

A. BEL HALA

**Congestion optimale du plongement de l'hypercube
 $H(n)$ dans la chaîne $P(2^n)$**

Informatique théorique et applications, tome 27, n° 5 (1993),
p. 465-481

http://www.numdam.org/item?id=ITA_1993__27_5_465_0

© AFCET, 1993, tous droits réservés.

L'accès aux archives de la revue « Informatique théorique et applications » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques

<http://www.numdam.org/>

CONGESTION OPTIMALE DU PLONGEMENT DE L'HYPERCUBE $H(n)$ DANS LA CHAÎNE $P(2^n)$ (*)

par A. BEL HALA ⁽¹⁾

Communiqué par R. CORI

Résumé. — Un plongement d'un graphe G dans un graphe H est donné par une application injective f des sommets de G vers les sommets de H et un routage ρ qui associe à chaque arête (u, v) du graphe G une chaîne dans H d'extrémité initiale $f(u)$ et d'extrémité finale $f(v)$.

Parmi les paramètres qui permettent de mesurer l'efficacité du plongement, il y a la congestion et la dilatation. La congestion est le nombre de chaînes passant par une arête (la charge d'un lien de communication), la dilatation est la distance maximale entre les images par f de deux sommets adjacents de G (dans le cas où ρ est un routage de plus court chemin).

Dans cet article, on s'intéresse au plongement de l'hypercube $H(n)$ dans la chaîne, qui minimise la congestion, et on montre que la valeur de cette congestion est

$$\text{cong}(H(n), P(2^n)) = 1/3 \times (2^{n+1} - 2 + (n \text{ modulo } 2)).$$

On termine cet article par une conjecture sur la valeur de la congestion pour le plongement de l'hypercube $H(n)$ dans le cycle $C(2^n)$:

$$\text{cong}(H(n), C(2^n)) = 1/3 \times (5 \times 2^{n-2} - 2 + (n \text{ modulo } 2)).$$

Abstract. — An embedding f of a guest graph G , into a host graph H , is a one-to-one mapping from each node i in G to a unique node $f(i)$ in H , and from each edge (i, j) in G to a path in H starting at node $f(i)$ and ending at node $f(j)$. The congestion of f is the maximum number of times any edge of H is used by edges of G . The minimum congestion, over all embeddings, is called the congestion of G into H , and denoted by $\text{cong}(G, H)$.

In this paper, we consider the problem of optimally embedding the vertices of hypercube graph $H(n)$, in the vertices of a line $P(2^n)$, in order to minimize the congestion; and we show that

$$\text{cong}(H(n), P(2^n)) = 1/3 \times (2^{n+1} - 2 + (n \text{ modulo } 2)).$$

(*) Reçu en novembre 1992; accepté en février 1993.

(1) LaBRI, Université Bordeaux-I, 351, cours de La Libération, 33405 Talence Cedex, e-mail: belhala@geocub.greco-prog.fr.

Finally, we conjecture that the value of optimal congestion for embedding hypercube $H(n)$ into cycle $C(2^n)$ is

$$\text{cong}(H(n), C(2^n)) = 1/3 \times (5 \times 2^{n-2} - 2 + (n \text{ modulo } 2)).$$

1. INTRODUCTION

Le choix du placement d'un programme distribué composé de n processus communicants sur une architecture distribuée possédant p processeurs peut être partiellement représenté par la recherche de toutes les applications de l'ensemble des processus dans l'ensemble des processeurs. L'acheminement de messages entre sites non voisins nécessite l'utilisation d'une fonction de routage. Une telle fonction définit le choix des chemins entre sites non voisins dans le réseau de communication.

Si on suppose qu'un processus ne requiert pas plus d'un processeur pour son exécution, il y a théoriquement p^n cas à étudier. L'étude de tous ces cas et même leur énumération ne peuvent être envisagées dans la plupart des cas réels. Le problème est de trouver parmi ces p^n cas, un meilleur placement.

En théorie des graphes ce problème porte le nom de plongement de graphes. On représente les n processus par un graphe G non orienté, où les sommets représentent les processus et il y a une arête entre deux processus s'ils communiquent entre eux. De la même manière l'architecture des processeurs est représentée par un graphe non orienté H , où les sommets sont les processeurs et une arête entre deux sommets s'il y a un lien physique entre les deux processeurs qui lui sont associés.

