possède une mémoire globale de routage de $\Omega(nm/p)$ bits, où $m = \log_2\left(|\mathcal{M}_{/\sim}|/\binom{n'}{q}\right) - O(\log n')$.*

Ce lemme a une importante application. Supposons qu'un graphe de contraintes G d'ordre n et de degré maximum d soit tel que ses p sommets contraints ont un degré $\leqslant n/p$ avec une mémoire de routage d'au moins $f(n)$ bits chacun. Alors on peut construire un nouveau graphe G' d'ordre $9n$, de degré maximum identique à G et ayant une mémoire globale de routage de $\Omega(nf(n))$ bits.

Preuve. Soit $G \in \mathcal{G}$ un graphe de contraintes d'une matrice $p \times q$ $M \in \mathcal{M}$ tel que G soit d'ordre n et de degré maximum $d \geqslant 3$, et tel que la somme de la mémoire de routage de ses p sommets contraints soit d'au moins $\log_2\left(|\mathcal{M}_{/\sim}|/\binom{n}{q}\right) - O(\log n)$ bits. Un tel graphe existe d'après le lemme 6.1. Soit δ, $2 \leqslant \delta \leqslant n/p$, le degré maximum parmi les p sommets contraints de G, et soit x un sommet contraint. Montrons que, pour tout entier $k \geqslant 1$, nous pouvons dupliquer x en k sommets qui ont les mêmes contraintes que x pour les sommets cibles de G, en ajoutant au plus $6k + 3\delta/2 - 1$ sommets intermédiaires et en conservant de degré maximum de G.

Soit $\alpha = \lceil k/(\delta - 1)\rceil \geqslant 1$ et soit B_x l'ensemble des voisins de x. Nous modifions le graphe G comme suit (voir l'exemple de la figure 6.5) : nous supprimons x (ainsi que toutes ses arêtes incidentes) et, nous connectons la racine d'un arbre à chaque sommets de B_x. Chaque arbre est $(\delta{-}1)$-aire (ou bien un arbre binaire si $\delta = 2$) de α feuilles qui sont toutes à la même distance de la racine. Nous connectons également un ensemble A_x de k sommets supplémentaires aux feuilles des arbres tel que chaque sommet de A_x soit connecté à exactement $\deg(x)$ différentes feuilles (une de chaque arbre) et tel que chaque feuille soit connectée à au plus $\delta - 1$ sommets de A_x. C'est toujours possible car les k sommets de A_x sont tous connectés à exactement une feuille parmi α feuilles d'un l'arbre. Par conséquent le nombre maximum d'arêtes connectant une feuille aux sommets de A_x est $\lceil k/\alpha \rceil = \lceil k/\lceil k/(\delta-1)\rceil\rceil \leqslant \lceil k/(k/(\delta-1))\rceil$, ce qui est plus petit que $\delta - 1$. Donc les feuilles sont de degré au plus $\delta \leqslant d$ (si $\delta = 2$, le degré de ces feuilles peut être $3 \leqslant d$ par hypothèse). Le degré des autres sommets est clairement $\leqslant d$. D'autre part, il existe un unique plus court chemin entre tout sommet de A_x et tout sommet de B_x. Le nombre de sommets de chaque arbre est au plus $5\alpha/2 - 1$ (car $\alpha \geqslant 1$ et ce nombre est plus petit que $2 \cdot 2^{\lfloor \log_2 \alpha \rfloor} - 1 + \lfloor \log_2 \alpha \rfloor + 1$). Par conséquent le nombre de sommets ajoutés pour la duplication de x est au plus $-1+\delta(5\alpha/2-1)+k \leqslant 5\delta(k/(\delta-1)+1)/2-\delta+k-1 \leqslant 6k+3\delta/2-1$.

Nous appliquons maintenant la duplication aux p sommets contraints de G avec $k = \lceil n/(6p) \rceil \geqslant 1$. Pour obtenir un graphe d'ordre exactement n' nous ajoutons un chemin que nous connectons à un sommet de degré 2. Si le graphe obtenu ne contient pas de sommet de degré 2, nous pouvons choisir un bloc de sommets dupliqués à partir d'un sommet contraint x et nous supprimons les arêtes connectant B_x à la racine des arbres en les remplaçant pas des chemins de longueur 2, créant ainsi $\deg(x) \leqslant \delta$ sommets de degré 2. À cette étape de la construction l'ordre du graphe obtenu est $m \leqslant n+p(6k+3\delta/2-1)+\delta \leqslant n+p(6(n/(6p)+1)+3\delta/2-1)+\delta \leqslant 2n + 3n/2 + 5p + n/p \leqslant 7n/2 + 5n \leqslant 9n \leqslant n'$ ($5p + n/p$ est maximum pour $p = n$). Puisqu'il existe un sommet de degré 2, nous pouvons ajouter un chemin de longueur $n' - m \geqslant 0$ pour obtenir un graphe G' d'ordre n et de degré maximum d.

Par construction, il y a un unique plus court chemin entre tout sommet de A_x et tout sommet de B_x. Par conséquent tout sommet de A_x a les mêmes contraintes que le sommet contraint original x de G. Nous pouvons choisir $k = |A_x| = \Theta(n/p)$ ensembles disjoints de sommets $C_1, \ldots, C_k$ de p sommets chacun tel que chacun des sommets de A_x appartient à un ensemble différent C_i, pour tout sommet contraint x de G et tout $i \in \{1, \ldots, k\}$. Tout ensemble C_i, $i \in \{1, \ldots, k\}$, est un ensemble de p sommets contraints de la matrice M. Par application du lemme 6.1, la somme de la mémoire de routage des p sommets de C_i, $i \in \{1, \ldots, k\}$, est donc au moins $m = \log_2 (|\mathcal{M}_{/\sim}| / \binom{n'}{q}) - O(\log n')$ bits. Puisque les ensembles C_i sont indépendants, k ensembles de sommets calculent indépendamment au moins m bits, c'est-à-dire un total d'au moins $km = \Omega(nm/p)$ bits. □

La duplication des sommets contraints d'un graphe G d'ordre n en k nouveaux sommets contraints fournit un nouveau graphe, noté Split-G, d'ordre $\Theta(n)$ et de même degré maximum mais avec $\Theta(kn)$ arêtes supplémentaires. Ceci peut accroître l'épaisseur du nouveau graphe. De plus la distance entre les sommets contraints et les voisins dans G du sommet contraint original peut croître jusqu'à $\Theta(\log_\delta k)$, où δ est le degré maximum des sommets contraints.

Le résultat suivant montre qu'il existe un graphe possédant une mémoire globale de routage de $\Theta(n^2 \log d)$ bits.

Théorème 6.2 *Soit ε, $0 < \varepsilon < 1$, une constante fixée. Pour tout entier d, $3 \leqslant d \leqslant \varepsilon n$, il existe un graphe d'ordre n et de maximum degré d qui requiert une mémoire de routage de $\Theta(n \log d)$ bits sur $\Theta(n)$ sommets.*

Preuve. Le théorème 6.1 montre que le graphe $G_{\lfloor n/9 \rfloor, d}$ d'ordre $\lfloor n/9 \rfloor$ et de degré maximum d, pour tout $3 \leqslant d \leqslant \varepsilon n$, ε étant un constante fixée < 1, a une mémoire de routage de $\Theta(n \log d)$ bits sur $\Theta(\log_d n)$ sommets contraints. Cette mémoire de routage est due à un grand ensemble de matrices $p \times q$ entières $\mathcal{M}$ tel que $\log_2 |\mathcal{M}_{/\sim}| = \Omega(n \log n)$. Le degré maximum des sommets contraints est $\delta = O(\sqrt{n})$ (voir la construction dans la preuve du théorème 6.1) et, de plus, $n/p = \Theta(n/\log_d n) = \Omega(n/\log n)$. Pour un entier n suffisamment grand,

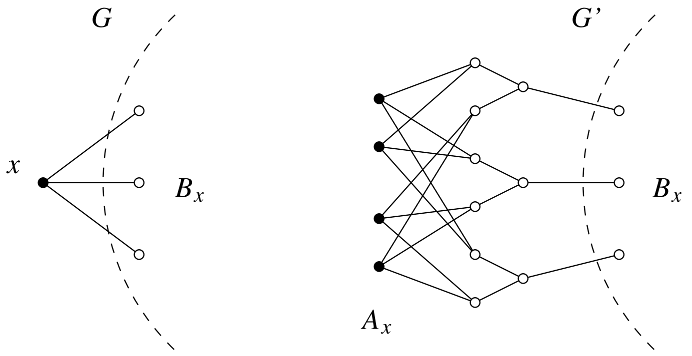


FIG. 6.5 – *Un sommet contraint de G dupliqué en 4 sommets contraints dans G'.*

$O(\sqrt{n}) \leqslant \Theta(n/\log n)$, *i.e.* $\delta \leqslant n/p$. Nous pouvons appliquer par conséquent le lemme 6.6 et construire ainsi un graphe, noté Split-$G_{n,d}$, d'ordre n, de degré maximum d et possédant une mémoire de routage de $\Omega(n \log d)$ bits par sommet sur $\Theta(n)$ sommets. □

Remarque. Dans la construction du graphe Split-$G_{n,d}$, nous utilisons la duplication de sommets contraints de $G_{\lfloor n/9 \rfloor, d}$ avec $k = \Theta(n/p) = \Theta((n/\log n)\log d)$. Par conséquent le diamètre de Split-$G_{n,d}$ est $\Theta(\log_d n)$ et est donc le même que le diamètre de $G_{n,d}$ (voir les remarques à la suite du lemme 6.6 et du théorème 6.1). Cependant son épaisseur[2] devient $\Omega(k)$ et peut être donc de l'ordre de n pour certaines valeurs de d.