La théorie du plongement des graphes a pris une grande importance ces dernières années (pour un résumé, voir [11, 12]). Parmi les applications de la théorie des plongements, il y a l'émulation d'une architecture par une autre architecture d'une manière efficace (la possibilité pour un réseau d'interconnexion de pouvoir simuler efficacement un programme écrit pour une autre architecture). Le cas particulier du plongement d'un graphe G dans une chaîne (*i.e.* étiquetage des sommets de G par des entiers) est un problème important (voir [2, 3, 4, 10]). Dans ce cas, la congestion et la dilatation sont appelées respectivement *cutwidth* et *bandwidth*. Ce cas est intéressant dans la théorie de plongement de graphes dans la grille (voir [1, 3]). Leighton, Maggo et Rao montrent dans [9] que si un graphe G peut être plongé dans un graphe H avec une dilatation D et une congestion C , alors l'architecture H peut simuler T étapes d'un algorithme modélisé par l'architecture G en $O(C+D)T$ étapes. Le problème revient donc à trouver un plongement qui minimise ce facteur (*congestion + dilatation*).

Or dans la plupart des travaux sur les plongements, les auteurs cherchent surtout un plongement qui permet de minimiser la dilatation, ensuite ils calculent la congestion associée à ce plongement. Ce qui donne dans beaucoup de cas une congestion assez grande et par conséquent une mauvaise valeur du facteur (*congestion + dilatation*).

Comme le problème de la recherche d'un plongement, minimisant le facteur (*congestion + dilatation*) est assez difficile, on propose dans cet article, de chercher un plongement qui minimise la congestion (plongement à congestion optimale), calculer la dilatation associée à ce plongement, et enfin comparer le facteur obtenu (*congestion + dilatation*) avec celui du plongement minimisant la dilatation (plongement à dilatation optimale). Ceci nous permet de savoir lequel de ces deux plongements est meilleur.

L'hypercube est une architecture parmi les plus populaires; de nombreuses machines commercialisées sont de ce type; on peut prendre comme exemple les machines *iPSC* d'*Intel* et du *nCube*.

Dans cet article, on s'intéresse au plongement à congestion optimale de l'hypercube $H(n)$ dans la chaîne $P(2^n)$. Ensuite, on calcule la valeur du facteur $(C + D)$ (*congestion + dilatation*) pour deux plongements de l'hypercube dans la chaîne (le plongement à congestion optimale et le plongement de Harper [6] à dilatation optimale) ce qui nous permet de savoir lequel de ces deux plongements est le mieux adapté pour une émulation efficace de l'hypercube par la chaîne.

Connaissant le facteur $(C + D)$ pour ce plongement, on peut donc majorer par ce facteur, le facteur $(C + D)$ pour tout plongement de l'hypercube dans un graphe contenant une chaîne hamiltonienne. Dans [7], les auteurs montrent que pour certains graphes, la dilatation dans la chaîne est la même que celle dans le cycle, l'hypercube faisant partie de ces graphes. Connaissant donc la dilatation dans le cycle et sachant que la congestion dans le cycle est majorée par la congestion dans la chaîne, on a donc une estimation du facteur $(C + D)$ pour le plongement de l'hypercube dans le cycle.

On rappelle dans la section 2 quelques définitions de base sur les plongements. La valeur exacte de la congestion optimale est calculée dans la section 3. Dans la section 4 on calcule la congestion du plongement de Harper à dilatation optimale, puis on compare le facteur $(C + D)$ de ce plongement avec celui du plongement à congestion optimale. On termine cet article par une conjecture sur le plongement à congestion optimale de l'hypercube $H(n)$ dans l'anneau $C(2^n)$.

2. DÉFINITIONS DE BASE

Un n -cube $H(n)$ (i.e. hypercube de dimension n) est un graphe dont les sommets sont les nombres de 0 à $2^n - 1$. Deux sommets sont connectés par une arête, si leurs écritures binaires diffèrent en un seul bit. Une chaîne $P(n)$ est le graphe $(V_{P(n)}, E_{P(n)})$ où

$$V_{P(n)} = \{i, 0 \leq i < n\}, E_{P(n)} = \{e_i = (i-1, i), 1 \leq i \leq n\}.$$

Un plongement d'un graphe $G = (V_G, E_G)$ dans un graphe $H = (V_H, E_H)$ est défini par (f, ρ) , où $f: V_G \rightarrow V_H$ est une application injective et ρ un routage de $E_G \rightarrow E_H^*$ qui associe à chaque arête (u, v) de E_G une chaîne dans E_H^* , d'extrémité initiale $f(u)$ et d'extrémité finale $f(v)$.

Par abus de langage nous désignerons le plus souvent un plongement par le même nom que l'application f et nous ne préciserons pas toujours le routage ρ . Beaucoup de paramètres ont été définis pour mesurer l'efficacité des plongements. Parmi ces paramètres, la dilatation du plongement, noté $\text{dil}(f)$, est la longueur maximum des chaînes $\rho_f(u, v)$ associées aux arêtes (u, v) . Dans le cas où l'on considère des chaînes de plus courte longueur, la dilatation s'exprime uniquement en fonction de f ,

$$\text{dil}(f) = \max \{ \text{distance}(f(u), f(v)) : (u, v) \in E_G \}.$$

La dilatation minimum d'un plongement de G dans H , noté $\text{dil}(G, H)$, est le minimum de $\text{dil}(f)$ pris sur tous les plongements f de G dans H .

La congestion d'un plongement f d'un graphe G dans un graphe H , noté $\text{cong}(f)$, est le maximum, pris sur toutes les arêtes e de H de $c_f(e)$, où

$$c_f(e) = | \{ (u, v) \in E_G : e \in \rho(u, v) \} |.$$

La congestion minimum d'un plongement de G dans H , notée $\text{cong}(G, H)$, est le minimum de $\text{cong}(f)$ pris sur tous les plongements f de G dans H .

3. LA CONGESTION DU PLONGEMENT DE L'HYPERCUBE $H(n)$ DANS LA CHAÎNE $P(2^n)$

Soit f_n un plongement du n -cube $H(n)$ dans la chaîne $P(2^n)$.

Soit $e_i = (i-1, i) \in E_{P(2^n)}$, $1 \leq i \leq 2^n - 1$. La congestion de e_i est définie par :

$$c_{f_n}(e_i) = | \{ (u, v) \in H(n) : e_i \text{ est dans la chaîne allant de } f_n(u) \text{ à } f_n(v) \} |.$$

$e_i \in$ la chaîne allant de $f_n(u)$ à $f_n(v) \Leftrightarrow (f_n(u), f_n(v)) \in V_i \times (V_{P(2^n)} \setminus V_i)$, où $V_i = \{0, 1, \dots, i-1\}$.

$$c_{f_n}(e_i) = |\{u, v \in H(n), u \in f_n^{-1}(V_i) \text{ et } v \notin f_n^{-1}(V_i)\}|,$$

où

$$f_n^{-1}(V_i) = \{f_n^{-1}(0), f_n^{-1}(1), \dots, f_n^{-1}(i-1)\}.$$

Soit $K_{f_n, i} = (V_{f_n, i}, E_{f_n, i})$ le sous-graphe engendré par $V_{f_n, i} = f_n^{-1}(V_i)$, comme $H(n)$ est n -régulier, alors

$$c_{f_n}(e_i) = \sum_{v \in V_{f_n, i}} (n - d_{K_{f_n, i}}(v)),$$

où $d_{K_{f_n, i}}(v)$ est la valeur du degré du sommet v dans le graphe $K_{f_n, i}$. D'où

$$c_{f_n}(e_i) = i \times n - 2 |E_{f_n, i}|.$$

$|E_{f_n, i}|$ est le nombre d'arêtes du graphe $K_{f_n, i}$.

THÉORÈME : $\text{cong}(H(n), P(2^n)) = 1/3 \times (2^{n+1} - 2 + (n \text{ modulo } 2))$.

Ce théorème donne la valeur de la congestion optimale de l'hypercube $H(n)$ dans la chaîne $P(2^n)$. Les étapes de la démonstration sont les suivantes :

1. On définit un plongement $f_{c, n}$ pour lequel, on montre que la congestion est optimale (lemme 1).
2. On calcule pour tout $k, 1 \leq k \leq 2^n - 1$ la valeur de la congestion $c(k, n)$ (corollaire 1).
3. On montre que le maximum des valeurs $c(k, n)$ est atteint pour $k = k_n$ (corollaire 2).
4. On calcule enfin la valeur $c(k_n, n)$ de ce maximum (corollaire 3), qui est la valeur de la congestion optimale.

DÉFINITION 1 : Soit $G = (V_G, E_G)$ un sous-graphe du n -cube $H(n)$. On dit que le graphe G est k -maximal dans $H(n)$ ssi G a k -sommets et un nombre maximum d'arêtes. [i.e. si $G' = (V_{G'}, E_{G'})$, sous-graphe de $H(n)$, a k -sommets, alors $|E_G| \geq |E_{G'}|$].

Construction du sous-graphe G_k, k -maximal dans $H(n)$ ($1 \leq k \leq 2^n$)

Si $k = 1$, alors $G_1 = H(0)$, c'est le sous-graphe de $H(n)$ engendré par un sommet v_0 de $V_{H(n)}$.