6.6 Routage presque de plus courts chemins

Plusieurs auteurs ont suggéré de relâcher la contrainte des plus courts chemins. En pratique on peut considérer les méthode de routage *presque* de plus courts chemins, c'est-à-dire utilisant un facteur d'élongation petit. En effet les méthode de routage de facteur d'élongation en $O(n)$ et même en $O(\log n)$ peuvent être décrites d'une manière très compacte (voir le résumé page 89) mais cela introduit des temps plus long de communication. La borne inférieure de $\Omega(n^2)$ bits de la mémoire globale de routage a été prouvée pour tout facteur d'élongation < 2 dans le chapitre précédent (voir aussi [27]). Dans [29] nous avions déjà montré une borne inférieure exacte sur la mémoire locale de routage en $\Theta(n \log n)$ bits pour tout facteur d'élongation < 2, mais seulement requise par $\Omega(n^{\varepsilon})$ sommets d'une construction de degré maximum $\Theta(n)$ (ε constante telle que $\varepsilon < 1$). En utilisant le théorème 6.1, le graphe $G_{n,\lfloor \sqrt{n/10} \rfloor}$ permet de donner une borne inférieure de la mémoire locale de routage en $\Theta(n \log n)$ bits sur seulement un nombre constant de routeurs, mais pour tout facteur d'élongation $< 5/3$, et avec une construction de degré maximum au plus $O(\sqrt{n})$. De plus il est simple de voir que le graphe Split-$G_{n,\lfloor \sqrt{n/10} \rfloor}$ possède une mémoire globale de routage de $\Theta(n^2 \log n)$ bits avec un degré maximum en $O(\sqrt{n})$, mais avec un facteur d'élongation $< 7/5$ (dû à la transformation du lemme 6.6 qui augmente la distance entre les sommets contraints et les sommets cibles de 2).

2. L'épaisseur de tout graphe non orienté $G = (V, E)$ est au moins $\lceil |E|/(3|V| - 6) \rceil$ puisque tout graphe planaire d'ordre n a au plus $3n - 6$ arêtes.

Nous montrons dans le théorème suivant qu'une borne inférieure de la mémoire de routage des réseaux de degré maximum borné peut être donnée pour tout les facteurs d'élongation < 2.

Théorème 6.3 *Pour toute constante entière $d \geqslant 3$, il existe un graphe d'ordre n et de degré maximum d qui possède une mémoire globale de routage d'au moins $\Omega(n^2/\log^2 n)$ bits, pour tout facteur d'élongation < 2.*

Ce résultat implique donc une borne inférieure de la mémoire locale de routage en $\Omega(n/\log^2 n)$ bits pour les réseaux de degré maximum borné et pour tout facteur d'élongation < 2.

Preuve. Nous allons construire un réseaux particulier d'ordre n et de degré maximum 3. En effet, sans perte de généralité nous pouvons toujours connecter un graphe $K_{1,d}$ à un sommet non contraint (voir la preuve du théorème 6.1) pour obtenir une construction de degré maximum $d = O(1)$. Soit N un entier suffisamment grand et p un entier en $\Theta(\log N)$. Nous allons construire une nouvelle classe de graphes de contraintes de $\mathcal{A}_{p,2}$, d'ordre N et de degré maximum 3, et prouver que l'un de ces graphes à une mémoire globale de routage de $\Omega(n^2/\log^2 n)$ bits. On considère 2 copies du graphe Split-$G_{N,3}$ construit dans la preuve du théorème 6.2 dont on a supprimé les arêtes définissant le couplage parfait. On appelle ces sommets, maintenant déconnectés, sommets *frontaux*. Le nombre de ces sommets est 2^p. On note 2α le nombre de sommets contraints du graphe Split-$G_{N,3}$. Les 4 graphes connexes résultants (A, B, A' et B') sont connectés comme suit (voir aussi la figure 6.6) : les α sommets contraints, de degré 2, des graphes A et A' sont connectés deux à deux par un chemin de longueur $2t$. Le sommet central est noté x_i, $i \in \{1, \dots, \alpha\}$. De même les graphes B et B' sont connectés par un chemin de même longueur et dont le sommet central est noté x'_i, $i \in \{1, \dots, \alpha\}$. Soit $M \in \mathcal{A}_{p,2}$. Les sommets frontaux des graphes connectés $A - A'$ et $B - B'$ sont maintenant reliés suivant un couplage parfait en accord avec M (M peut être vue comme équivalente sous $\sim$ à une matrice $\begin{bmatrix} M' \\ \pi(M') \end{bmatrix}$, π étant une permutation des lignes de M', *i.e.* une bijection de $\{1, \dots, 2^p\}$ dans $\{1, \dots, 2^p\}$). Enfin, chacune des arêtes du couplage parfait est subdivisée en $2t$ arêtes, et le sommet central est noté y_j, $j \in \{1, \dots, 2^p\}$. Les sommets x_i et x'_i forment l'ensemble X des sommets contraints et les sommets y_j forment l'ensemble Y des sommets cibles. On appelle G le graphe final et n sont ordre.

M est une matrice de contraintes de G, sans être cependant une matrice de contraintes de facteur d'élongation < 2. En effet les chemins de routage entre un sommet $x \in X$ et n'importe quel sommet $y \in Y$ n'étant pas nécessairement simples, il est possible que certaines fonctions de routage réalisent un facteur d'élongation au plus $(d_G(x, y) + 2)/d_G(x, y) = O(1/\log N)$ sans vérifier les contraintes. Soit R une fonction de routage sur G de facteur d'élongation $s(R, G) < 2$. Sans perte de généralité supposons que $x \in X$ soit un sommet connectant A et A'. On pose u_i, $i \in \{1, \dots, t\}$, (respectively u'_i) les sommets rencontrés sur le chemins allant de x jusqu'au premier sommet du graphe A (respectively A'). Tous les sommets frontaux de A sont à une même distance de u_t, distance qui vaut $D = \Theta(\log N)$.

Nous allons montrer que les routeurs R_x, R_{u_i}, et $R_{u'_i}$, $x \in X, i \in \{1, \dots, t-1\}$, « calculent » la matrice $C_\mu(M)$, qui est la matrice canonique de M. La preuve est similaire à la preuve celle du lemme 6.1.

Sans perte de généralité, on suppose que le port de l'arc (x, u_1) est numéroté 1 et le port de l'arc (x, u'_1) est numéroté 2. Pour uniformiser les notations on pose $u_0 = u'_0 = x$. On note p_i (respectively p'_i), $i \in \{0, \dots, t-1\}$, le port de l'arc (u_i, u_{i+1}) (respectively

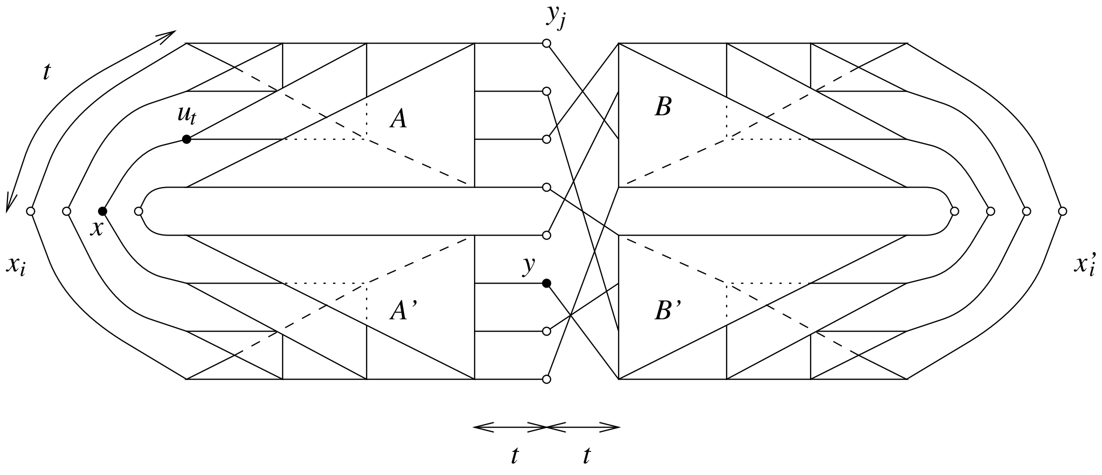


FIG. 6.6 – *Graphe de contraintes de degré maximum* 3 *pour le théorème 6.3.*

(u'_i, u'_{i+1})). Soit $y \in Y$. x est à distance $d_G(x,y) = 2t + D$ de y. Supposons que R_x route un message à destination de y dans la mauvaise direction, par exemple se dirigeant vers le graphe A alors que tous les plus courts chemins allant de x à y passent par u'_1. Fixons $2t \geqslant d_G(x,y) = \Theta(\log N)$. Montrons que si $s(R,G) < 2$ alors le message ne peut pas passer par u_t. En effet, si le message circule jusqu'en u_t puis revient en x pour router ensuite vers y, alors $d_R(x,y) \geqslant 2t + d_G(x,y) \geqslant 2d_G(x,y)$, ce qui implique un facteur d'élongation d'au moins 2. D'autre part si le message arrive jusqu'à u_t, traverse le graphe A jusqu'en un sommet y' (qui ne peut pas être y), puis arrive dans le graphe B ou B' pour ensuite se diriger vers y, alors $d_R(x,y) \geqslant d_G(x,y) + 2t + 2$, impliquant aussi $s(R,G) \geqslant 2$. Par conséquent le message doit passer nécessairement par u'_t.

Supposons qu'un observateur connaisse la liste des numéros des sommets cibles Y, la fonction C_μ, les fonctions R_x, R_{u_i}, $R_{u'_i}$ et les ports p_i et p'_i donnés dans l'ordre. Une procédure simple permet alors de trouver une matrice équivalente à M sous $\sim$ et donc de calculer $C_\mu(M)$. En effet, l'observateur à suffisamment d'informations pour simuler le parcourt d'un message d'un sommet contraint vers tout sommet cible, puisque le message traverse de manière déterminée (par la matrice) soit le sommet u_t soit le sommet u'_t.

Nous avons vu que certaines matrices M nécessitaient $\Omega(N \log N)$ bits pour être décrites. D'autre part, nous avons vu également que la liste des sommets cibles de Y peut être décrite par une fonction caractéristique f_Y nécessitant $\log_2 \binom{n}{2^p} + O(\log n) = O(n)$ bits. Il suffit d'un bit pour spécifier chaque p_i et p'_i. Nous codons avec $\lceil \log_2 \mathrm{MEM}(G,R,x) \rceil$ bits la longueur de la définition de la fonction de routage R_x, information supplémentaire suffisante pour pouvoir ordonner la liste des fonctions R_x, R_{u_i} et $R_{u'_i}$ données à l'observateur. Pour tout sommet z, on note $M(z) = \mathrm{MEM}(G,R,z) + \lceil \log_2 \mathrm{MEM}(G,R,z) \rceil$, le nombre de bits nécessaire pour coder R_z et la longueur du codage de R_z. D'après le théorème 6.2, $\Theta(N/\log N)$ blocs de sommets contraints calculent chacun de manière indépendante $\Omega(N \log N)$ bits, soit un total de $\Omega(N^2)$ bits. Par conséquent,

$$\sum_{x \in X} \left(M(x) + \sum_{i=1}^{t-1} (M(u_i) + M(u'_i) + 2) \right) + O(n) = \Omega(N^2)$$

D'après la construction de G, $n = O(N \log N)$, *i.e.* $N = \Omega(n/\log n)$. Soit I la somme de la mémoire de routage de tous les sommets contraints x et des sommets u_i et u'_i. L'équation précédente se réécrit donc en $I + O(\log I) = \Omega(N^2) = \Omega((n/\log n)^2)$, ce qui implique $I = \Omega(n^2/\log^2 n)$, et qui termine la preuve. □

6.7 Borne inférieure exacte de la compacité maximale

Nous avons déjà étudié la compacité maximale dans le chapitre 4. Nous avons montré alors que $\mathrm{IRS}(n) \geqslant n/12$. Nous avons également montré que, lorsque le degré maximum est borné par une constante d, alors $\mathrm{IRS}(n,d) = \Omega(n/\log^2 n)$. Cette même borne inférieure, a été également établie de manière indépendante dans [9], un document non encore publié. Récemment, Kranakis et Krizanc ont amélioré cette borne [57] en montrant $\mathrm{IRS}(n,d) = \Omega(n/\log n)$, pour tout d constant, $d \geqslant 3$. De plus, ils ont montré que ce nombre d'intervalles peut être requis par $\Omega(n)$ arcs. Nous allons montrer que l'on peut encore améliorer cette borne pour la rendre exacte.

Le théorème 6.1 indique que certains graphes d'ordre n et de degré maximum borné peuvent requérir une mémoire de routage de $\Theta(n)$ bits par routeurs. Par conséquent nous obtenons une nouvelle borne inférieure de la compacité maximale des graphes de degré maximum borné en remarquant que : puisqu'un intervalle de numéros de destinations ne nécessite qu'au plus $O(\log n)$ bits pour être codé il existe donc au moins un arc qui nécessite $\Omega(n/\log n)$ intervalles de destinations (le nombre d'arcs incidents à un sommet étant borné). Le théorème suivant montre que cette borne inférieure n'est pas exacte en général.

Théorème 6.4 *Soit G un graphe d'ordre n possédant au moins p sommets de degré au plus d et nécessitant chacun une mémoire de routage d'au moins m bits. Alors la compacité de G est au moins k intervalles avec*

$$\frac{m}{d} < 2k \log_2 \left(\frac{n}{2k} + 1 \right) + 2k$$

De plus, on trouve ce nombre d'intervalles sur au moins p arcs.

Preuve. Soit x un sommet de degré au plus d dans un graphe d'ordre n. Supposons que la mémoire de routage de x soit d'au moins m bits. Supposons que x utilise la méthode du routage par intervalles. Soit $I = [a_1, b_1] \cup \ldots \cup [a_k, b_k]$ l'ensemble des numéros de destinations de $\{1, \ldots, n\}$ associés à un arc incident à x. I est codé par au plus k intervalles. Soit $f(n,k)$ le nombre minimum de bits pour coder l'ensemble I.

Nous pouvons imaginer en effet que x utilise un codage qui tient au mieux compte de l'information contenu dans I. Nous allons décrire un exemple de procédé de codage de I qui diffère de celui des intervalles. Nous pouvons représenter l'ensemble I par une liste $(d_1, \ldots, d_{2k})$ de $2k$ entiers tels que $d_1 = a_1$, $d_{2i} = b_{2i} - a_{2i}$, $i \in \{1, \ldots, k\}$, et $d_{2j+1} = a_{2j+1} - b_{2j}$, $j \in \{1, \ldots, k-1\}$. En d'autres termes, I peut être codé par une valeur initiale $d_1 = a_1$ et une liste de $2k - 1$ entiers représentant des longueurs. La liste $(d_1, \ldots, d_{2k})$ peut être donc codée par au plus $\sum_{i=1}^{2k} \lceil \log_2 d_i \rceil < \sum_{i=1}^{2k} (\log_2(d_i) + 1)$ bits. Donc $f(n,k) < \log_2(\Pi_{i=1}^{2k} d_i) + 2k$. Puisque $\sum_{i=1}^{2k} d_i = n$, alors $\Pi_{i=1}^{2k} d_i < \lceil n/(2k) \rceil^{2k}$. Donc $f(n,k) < 2k \log_2(n/(2k) + 1) + 2k$. Par définition de la mémoire de routage, $m/d \leqslant f(n,k)$ car au moins un arc incident à x requiert plus de m/d bits (sinon la mémoire de routage de x serait inférieure à m) et moins de $f(n,k)$ bits en utilisant l'ensemble I en k intervalles. Par conséquent p sommets de G ont au moins un arc incident nécessitant au moins k intervalles tel que $m/d \leqslant f(n,k)$, ce qui termine la preuve. □

Remarque. Dans la preuve précédente, nous utilisons principalement le fait que I, composé de k intervalles, peut être codé par approximativement $\log_2 \binom{n}{2k}$ bits, ce qui est beaucoup plus petit que $k \log_2 n$ quand k est grand (en utilisant l'inégalité $\binom{x}{y} < (\frac{xe}{y})^y$ de [62, lemme 1.6]).