Soit G_k le sous-graphe k -maximal dans $H(n)$ engendré par l'ensemble des sommets $V_k \subset V_{H(n)}$.

Le sous-graphe de $H(n)$ $(k+1)$ -maximal dans $H(n)$ est le sous-graphe engendré par $V_{k+1} = V_k \cup \{v_{k+1}\}$, où v_{k+1} est un sommet de $(V_{H(n)} \setminus V_k)$ ayant le nombre maximum de voisins dans le graphe G_k .

Cette construction est donnée par Harper dans [5].

Soit $f_{c,n}$ un plongement de $H(n)$ dans $P(2^n)$ tel que : $\forall i, 1 \leq i \leq 2^n, K_{f_{c,n},i}$ est i -maximal, où $K_{f_{c,n},i}$ est le sous-graphe de $H(n)$ engendré par $f_{c,n}^{-1}(V_i)$, avec

$$f_{c,n}^{-1}(V_i) = \{f_{c,n}^{-1}(0), f_{c,n}^{-1}(1), \dots, f_{c,n}^{-1}(i-1)\}.$$

Un exemple de ce plongement, obtenu par l'application identité I , est illustré par la figure 1 ci-dessous.

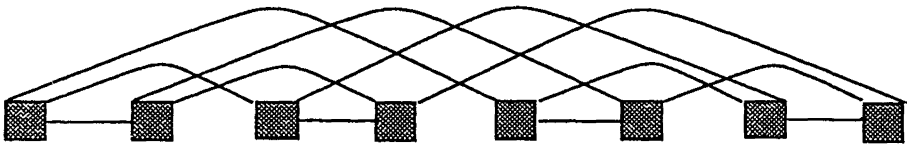


Figure 1. - Plongement $f_{c,3}$ de $H(3)$ dans $P(2^3)$.

Dans la suite, on désigne $c_{f_{c,n}}(e_i)$ par $c(i, n)$.

Remarque 1 : (1) Un tel plongement existe, car l'identité possède cette propriété.

(2) $K_{f_{c,n}}$, le sous-graphe de $H(n)$ engendré par

$$f_{c,n}^{-1}(V_{2^{n-1}}) = \{f_{c,n}^{-1}(0), f_{c,n}^{-1}(1), \dots, f_{c,n}^{-1}(2^{n-1}-1)\}$$

est isomorphe à l'hypercube $H(n-1)$, donc tout sommet v de $f_{c,n}^{-1}(V_{2^{n-1}})$ a $(n-1)$ voisins dans $f_{c,n}^{-1}(V_{2^{n-1}})$ et un voisin dans $V_{H(n)} \setminus f_{c,n}^{-1}(V_{2^{n-1}})$. La distance entre $v = f_{c,n}^{-1}(0)$ et son voisin dans $f_{c,n}^{-1}(V_{2^{n-1}})$ est $\geq 2^{n-1}$, donc la dilatation de $f_{c,n}$ est $\geq 2^{n-1}$.

(3) Si $f_{c,n} = I$ (identité), alors la dilatation est égale à 2^{n-1} . En effet, la distance entre deux sommets de $f_{c,n}^{-1}(V_{2^{n-1}})$ est $\leq 2^{n-1}$, la distance entre un sommet de $f_{c,n}^{-1}(V_{2^{n-1}})$ et un sommet de $V_{H(n)} \setminus f_{c,n}^{-1}(V_{2^{n-1}})$ est égale à 2^{n-1} .

(4) $(\forall i, 1 \leq i \leq 2^n - 1)$ on a : $c(i, n) = c(2^n - i, n)$ [Le sous-graphe de $H(n)$ engendré par $\{f_{c,n}^{-1}(i), f_{c,n}^{-1}(i+1), \dots, f_{c,n}^{-1}(2^n - 1)\}$ est $(2^n - i)$ -maximal dans $H(n)$].

(5) $\forall 1 \leq i \leq 2^n - 1 : c(i, n+1) = c(i, n) + i$. En effet, si G_i est i -maximal dans $H(n)$, alors G_i est i -maximal dans $H(n+1)$.