Corollaire 6.1 *Pour toute constante entière $d \geqslant 3$, il existe un graphe d'ordre n et de degré maximum d et de compacité $\Theta(n)$ par arc sur $\Theta(n)$ arcs.*

Nous aurions pu aussi déduire un autre corollaire un peu plus général mais ne donnant pas une formule explicite de la compacité maximale : pour toute constante fixée ε, $0 < \varepsilon < 1$, et pour tout d, $3 \leqslant d \leqslant \varepsilon n$, il existe un graphe d'ordre n et de degré maximum d (le graphe Split-$G_{n,d}$) possédant une compacité au moins k sur $\Omega(n)$ arcs avec $k \log(n/k) = \Omega(n(\log d)/d)$.

Preuve. Nous appliquons le théorème 6.4 avec d constant. Le graphe Split-$G_{n,d}$ possède une mémoire de routage de $\Theta(n)$ bits par sommets sur $\Theta(n)$ sommets. L'inégalité du théorème 6.4 établit dans ce cas que $k \log(n/k) = \Omega(n)$, ce qui implique $k = \Omega(n)$ intervalles par arc sur $\Omega(n)$ arcs (pour s'en convaincre prendre $k = n/f(n)$ avec f une fonction quelconque de n, et montrer que nécessairement f est bornée par une constante). Rappelons que la compacité de tout graphe est au plus $n/2$ sur tous ses arcs, et que le nombre d'arcs est au plus $O(n)$ pour les graphes (symétriques) de degré borné. □

Proposition 6.2 *Il existe un graphe 3-régulier d'ordre $\Theta(n)$ de compacité $\Theta(n)$ par arc sur $\Theta(n)$ arcs.*

En effet, il suffit de modifier chacun des sommets de degré 2 du graphe Split-$G_{n,3}$ par une structure adéquate (voir aussi la remarque page 81) pour obtenir un graphe régulier sans changer les contraintes.

6.8 Résumé

La table suivante résume les différentes bornes inférieures et supérieures de la mémoire de routage pour les fonctions de routage presque de plus courts chemins. Dans cette table, ε est une constante réelle quelconque telle que $0 < \varepsilon < 1$.

Remarques. Nous avons supposé jusqu'à présent des coûts uniformes sur les arcs des réseaux. Toutes les bornes inférieures de la mémoire de routage sont évidemment encore valables pour des coûts non uniformes. Cependant en ce qui concerne les bornes supérieures, les méthodes de routage proposées doivent être adaptées pour toute valeur de coûts non négatif sur les arcs. Cette option peut être importante pour des applications pratiques afin de simuler des différences entre les liens d'un même réseau : différence de latence ou différence de bande

élongation	degré	mémoire locale	mémoire globale
$s = 1$	$3 \leqslant d \leqslant \varepsilon n$	$\Theta(n \log d)$ [théorème 6.1]	$\Theta(n^2 \log d)$ [théorème 6.2]
$1 \leqslant s < 1.4$	$3 \leqslant d \leqslant \sqrt{n/10}$	$\Theta(n \log n)$ [section 6.6]	$\Theta(n^2 \log n)$ [théorème 6.2]
$1 \leqslant s < 2$	$3 \leqslant d < n$	$\Theta(n \log n)$ [29]	$\Omega(n^2)$ [théorème 5.1] $O(n^2 \log d)$
$1 \leqslant s < 2$	$3 \leqslant d = O(1)$	$\Omega(n/\log^2 n)$ [théorème 6.3] $O(n)$	$\Omega(n^2/\log^2 n)$ [théorème 6.3] $O(n^2)$

TAB. 6.1 –

passante par exemple. La méthode de routage des tables de routage permet de réaliser une telle adaptation au poids des arcs.

Pour les facteurs d'élongation plus importants, la table ci-après présente l'état de l'art des meilleures méthodes de routage compact connues en fonction du facteur d'élongation. Notez que peu d'études ont été menées sur la mémoire de routage en considérant à la fois le degré maximum des réseaux et un facteur d'élongation plus grand que 2.

Dans les articles originaux les méthodes de routage résumées dans cette table sont données pour une mémoire de routage dépendante d'un terme en $n^{1/k}$, où $k \geqslant 1$ est une fonction entière croissante du facteur d'élongation. La table donne une présentation concise des résultats de façon à ce qu'ils puissent être comparés en fonction de la valeur du facteur d'élongation.

élongation	mémoire locale	mémoire globale
$2 \leqslant s < 3$	$\Omega(n^{1/(2s+4)})^\dagger$ [67] $O(n \log n)$	$\Omega(n^{1+1/(2s+4)})$ [67] $O(n^2 \log n)$
$3 \leqslant s < 16$	$\Omega(n^{1/(2s+4)})^\dagger$ [67] $O(n \log n)$	$\Omega(n^{1+1/(2s+4)})$ [67] $O(n^{1+1/\lfloor \log_2(s+1) \rfloor} \log^2 n)$ [3]
$16 \leqslant s < 87$	$\Omega(n^{1/(2s+4)})^\dagger$ [67] $O(n^{1/\lfloor \sqrt{s}/4 \rfloor} \log^2 n)$ [4]	$\Omega(n^{1+1/(2s+4)})$ [67] $O(n^{1+1/\lfloor \log_2(s+1) \rfloor} \log^2 n)$ [3]
$s \geqslant 87$	$\Omega(n^{1/(2s+4)})^\dagger$ [67] $O(\sqrt{s} n^{1/\lfloor \sqrt{s}/4 \rfloor} \log^2 n)$ [4]	$\Omega(n^{1+1/(2s+4)})$ [67] $O(s^3 n^{1+1/\lfloor (s-3)/12 \rfloor} \log n)$ [67]
$s = O(\log n)$	$\Omega(1)^\dagger$ $O(e^{\sqrt{\log n}} \log^{3/2} n)$ [4]	$\Omega(n)$ [67] $O(n \log^4 n)$ [67]
$s = O(\sqrt{n})$	$\Omega(1)^\dagger$ $O(e^{\sqrt{\log n}} \log^{3/2} n)$ [4]	$\Omega(n)$ [67] $O(n \log n)$ [66]

†Borne déduite de la borne inférieure de la mémoire globale de routage.

TAB. 6.2 –

Commentaires.

– Dans [3], le facteur d'élongation est $2^k - 1$ et la méthode de routage permet des coûts non uniformes sur les arcs. Une autre méthode de routage est également présentée dans [3]. Elle nécessite localement au plus de $O(k(d+n^{1/k})\log n)$ bits mémoire sur un sommet de degré d, et ce pour tout facteur d'élongation $2{\cdot}3^k - 1$. Cette méthode de routage est donc intéressante dans le cas des réseaux de degré maximum borné.

- Dans [4], le facteur d'élongation est $16k^2$ et les coûts peuvent être non uniformes sur les arcs. La borne de la mémoire locale de routage est donnée en fonction du diamètre D du graphe : $O(kn^{1/k}\log n \log D)$ bits par sommet. Afin de pouvoir comparer les différentes valeurs de la tables, nous avons supposé $D = O(n)$. Les entêtes, tout comme les numéros des sommets, sont de taille $O(\log n)$ bits. Notez que $n^{1/\sqrt{\log n}} = e^{\sqrt{\log n}}$.
- Dans [67], le facteur d'élongation est $12k+3$, mais la méthode de routage ne supporte que des coûts uniformes sur les arcs et n'est pas équitablement répartie sur les sommets, ce qui ne permet pas de borner la mémoire locale de routage aussi finement que la mémoire globale. Les entêtes utilisés sont de taille seulement $O(\log n)$ bits. Cette méthode de routage utilise une numérotation des sommets avec $O(\log^2 n)$ bits, ce qui ne correspond pas au modèle de la mémoire de routage des bornes inférieures données dans le mêmes article avec une numérotation sur $O(\log n)$ bits. Cependant, pour de grande valeur du facteur d'élongation, cette méthode de routage fournit une mémoire de routage asymptotiquement optimale.

En définitive, nous constatons qu'en ce qui concerne la mémoire locale de routage, il y a pas de méthode de routage universel connues plus compacte que les tables de routage, même pour un facteur d'élongation approchant 16.

CHAPITRE 7

Conclusion et perspectives

Cette thèse n'apporte pas directement de solutions au problème du routage compact car elle ne propose pas de nouvelles méthodes ni de nouvelles stratégies de routage économes en mémoire. Elle répond cependant précisément à plusieurs questions de fond, telle que la limite de complexité mémoire des routeurs universels, ou la détermination des paramètres fondamentaux permettant de limiter la taille de la mémoire de routage (le facteur d'élongation, le degré du réseau, la planarité, la taille des noms donnés aux nœuds, *etc*). Malgré cela, ce travaille laisse de nombreux problèmes ouverts dont nous donnons quelques exemples dans la section 7.2. Il ouvre également de larges perspectives de recherches dont nous traçons les grandes lignes en section 7.3. La section 7.1 reprend les résultats importants des quatre chapitres principaux de cette thèse.