LEMME 1 : Soit f_n un plongement de $H(n)$ dans $P(2^n)$, alors :

$$(\forall e_i \in E_{P(2^n)}), \quad c_{f_n}(e_i) \geq c(i, n).$$

Preuve : Pour tout $e_i \in E_{P(2^n)}$, $1 \leq i \leq 2^n - 1$ on a $c_{f_n}(e_i) = i \times n - 2 |E_{f_n, i}|$, où

$$|E_{f_n, i}| = |\{(u, v) \in E_{H(n)}, (u, v) \in f_n^{-1}(V_i) \times f_n^{-1}(V_i)\}|$$

avec

$$f_n^{-1}(V_i) = \{f_n^{-1}(0), f_n^{-1}(1), \dots, f_n^{-1}(i-1)\}.$$

$K_{f_n, i} = (f_n^{-1}(V_i), E_{f_n, i})$ est i -maximal dans $H(n)$, donc

$$|E_{f_n, i}| \geq |E_{f_n, i}| \Rightarrow c_{f_n}(e_i) \geq c_{f_n, i}(e_i) = c(i, n). \quad \square$$

DÉFINITION 2 : Soient $H = (V_H, E_H)$ et $G = (V_G, E_G)$ deux graphes tels que H est un sous-graphe de G (i.e. il existe une application injective f de V_H dans V_G telle que : $((v, w) \in E_H \Rightarrow (f(v), f(w)) \in E_G)$.

Le graphe $G \mp H$ est défini par :

$$V_{G \mp H} = V_G \cup V_H \quad \text{et} \quad E_{G \mp H} = E_G \cup E_H \cup \{(u, f(u)), u \in V_H\},$$

où \cup représente l'union disjointe.

L'exemple ci-dessous de la figure 2, donne une illustration des graphes $H(1) \mp H(2)$ et $H(2) \mp H(2)$.

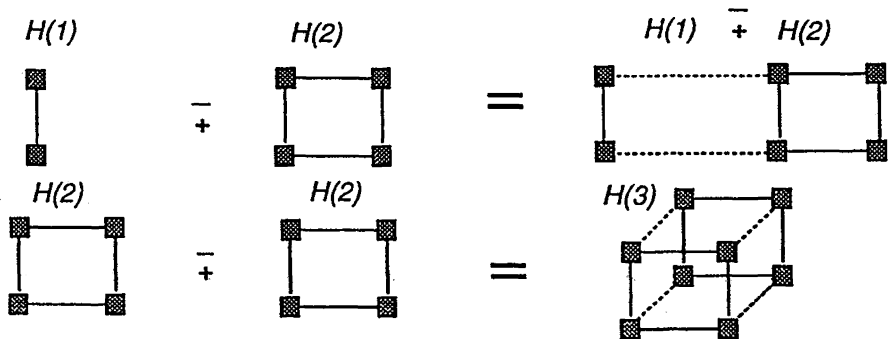


Figure 2. - Graphes $H(1) \mp H(2)$ et $H(2) \mp H(2)$.

Remarque 2 : La définition du graphe $G \mp H$ dépend de l'application f . Si on ne précise pas cette application f , le graphe $G \mp H$ est défini à un isomorphisme près.

PROPRIÉTÉ 1 : Soit G_k un sous-graphe de l'hypercube $H(m)$ k -maximal avec $m < n$.

- (1) $H(m)$ est 2^m -maximal dans $H(n)$
- (2) $G_k \bar{\mp} H(m)$ est $(k + 2^m)$ -maximal dans $H(n)$.

Preuve : (1) D'après la remarque 1.1, le graphe $K_{I, 2^m}$ est 2^m -maximal. Or $K_{I, 2^m} = H(m)$, donc $H(m)$ est 2^m -maximal.

(2) (par récurrence sur k).

Le graphe $(k + 2^m)$ -maximal est construit comme suit :

– On construit d'abord le graphe $G_{2^m}(2^m)$ -maximal [d'après (1) $G_{2^m} = H(m)$].

– On ajoute ensuite les k -sommets au graphe G_{2^m} de telle façon à ajouter un nombre maximum d'arêtes.

Sachant que les k sommets qu'on ajoute ont au plus k voisins dans G_{2^m} et que G_k est k -maximal, alors par définition de $\bar{\mp}$, $G_k \bar{\mp} H(m)$ est $(k + 2^m)$ -maximal dans $H(n)$. \square

PROPRIÉTÉ 2 : Soit k , $k = 2^{n_1} + 2^{n_2} + \dots + 2^{n_{p_k}}$, $n_1 < n_2 < \dots < n_{p_k}$ [$p_k = \text{poids}(k) = \text{nombre de 1 dans l'écriture binaire de } k$]. Soit le graphe G_k défini par

$$G_k = ((\dots((H(n_1) \bar{\mp} H(n_2)) \bar{\mp} H(n_3)) \bar{\mp} \dots) \bar{\mp} H(n_{p_k-1})) \bar{\mp} H(n_{p_k})).$$

Pour tout n , $n \geq k$, le graphe $G_k = (V_k, E_k)$ est k -maximal dans $H(n)$.

Preuve : La preuve se fait par récurrence sur le poids de k , p_k .

$$p_k = 1 \Rightarrow k = 2^{n_1} \Rightarrow G_k = H(n_1).$$

La propriété est vraie d'après la propriété précédente.

$$p_k = 2 \Rightarrow k = 2^{n_1} + 2^{n_2} \Rightarrow G_k = H(n_1) \bar{\mp} H(n_2).$$

D'après la propriété 1, $H(n_1) \bar{\mp} H(n_2)$ est $(2^{n_1} + 2^{n_2})$ -maximal.

On suppose que pour tout k , $1 \leq p_k \leq l$, G_k est k -maximal.

Soit k , tel que $p_k = l + 1 \Rightarrow k = k' + 2^{n_{l+1}}$, où $k' = 2^{n_1} + 2^{n_2} + \dots + 2^{n_l}$.

D'après l'hypothèse de récurrence $G_{k'}$ est k' -maximal. $H(n_{l+1})$ est $2^{n_{l+1}}$ -maximal.

Donc d'après la propriété 1, G_k est k -maximal. \square

Une illustration du graphe G_{11} est donnée par la figure 3 ci-après.

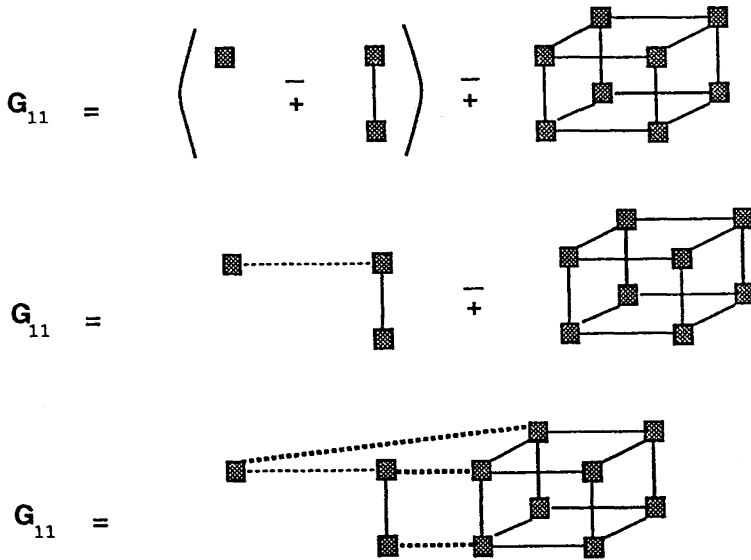


Figure 3. - G_{11} , graphe 11-maximal dans $H(n)$.

$$11 = 2^0 + 2^1 + 2^3, \quad n_1 = 0 < n_2 = 1 < n_3 = 3 \quad \text{et} \quad p_k = 3,$$

donc

$$G_{11} = (H(0) \mp H(1)) \mp H(3).$$

LEMME 2 : Soit $k = 2^{n_1} + 2^{n_2} + \dots + 2^{n_{p_k}}$, $n_1 < n_2 < \dots < n_{p_k-1} < n_{p_k}$.
 Pour le graphe $G_k = (V_k, E_k)$ k -maximal dans $H(n)$ (propriété 2), on a

$$2|E_k| = \sum_{i=1}^{p_k} n_i 2^{n_i} + 2 \sum_{i=1}^{p_k-1} (p_k - i) 2^{n_i}.$$

Dans la suite, on définit $m(G) = 2|E|$ pour un graphe $G = (V, E)$.

Preuve : (par récurrence sur p_k) : $p_k = 1 \Rightarrow k = 2^{n_1}$, $G_k = H(n_1)$, donc $m(G_k) = 2|E_{H(n_1)}| = n_1 2^{n_1}$.

$p_k = 2 \Rightarrow k = 2^{n_1} + 2^{n_2}$, $G_k = H(n_1) \mp H(n_2)$ (propriété 2), donc

$$m(G_k) = m(H(n_1)) + m(H(n_2)) + 2|V_{H(n_1)}| = n_1 2^{n_1} + n_2 2^{n_2} + 2 \cdot 2^{n_1}.$$

On suppose que pour $k = 2^{n_1} + 2^{n_2} + \dots + 2^{n_{p_k}}$, le lemme est vérifié.