7.1 Conclusion

Théorème 3.2 $G \in 1\text{-Lirs} \Leftrightarrow G$ *n'est pas un lithium.*

Cela signifie que tout réseau qui n'est pas de la forme d'un lithium supporte une fonction de routage par intervalles avec un intervalle linéaire sur chaque arc. Inversement les réseaux en lithium ne peuvent pas supporter de telles fonctions de routage.

Théorème 4.1 *Pour tout entier* $n \geqslant 2$, $\mathrm{IRS}(n) \geqslant \dfrac{n}{12}$.

Cela signifie qu'il existe un réseau d'ordre n sur lequel toute fonction de routage par intervalles de plus courts chemins nécessite au moins $n/12$ intervalles sur au moins un arc. Nous avons également montré que ce nombre d'intervalles pouvait être nécessaire sur $\Theta(n)$ routeurs. Le principe de grouper les destinations utilisant le même lien ne permet pas, en général, de router selon des plus courts chemins d'une manière compacte, et ce quelle que soit la façon de choisir la numérotation des nœuds et quel que soit le choix des plus courts chemins à travers le réseau.

Corollaire 6.1 *Pour tout constante entière* $d \geqslant 3$, *il existe un graphe d'ordre* n *de degré maximum* d *et de compacité* $\Theta(n)$ *par arc, sur* $\Theta(n)$ *arcs.*

Cela signifie qu'il existe un réseau de n nœuds et de degré 3 pour lequel toute fonction de routage par intervalles de plus courts chemins nécessite au moins $\Theta(n)$ intervalles sur presque tous ses arcs. Nous avons ainsi montré que la compacité maximale (pire cas) du routage par intervalles ne dépendait pas du degré maximal du réseau, contrairement à la mémoire de routage pour les tables de routage.

Théorème 5.1 *Pour tout $s < 2$,* $\mathrm{MEM}_{globale}(n,s) = \Omega(n^2)$.

Cela signifie qu'il existe un réseau d'ordre n pour lequel le nombre total de bits nécessaires pour coder l'information de routage est au moins $\Omega(n^2)$ bits pour réaliser un routage où les chemins restent proches des plus courts à un facteur 2 près, et ceci quelque soit la manière de choisir les chemins de routage, quelque soit la manière de numéroter les nœuds et les ports des liens du réseau et quelque soit la manière de coder l'information de routage.

Théorème 6.2 *Soit ε, $0 < \varepsilon < 1$, une constante fixée. Pour tout entier d, $3 \leqslant d \leqslant \varepsilon n$, il existe un graphe d'ordre n et de degré maximum d qui requiert une mémoire de routage de $\Theta(n \log d)$ bits sur $\Theta(n)$ sommets.*

Cela signifie qu'il existe un réseau d'ordre n et de degré maximum d où il n'est pas possible de router selon des plus courts chemins avec moins de $\Theta(n \log d)$ bits d'information par routeur. Ce résultat reste valable quel que soit le choix des plus courts chemins sur ce réseau et la numérotation des nœuds et les ports des liens du réseau. Il n'est donc pas possible, sur un tel réseau, de définir des tables de routage compactes de plus courts chemins.

7.2 Problèmes ouverts

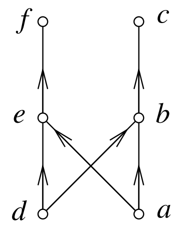
Tout au long de cette thèse nous avons précisé au fur et à mesure de nombreux problèmes particuliers non encore résolus. Dans cette section nous signalons d'autres problèmes ouverts qui nous semble de première importance dans l'étude du routage compact.

7.2.1 Routage par ordre partiel

Grouper les noms des destinations utilisant un même arc en intervalles linéaires revient à trouver un ordre *total* sur les sommets. On peut se demander ce qu'il adviendrait si l'on groupait les destinations suivant un *intervalle* d'un ordre *partiel* noté $\leqslant_p$: dans ce cas l'intervalle fermé $[a,b]$ représente tous les éléments x de l'ordre partiel tels que $a \leqslant_p x \leqslant_p b$. À l'instar des classes de graphes supportant une fonction de routage par intervalles avec k intervalles au plus par arc, on note k-Ors l'ensemble des graphes pour lesquels il existe un ordre partiel $\leqslant_p$ et tel qu'il existe au plus k intervalles de cet ordre sur chaque arc. La figure 7.1 représente un exemple d'ordre partiel (à gauche) et un graphe avec une fonction de routage par ordre partiel (à droite). Cette fonction est une fonction de routage de plus courts chemins.

Bien sûr, k-Lirs $\subset$ k-Ors, car l'ordre « linéaire » est un ordre partiel particulier. Nous avons montré dans [26] que la classe des graphes 1-Ors coïncide exactement avec celle des graphes 1-Lirs. En effet pour tout graphe lithium, il n'existe aucun ordre partiel permettant de router avec au plus un intervalle par arc. Notons k-ORS l'ensemble des graphes supportant une fonction de routage de plus courts chemins par ordre partiel avec au plus k intervalles par arc. L'exemple de la figure 7.1, montre que $C_6 \in$ 1-ORS. On peut montrer que $C_n \in$ 1-ORS si et seulement si $n \leqslant 6$.

Il serait intéressant de pousser plus loin cette étude et de montrer l'étendue relative des classes k-LIRS, k-IRS et k-ORS. En ce qui concerne l'espace mémoire requis par une fonction



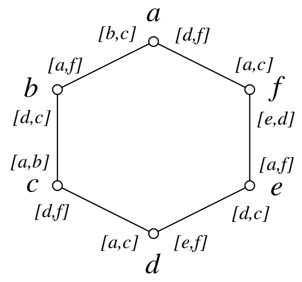


FIG. 7.1 – *Un graphe supportant une fonction de routage par ordre partiel de plus courts chemins mais pas dans* 1-LIRS.

de routage par ordre partiel, il est certain que la dimension de l'ordre (le nombre minimum d'extensions linéaires pour recouvrir l'ordre) influence son codage en mémoire. Dans ce cas, quels sont les graphes de (k, p)-ORS, c'est-à-dire les graphes supportant une fonction de routage par ordre partiel de dimension p et ayant au plus k intervalles par arc ?

7.2.2 Graphes planaires

Tout comme le degré maximum d'un réseau, la planarité est un facteur important en pratique. En effet, limiter de degré et restreindre la planarité permet de diminuer de manière significative la complexité de réalisation des réseaux réels (voir [35]). Dans le cas des réseaux planaires, Frederickson et Janardan montrent dans [31] des résultats sur le routage compact dans de tels graphes. Ils obtiennent des méthodes de routage utilisant $O(n^{4/3} \log n)$ bits de mémoire globale, mais ayant un facteur d'élongation 3. Ils proposent également une autre méthode de routage utilisant, pour toute constante ε, $0 < \varepsilon < 1/3$, une mémoire globale de routage de $O((1/\varepsilon)n^{1+\varepsilon} \log n)$ bits, mais ayant un facteur d'élongation 7 et utilisant une numérotation des sommets sur $O((1/\varepsilon) \log n)$ bits (et non pas sur $O(\log n)$ bits).

On peut donc se demander quelle est la mémoire de routage d'un graphe planaire utilisant des fonctions de routage de plus courts chemins. Étant donné les techniques développées dans cette thèse, la question pourrait être formulée comme suit : quel est le plus grand ensemble de matrices $\mathcal{M}$ tel qu'il existe une classe de graphes planaires $\mathcal{G}$, qui soit une classe de graphes de contraintes de $\mathcal{M}$? De cette réponse découle des limites théoriques sur le routage par intervalles de plus courts chemins ainsi que, plus généralement, des limites sur le routage compact dans les réseaux planaires.

En pratique, les arbres représentent une classe de réseaux importante parmi les graphes planaires. Nous avons vu au chapitre 5 une borne supérieure triviale de $O(n)$ bits sur la mémoire locale de routage pour les arbres d'ordre n. Bien sûr, la mémoire globale de routage est au plus $O(n \log n)$ bits en utilisant le routage par intervalles, mais on ne connaît pas de meilleures bornes de la mémoire locale de routage des arbres, et sa complexité reste un problème ouvert.

7.2.3 Étude en moyenne de la mémoire de routage

L'analyse du pire cas consiste à mettre en évidence ce qui pourrait arriver de pire lorsqu'un constructeur de réseaux dessine un nouveau réseau sans tenir compte de l'espace mémoire requis par le routage.

Contrairement à l'analyse du pire cas, l'analyse en moyenne consiste à étudier ce qui se passe dans la « majorité » des cas. Certes, des réseaux réalisant les pires cas pour toutes méthodes de routage ont été construits, mais rien ne dit que les réseaux se comportent aussi mal en moyenne. Les récents travaux de Burhman, Hoepman et Vitányi [10] sur les graphes aléatoires de Kolmogorov [63] semblent en effet montrer un meilleur comportement moyen.

Plus précisément, quelle est la mémoire de routage pour la classe des graphes aléatoires d-réguliers, spécialement pour les petits degrés en $o(\sqrt{n})$? Par exemple, il n'est pas clair qu'en moyenne un graphe de degré maximum 3 requiert une mémoire globale de routage de $\Theta(n^2)$ bits, bien que nous sachions maintenant que certains graphes de cette famille nécessitent $\Omega(n^2)$ bits.

Un effort considérable a été voué depuis le début des années 80 à l'étude des graphes aléatoires d-réguliers. Malgré cela et d'après Bollobás [7], l'étude de ces graphes est rendue difficile par le fait que, pour $d \geqslant 3$, il n'y a pas de formule explicite simple décrivant le nombre de graphes d-réguliers d'ordre n.

7.2.4 Compromis *espace-temps*

La mémoire de routage que nous avons défini comme la mémoire ROM est une borne inférieure de la *surface* d'un routeur. Cependant, un routeur à besoin de mémoire RAM pour le calcul local de la fonction de routage qu'il met en œuvre. Certains graphes ayant une mémoire de routage réduite peuvent nécessiter une mémoire RAM importante. C'est par exemple le cas des graphes de Cayley [15, 46, 61]. Pour tenir compte de ce phénomène, il serait bon de considérer un compromis entre la mémoire requise pour le stockage de la fonction de routage (ROM) et celle nécessaire à son calcul (RAM). Notons cependant que, pour les réseaux d'ordre n et de degré maximum d, la mémoire requise pour le calcul de la fonction de routage mise en œuvre par une table de routage est au plus $O(\log n)$ bits, ce qui est très inférieur à la mémoire locale de routage $O(n \log d)$. De la même manière un compromis pourrait être réalisé entre le temps de calcul et l'espace mémoire requis pour le calcul des fonctions de routage.

7.2.5 Complexité d'une méthode de routage donnée

Nous avons vu au chapitre 5 qu'une stratégie de routage est une fonction $RS : \mathcal{G} \to \mathcal{R}$ qui retourne, pour chaque graphe $G \in \mathcal{G}$, une fonction de routage $RS(G) \in \mathcal{R}$ sur G. Cette thèse s'est principalement attachée à l'étude de la complexité des fonctions de routage $RS(G)$, mais il est légitime de se demander comment mettre en œuvre la stratégie de routage RS elle-même. Par exemple, quel algorithme utiliser pour générer efficacement les intervalles de routage d'un graphe pour la méthode du routage par intervalles de plus courts chemins ? De plus est-il possible de réaliser cette génération de manière parallèle afin d'en diminuer le temps de calcul ? Également comment tenir à jour la fonction $RS(G)$ et son codage en cas d'évolution dynamique du réseau (tolérance aux pannes, ...).

7.2.6 Puissance de la numérotation des nœuds et des ports

Lors de l'étude du routage par intervalles faites aux chapitres 3 et 4, nous avons vu l'influence de la numérotation des nœuds sur l'efficacité du codage des fonctions de routage. Rappelons également qu'au chapitre 2, nous avons vu que, dans le cas du graphe complet, sa mémoire locale requise pour le routage variait de $O(n \log n)$ à $O(\log n)$ bits, jusqu'à même $O(1)$ bits, selon la numérotation des ports.

Les meilleures bornes supérieures proposées sur le problème du routage compact sont basées sur des méthodes de routage qui génèrent des chemins qui ne sont pas toujours des plus courts chemins. Ils sont seulement proches des plus courts à un certain facteur près (le facteur d'élongation). De plus ces méthodes de routage utilisent une numérotation des nœuds sur $O(\log^2 n)$ bits (voir [31, 67]). Or, notre borne inférieure exprimée dans le théorème 6.2 n'est valable que sous l'hypothèse d'une numérotation des nœuds sur $O(\log n)$ bits. Sous l'hypothèse d'une numérotation des nœuds en $O(\log^2 n)$ bits, quelle est la borne inférieure de la mémoire de routage pour un routage de plus courts chemins ? ou pour un routage de facteur d'élongation constant ?

7.2.7 Routage adaptatif

Nos bornes inférieures de la mémoire de routage peuvent aussi être appliquées au cas des fonctions de routage adaptatives. En effet, pour tout graphe G, toute fonction de routage adaptative sur G réalise nécessairement au moins une fonction de routage statique particulière sur G.

Le *routage booléen* est un modèle standard des méthodes de routage adaptatives, introduit par Flammini, Gambosi et Salomone dans [17]. Il peut être vu comme une extension du routage par intervalles. À chaque arc e est associé un prédicat P_e qui est vrai, en fonction d'une destination y, si et seulement s'il existe un plus court chemin allant en y en passant par e. Cette technique peut donc servir à coder localement tous les plus courts chemins pour des routeurs universels *adaptatifs*. Plusieurs études sur les bornes inférieures et supérieures d'un tel mode de routage ont été menées pour certaines classes de graphes : les anneaux avec cordes [18, 60], les graphes de Cayley [55]. Une borne inférieure optimale de $\Omega(n^2)$ bits par routeur pour les réseaux d'ordre n est donnée dans [59]. Récemment, Kranakis et Krizanc ont montré dans [56] une borne inférieure pour les réseaux de degré maximum d en $\Omega(nd \log d / \log n)$ bits par routeur et en $\Omega(n^2 d \log^2 d / \log^2 n)$ bits au total.

Cette dernière borne inférieure peut sans aucun doute être améliorée avec les constructions présentées dans cette thèse. L'étude généralisée pour des facteurs d'élongation supérieurs à 1 reste à faire, tout comme l'étude limitant à une constante k fixée le nombre de chemins à représenter.

7.3 Perspectives

7.3.1 Architectures parallèles

Le problème du routage compact est en grande partie résolu. Nous avons montré qu'il n'y a pas d'espoir de trouver une méthode de routage universel plus compacte que les tables de routage, même pour la classe des réseaux de degré borné. Nous avons vu que certaines méthodes, comme le routage par intervalles, pouvaient être pires que les tables.

Notre travail s'oriente donc vers une nouvelle approche du routage compact : l'étude du routage compact pour des classes de réseaux et des classes de fonctions de routage particulières. En pratique il est important, pour un concepteur de routeurs, de connaître la classes des architectures qu'il peut réaliser avec un type de routeur fixé. Même si l'on sait qu'on ne peut pas interconnecter toutes les architectures possibles avec un routeur basé, par exemple, sur le principe du routage par intervalles avec un intervalle par lien, il est crucial, en pratique, de caractériser les architectures que l'on pourra réaliser en utilisant cette méthode. En conséquence, les questions relatives au routage compact sont de la forme : quelles est la classe des réseaux ayant une mémoire de routage au plus $O(f)$ bits, avec par exemple f une fonction telle que $\log n$? Quelle est la mémoire de routage des graphes d'ordre n pour un routage de plus courts chemins sans inter-blocage ? *etc.*

7.3.2 Algorithmes distribués

Nous avons vu au début du chapitre 5 que le routage ne nécessite pas en général la description complète du réseau dans chaque routeur. Par exemple, les tables utilisent localement au plus $O(n \log n)$ bits alors qu'un graphe d'ordre n peut nécessiter $\Theta(n^2)$ bits pour être décrit (voir la section 5.2.2). Cependant, nous avons construit un réseau d'ordre n de degré maximum d pour lequel il n'existe pas de méthode de routage de plus courts chemins plus compacte (en ordre de grandeur) que les tables de routage. Notre construction est basée sur un couplage parfait aléatoire entre $\Theta(n)$ sommets du réseau. Ce réseau peut être complètement décrit avec au plus $O(n \log n)$ bits. D'autre part, dans le cas de degré d maximum non borné, nous prouvons par le théorème 6.2 que la mémoire de routage sur $\Theta(n)$ routeurs de notre réseau est de $\Theta(n \log n)$ bits par routeur. Cela signifie qu'il n'existe pas de méthode pour distribuer efficacement (en espace mémoire) une fonction de routage de plus courts chemins sur ce réseau car $\Theta(n)$ routeurs requièrent localement le stockage (en ordre de grandeur) de la description complète du réseau.