Soit k' tel que $p_{k'} = p_k + 1$, donc $G_{k'} = G_k \mp H(n_{p_k+1})$ car $k' = k + 2^{n_{p_k+1}}$.

$$m(G_{k'}) = m(G_k) + m(H(n_{p_k+1})) + 2|V_{G_k}|.$$

$$\begin{aligned}
 m(G_{k'}) &= \left(\sum_{i=1}^{i=p_k} n_i 2^{n_i} \right) + 2 \left(\sum_{i=1}^{i=p_k-1} (p_k - i) 2^{n_i} \right) + n_{p_k+1} 2^{n_{p_k+1}} + 2 \left(\sum_{i=1}^{i=p_k} 2^{n_i} \right). \\
 m(G_{k'}) &= \sum_{i=1}^{i=p_k+1} n_i 2^{n_i} + 2 \sum_{i=1}^{i=p_k-1} (p_k - i) 2^{n_i} + 2 \sum_{i=1}^{i=p_k} 2^{n_i}. \\
 m(G_{k'}) &= \sum_{i=1}^{i=p_k+1} n_i 2^{n_i} + 2 \sum_{i=1}^{i=p_k} (p_k + 1 - i) 2^{n_i}.
 \end{aligned}$$

D'où le lemme. \square

COROLLAIRE 1 : Soit $k = 2^{n_1} + 2^{n_2} + \dots + 2^{n_{p_k}}$, $n_1 < n_2 < \dots < n_{p_k-1} < n_{p_k}$.

$$c(k, n) = k \times n - \left[\sum_{i=1}^{i=p_k} n_i 2^{n_i} + 2 \sum_{i=1}^{i=p_k-1} (p_k - i) 2^{n_i} \right].$$

Preuve : Le corollaire s'obtient en remplaçant dans la formule

$$c(k, n) = k \times n - 2 |E_{G_k}|, |E_{G_k}|$$

par la valeur trouvée dans le lemme 2, qui est le nombre d'arêtes du graphe k -maximal dans $H(n)$.

Dans l'exemple ci-dessous, on montre comment calculer $c(13,4)$, puis on donne les valeurs de $c(k, 4)$, $1 \leq k \leq 15$.

Exemple : $13 = 2^0 + 2^2 + 2^3$, $n_1 = 0 < n_2 = 2 < n_3$ et $p_{13} = 3$, donc

$$c(13,4) = 13 \times 4 - \left[\sum_{i=1}^{i=3} n_i 2^{n_i} + 2 \sum_{i=1}^{i=2} (3 - i) 2^{n_i} \right] = 8.$$

$$\begin{aligned}
 c(1,4) &= c(15,4) = 4, \quad c(2,4) = c(14,4) = 6, \quad c(3,4) = c(13,4) = 8, \\
 c(4,4) &= c(12,4) = 8, \quad c(5,4) = c(11,4) = 10, \quad c(6,4) = c(10,4) = 10, \\
 c(7,4) &= c(9,4) = 10 \text{ et } c(8,4) = 8.
 \end{aligned}$$

LEMME 3 : Soit k_n la plus petite valeur de i pour laquelle $c(i, n)$ est maximum ($k_n = \min \{ 1 \leq i \leq 2^n - 1 : c(i, n) = \max_{1 \leq k \leq 2^n - 1} c(k, n) \}$). Alors

$$\begin{cases} k_1 = 1 \\ k_n = 2^{n-1} - k_{n-1}, \quad \forall n \geq 2 \end{cases}$$

Preuve : (par récurrence sur n) : $c(1,1) = 1$, donc $k_1 = 1$.

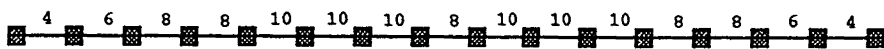


Figure 4. - $c(k, 4), 1 \leq k \leq 15$.

$c(1,2) = c(2,2) = c(3,2) = 2$, donc $k_2 = 1$. Le lemme est donc vérifié pour $n = 2$.

$\forall i, 1 \leq i \leq 2^n - 1 : c(i, n) = c(2^n - i, n)$, donc $k_n \in \{1, 2, \dots, 2^{n-1}\}$.

$c(2^{n-1} - 1, n) = c(2^{n-1}, n) + n - 2$. Donc $\forall n \geq 2, k_n \in \{1, 2, \dots, 2^{n-1} - 1\}$.

Il suffit donc de montrer que

$$2^{n-1} - k_{n-1} = \min \{ 1 \leq i \leq 2^{n-1} - 1 : c(i, n) = \max_{1 \leq k \leq 2^{n-1}} c(k, n) \}$$

Supposons que pour tout $m < n, k_m = 2^{m-1} - k_{m-1}$.