Cet exemple montre donc que le routage de plus courts chemins se distribue mal. Puisqu'une fonction de routage est un algorithme distribué particulier, introduisons un nouveau paramètre de complexité mémoire : le *coût mémoire de distribution*.

Soit A un algorithme résolvant un problème P. Notons $K(A)$ la complexité de Kolmogorov de A, c'est-à-dire la taille du plus petit programme qui retourne A et qui s'arrête. Soit $G = (V, E)$ un graphe d'ordre n et $\{A_x \mid x \in V\}$ la version distribuée de A sur le graphe G (voir [2] et [71] pour une discussion générale des algorithmes distribués). Le *coût mémoire global* (respectively *local*) *de distribution* est défini par :

$$\mathrm{CM}_{global}(A, G) = \frac{\sum_{x \in V} K(A_x)}{K(A)} \quad \text{et} \quad \mathrm{CM}_{local}(A, G) = n \cdot \frac{\max_{x \in V} K(A_x)}{K(A)}$$

qui représente le rapport de complexité mémoire entre la version séquentielle et la version distribuée de l'algorithme A pour G. $\mathrm{CM}_{global}(A,G)$, tout comme $\mathrm{CM}_{local}(A,G)$, est compris entre 1 et n, et tend vers 1 si A se distribue parfaitement sur G. Il tend au contraire vers n si A ne peut pas être distribué sur G. Au pire, il suffit de copier A dans chacun des nœuds de G et d'exécuter l'algorithme A de manière séquentielle. Bien sûr, en général, pour tout x, $K(A_x) > K(A)/n$ car chacun des algorithmes A_x doit prendre en compte certaines communications et redondances absentes de la version séquentielle. Le coût mémoire de distribution permet de quantifier ce sur-coût mémoire lié directement à la distribution de l'algorithme.

En particulier pour le problème du routage de plus court chemin noté S, nous avons donné un exemple de graphe G_0 d'ordre n pour lequel tout algorithme A qui résoud S est tel que $\mathrm{CM}_{global}(A,G_0) = \Theta(n)$ et $\mathrm{CM}_{local}(A,G_0) = \Theta(n)$.

L'étude générale de CM_{global} et CM_{local} avec des contraintes de communications pourrait permettre d'établir des classes de problèmes qui sont faciles ou difficiles à distribuer.

Bibliographie

[1] S. B. Akers, D. Harel, and B. Krishnamurthy, *The star graph: An attractive alternative to the n-cube*, in International Conference on Parallel Processing (ICPP), St. Charles, Illinois, Aug. 1987, pp. 393–400.

[2] B. Awerbuch, *Optimal distributed algorithms for minimum weight spanning tree, counting, leader election and related problems*, in 19^{th} Symposium on Theory of Computing (STOC), May 1987, pp. 230–240.

[3] B. Awerbuch, A. Bar-Noy, N. Linial, and D. Peleg, *Improved routing strategies with succinct tables*, Journal of Algorithms, 11 (1990), pp. 307–341.

[4] B. Awerbuch and D. Peleg, *Routing with polynomial communication-space trade-off*, SIAM Journal on Discrete Mathematics, 5 (1992), pp. 151–162.

[5] E. M. Bakker, J. van Leeuwen, and R. B. Tan, *Linear interval routing*, Algorithms Review, 2 (1991), pp. 45–61.

[6] A. E. Baratz and J. M. Jaffe, *Establishing virtual circuits in large computer networks*, Computer Networks, 12 (1986), pp. 27–37.

[7] B. Bollobás, *Lectures on random graphs*, in Progress in Graph Theory, A. J. Bondy and U. S. R. Murty, eds., Academic Press, 1984, pp. 3–41.

[8] K. S. Booth and G. S. Lueker, *Testing for the consecutive ones property, interval graphs and graphs planarity using pq-tree algorithms*, Journal of Computer and System Science, 13 (1976), pp. 335–379.

[9] V. Braume, *Theoretische und experimentelle Analyse von Intervall-Routing Algorithmen*, PhD thesis, Department of Mathematics and Computer Science, University of Paderborn, 1993.

[10] H. Buhrman, J.-H. Hoepman, and P. Vitányi, *Optimal routing tables*, in 15^{th} Annual ACM Symposium on Principles of Distributed Computing (PODC), ACM PRESS, May 1996, pp. 134–142.

[11] W. J. Dally and C. L. Seitz, *The torus routing chip*, Distributed Computing, 1 (1986), pp. 187–196.

[12] ———, *Deadlock-free message routing in multiprocessor interconnection networks*, IEEE Transaction on Computers, C-36 (1987), pp. 547–553.

[13] F. Desprez, E. Fleury, and M. Loi, *T9000 et C104: La nouvelle génération de transputers*, Research Report 93-01, LIP, École Normale Supérieure de Lyon, 69364 Lyon Cedex 07, France, Feb. 1993.

[14] E. W. Dijkstra, *A note on two problems in connexion with graphs*, Numerische Mathematik, 1 (1959), pp. 269–271.

[15] S. Even and O. Goldreich, *The minimum-length generator sequence problem is NP-hard*, Journal of Algorithms, 2 (1981), pp. 311–313.

[16] M. Flammini, *Private communication in New York*, June 1994.

[17] M. Flammini, G. Gambosi, and S. Salomone, *Boolean routing*, in 7^{th} International Workshop on Distributed Algorithms (WDAG), vol. 725 Lecture Notes in Computer Science, Springer-Verlag, 1993, pp. 219–233.

[18] ——, *Interval labeling schemes for chordal rings*, in 1^{st} International Colloquium on Structural Information & Communication Complexity (SIROCCO), P. Flocchini, B. Mans, and N. Santoro, eds., Carleton University Press, May 1994, pp. 111–124.

[19] ——, *Interval routing schemes*, in 12^{th} Annual Symposium on Theoretical Aspects of Computer Science (STACS), E. W. Mayr and C. Puech, eds., vol. 900 of Lecture Notes in Computer Science, Springer-Verlag, Mar. 1995, pp. 279–290.

[20] M. Flammini, J. van Leeuwen, and A. Marchetti-Spaccamela, *Lower bounds on interval routing*, Tech. Rep. 69, Università di L'Aquila, Dipartimento di matematica Pura ed Applicata, Oct. 1994.

[21] E. Fleury and M. Picquendar, *Massivelly parallel machine based on T9000 and C104*, Research Report 94-14, LIP, École Normale Supérieure de Lyon, 69364 Lyon Cedex 07, France, May 1994. (short version in Transputers'94).

[22] R. W. Floyd, *Shortest path*, Comm. ACM, 5 (1962), p. 345.

[23] P. Fraigniaud and C. Gavoille, *A characterization of networks supporting linear interval routing*, in 13^{th} Annual ACM Symposium on Principles of Distributed Computing (PODC), ACM PRESS, Aug. 1994, pp. 216–224.

[24] ——, *Interval routing schemes*, Research Report 94-04, LIP, École Normale Supérieure de Lyon, 69364 Lyon Cedex 07, France, Jan. 1994.

[25] ——, *Optimal interval routing*, in Parallel Processing: CONPAR '94 - VAPP VI, B. Buchberger and J. Volkert, eds., vol. 854 of Lecture Notes in Computer Science, Springer-Verlag, Sept. 1994, pp. 785–796.

[26] P. Fraigniaud and C. Gavoille, *Order routing scheme*. Manuscript, Jan. 1994.

[27] P. Fraigniaud and C. Gavoille, *Memory requirement for universal routing schemes*, in 14^{th} Annual ACM Symposium on Principles of Distributed Computing (PODC), ACM PRESS, Aug. 1995, pp. 223–230.

[28] ——, *Memory requirement for universal routing schemes*, Research Report 95-05, LIP, École Normale Supérieure de Lyon, 69364 Lyon Cedex 07, France, Mar. 1995.

[29] P. Fraigniaud and C. Gavoille, *Universal routing schemes*. Manuscript submitted to JDC, June 1995.

[30] G. N. Frederickson and R. Janardan, *Designing networks with compact routing tables*, Algorithmica, 3 (1988), pp. 171–190.

[31] ——, *Efficient message routing in planar networks*, SIAM Journal on Computing, 18 (1989), pp. 843–857.

[32] ——, *Space-efficient message routing in c-decomposable networks*, SIAM Journal on Computing, 19 (1990), pp. 164–181.

[33] D. R. Fulkerson and O. A. Gross, *Incidence matrices and interval graphs*, Pacific Journal of Mathematics, 15 (1965), pp. 835–855.

[34] M. R. Garey and D. S. Johnson, *Computers and Intractability - A Guide to the Theory of NP-Completeness*, W.H. Freeman, 1977.

[35] M. R. Garey, D. S. Johnson, and H. C. So, *An application of graph coloring to printed circuit testing*, IEEE Trans. Circuits and Systems, CAS-23 (1976), pp. 591–599.

[36] C. Gavoille, *Routage par intervalles*, stage de DEA, École Normale Supérieure de Lyon, 69364 Lyon Cedex 07, France, June 1993.

[37] ——, *Routages efficaces et compacts*, in RenPar'6: 6èmes Rencontres Francophones du Parallélisme, L. Bougé, M. Cosnard, and P. Fraigniaud, eds., LIP, École Normale Supérieure de Lyon, 69364 Lyon Cedex 07, France, June 1994, p. 298.

[38] C. Gavoille and E. Guévremont, *On the compactness of bounded degree graphs for shortest path interval routing*, in 2^{nd} International Colloquium on Structural Information & Communication Complexity (SIROCCO), L. M. Kirousis and E. Kranakis, eds., Carleton University Press, June 1995, pp. 113–121.

[39] ——, *Worst case bounds for shortest path interval routing*, Research Report 95-02, LIP, École Normale Supérieure de Lyon, 69364 Lyon Cedex 07, France, Jan. 1995.

[40] C. Gavoille and S. Pérennès, *Memory requirement for routing in distributed networks*, Research Report 95-46, LIP, École Normale Supérieure de Lyon, 69364 Lyon Cedex 07, France, Dec. 1995.

[41] ——, *Routing memory requirement for distributed networks*, Research Report 95-37, I3S, Université de Nice-Sophia Antipolis, 06903 Sophia Antipolis, France, Aug. 1995.

[42] P. C. Gilmore and A. J. Hoffman, *A characterization of comparability graphs and of interval graphs*, Canadian Journal of Mathematics, 16 (1964), pp. 539–548.

[43] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, Harcourt Brace Jovanovich, Academic Press ed., 1980.

[44] F. Harary, *Graph Theory*, Addison-Wesley, 1969.

[45] M. A. Harrison, *On the number of classes of binary matrices*, IEEE Transactions on Computers, C-22 (1973), pp. 1048–1051.

[46] M.-C. Heydemann, *Réseaux modélisés par des graphes de Cayley*, 1991. notes de cours de DEA d'informatique 1991-1992.

[47] Inmos, *The T9000 Transputer Products Overview Manual*, 1991.

[48] G. Jacobson, *Space-efficient static trees and graphs*, in 30^{th} Annual Symposium on Foundations of Computer Science (FOCS), IEEE Computer Society Press, Oct. 1989, pp. 549–554.

[49] R. Janardan and W. Siu Cheng, *Efficient distributed algorithms for single-source shortest paths and related problems on plane networks*, in 4^{th} International Workshop on Distributed Algorithms (WDAG), G. Goos, J. Hartmauis, J. v. Leeuwen, and N. Santoro, eds., vol. 486 of Lecture Notes in Computer Science, Springer-Verlag, Sept. 1990, pp. 133–150.

[50] ——, *Efficient distributed algorithms for single-source shortest paths and related problems on plane networks*, Mathematical Systems Theory, 25 (1992), pp. 93–122.

[51] M. Juganaru, *Routage avec table de routage compacte*, stage de DEA, ENSIMAG, Grenoble, June 1993.

[52] S. Kannan, M. Naor, and S. Rudich, *Implicit representation of graphs*, in 20^{th} Annual ACM Symposium on Theory of Computing (STOC), Chicago, IL, May 1988, pp. 334–343.

[53] L. Kleinrock and F. Kamoun, *Hierarchical routing for large networks ; performance evaluation and optimization*, Computer Networks, 1 (1977), pp. 155–174.

[54] ——, *Optimal clustering structures for hierarchical topological design of large computer networks*, Networks, 10 (1980), pp. 221–248.

[55] E. Kranakis and D. Krizanc, *Boolean routing on Cayley networks*. Draft, Sept. 1995.

[56] ——, *Complexity of boolean routing*, Tech. Rep. TR-95-17, Carleton University, July 1995.

[57] ——, *Lower bounds for compact routing*, Tech. Rep. TR-95-18, Carleton University, July 1995. To appear in STACS '96.

[58] E. Kranakis, D. Krizanc, and S. S. Ravi, *On multi-label linear interval routing schemes*, in 19^{th} International Workshop on Graph - Theoretic Concepts in Computer Science - Distributed Algorithms (WG), vol. 790 of Lecture Notes in Computer Science, Springer-Verlag, June 1993, pp. 338–349.

[59] E. Kranakis, D. Krizanc, and J. Urrutia, *Compact routing and shortest path information*, in 2^{nd} International Colloquium on Structural Information & Communication Complexity (SIROCCO), L. M. Kirousis and E. Kranakis, eds., Carleton University Press, June 1995, pp. 101–112.

[60] D. Krizanc and F. L. Luccio, *Boolean routing on chordal rings*, in 2^{nd} International Colloquium on Structural Information & Communication Complexity (SIROCCO), L. M. Kirousis and E. Kranakis, eds., Carleton University Press, June 1995, pp. 89–100.

[61] S. Lakshmivarahan, J.-S. Jwo, and S. K. Dhall, *Symmetry in interconnaction networks based on Cayley graphs of permutation groups: A survey*, Parallel Computing, 19 (1993), pp. 361–407.

[62] F. T. Leighton, *Introduction aux algorithmes et architectures : Grilles, Arbres, Hypercubes*, International Thomson Publishing, 1995.

[63] M. Li and P. M. B. Vitányi, *An Introduction to Kolmogorov Complexity and its Applications*, Springer-Verlag, 1993.

[64] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North Holland, 1977.

[65] M. May, P. Thompson, and P. Welch, *Networks, routers and transputers: Function, perfomance, and applications*, tech. rep., Inmos, SGS-THOMSON, 1993.

[66] D. Peleg and E. Upfal, *Efficient message passing using succinct routing tables*, Research Report RJ 5768, IBM, 1987. Appear in 20^{th} Annual ACM Symposium on Theory of Computing (STOC), p43-52.

[67] ——, *A tradeoff between space and efficiency for routing tables*, in 20^{th} Annual ACM Symposium on Theory of Computing (STOC), Chicago, IL, May 1988, pp. 43–52.

[68] ——, *A trade-off between space and efficiency for routing tables*, Journal of the ACM, 36 (1989), pp. 510–530.

[69] R. Perlman, *Hierarchical networks and the subnetwork partition problem*, in 5^{th} Conference on System Sciences, 1982.

[70] F. Preparata and J. Vuillemin, *The cube connected cycles: a versatile network for parallel computation*, Communications of the ACM, 24 (1981), pp. 300–309.

[71] M. Raynal, *Algorithmes Distribués et Protocoles*, Eyrolles, 1985.

[72] J. P. Robison and M. Cohn, *Counting sequences*, IEEE Transactions on Computers, C-30 (1981), pp. 17–23.

[73] P. Ružička, *On efficiency of interval routing algorithms*, in 13^{rd} International Symposium on Mathematical Foundations of Computer Science (MFCS), M. Chytil, L. Janiga,

and V. Koubek, eds., vol. 324 of Lectures Notes in Computer Science, Springer-Verlag, 1988, pp. 492–500.

[74] H. J. Ryser, *Combinatorial configurations*, SIAM Journal on Applied Mathematics, 17 (1969), pp. 593–602.

[75] N. Santoro and R. Khatib, *Labelling and implicit routing in networks*, The Computer Journal, 28 (1985), pp. 5–8.

[76] H. S. Stone, *Parallel processing with the perfect shuffle*, IEEE Transactions on Computers, C-20 (1971), pp. 153–161.

[77] C. A. Sunshine, *Addressing problems in multi-networks systems*, IEEE INFOCOM, (1982).

[78] S. S. H. Tse and F. C. M. Lau, *Lower bounds for multi-label interval routing*, in 2^{nd} International Colloquium on Structural Information & Communication Complexity (SIROCCO), L. M. Kirousis and E. Kranakis, eds., Carleton University Press, June 1995, pp. 123–134.

[79] G. Turán, *Succinct representations of graphs*, Discrete Applied Mathematics, 8 (1984), pp. 289–294.

[80] J. van Leeuwen and R. B. Tan, *Interval routing*, The Computer Journal, 30 (1987), pp. 298–307.

[81] ——, *Compact routing methods: A survey*, in 1^{st} International Colloquium on Structural Information & Communication Complexity (SIROCCO), P. Flocchini, B. Mans, and N. Santoro, eds., Carleton University Press, May 1994, pp. 99–110.

[82] E. M. Wright, *Graphs on unlabelled nodes with a given number of edges*, Acta Math., 126 (1971), pp. 1–9.