Montrons que $k_n = x_n$ où $x_n = 2^{n-1} - k_{n-1}$.

• Soit $k, 1 \leq k < x_n$,

$$c(k, n) - c(x_n, n) = c(k, n-1) - c(x_n, n-1) + (k - x_n) \text{ (car } k, x_n \in [1, 2^{n-1} - 1]).$$

Or $c(x_n, n-1) = c(2^{n-1} - k_{n-1}, n-1) = c(k_{n-1}, n-1)$, donc d'après l'hypothèse de récurrence $(c(k, n-1) - c(k_{n-1}, n-1)) \leq 0$. Comme $k < x_n$, alors $c(k, n) < c(x_n, n)$.

• Soit $k, x_n < k < 2^{n-1}$. Donc

$$c(k, n) - c(x_n, n) = c(k, n-1) - c(x_n, n-1) + k - x_n \text{ (car } k, x_n \in [1, 2^{n-1} - 1]).$$

$c(k, n) - c(x_n, n) = c(k', n-1) - c(k_{n-1}, n-1) - k' + k_{n-1}$, où $k' = 2^{n-1} - k$ (car $c(i, n-1) = c(2^{n-1} - i, n-1)$).

$k > x_n$ donc $k' < k_{n-1} = 2^{n-2} - k_{n-2}$, or $k_{n-1} \in [1, 2^{n-2} - 1]$, donc $c(k', n-1) - c(k_{n-1}, n-1) = c(k', n-2) - c(k_{n-1}, n-2) + k' - k_{n-1}$.

D'où $c(k, n) - c(x_n, n) = c(k', n-2) - c(k_{n-1}, n-2)$.

D'après l'hypothèse de récurrence $k_{n-1} = 2^{n-2} - k_{n-2}$, donc

$$c(k', n-2) - c(k_{n-2}, n-2) \leq 0.$$

D'où $c(k, n) - c(x_n, n) \leq 0$.

On a donc montré que

$$x_n = 2^{n-1} - k_{n-1} = \min \{ 1 \leq i \leq 2^{n-1} - 1 : c(i, n) = \max_{1 \leq k \leq 2^{n-1}} c(k, n) \}.$$

D'où le lemme ($k_n = 2^{n-1} - k_{n-1}$). □

COROLLAIRE 2

$$\left\{ \begin{array}{l} k_{2n} = \sum_{i=0}^{n-1} 2^{2i} = 1/3 (2^{2n} - 1) \\ k_{2n+1} = 4^n - k_{2n} \end{array} \right.$$

Preuve : D'après le lemme 3, $k_{2n} = 2^{2n-1} - k_{2n-1}$ et $k_{2n+1} = 2^{2n} - k_{2n}$.

$$k_{2n} = 2^{2n-1} - k_{2n-1} = 2^{2n-1} - 2^{2n-2} + k_{2n-2}.$$

$$(k_2 = 1 \text{ et } k_{2n} = k_{2n-2} + 2^{2n-2}) \Rightarrow k_{2n} = \sum_{i=0}^{i=n-1} 2^{2i} \quad (= 1/3 \cdot (4^n - 1)).$$

$$k_{2n+1} = 2^{2n} - k_{2n} = 4^n - k_{2n}. \quad \square$$

LEMME 4 : Si $k_{2n} = \sum_{i=0}^{n-1} 2^{2i}$, alors

$$m(G_{k_{2n}}) = 2/3 \times (n-1) \times (4^n - 1).$$

$1/2 \times m(G_{k_{2n}})$ est le nombre d'arêtes du graphe k_{2n} -maximal.

Preuve : $k_{2n} = \sum_{i=0}^{n-1} 2^{2i}$ (corollaire 2), donc d'après le lemme 2 :

$$m(G_{k_{2n}}) = \sum_{i=1}^n 2(i-1)2^{2(i-1)} + 2 \sum_{i=1}^{n-1} (n-i)2^{2(i-1)}.$$

$$m(G_{k_{2n}}) = 2(n-1)2^{2(n-1)} + 2(n-1) \sum_{i=1}^{n-1} 2^{2(i-1)}.$$

$$m(G_{k_{2n}}) = 2(n-1) \sum_{i=1}^n 2^{2(i-1)}.$$

$$m(G_{k_{2n}}) = 2(n-1) \sum_{i=1}^n 4^{i-1}.$$

D'où $m(G_{k_{2n}}) = 2/3 \times (n-1)(4^n - 1)$. \square

COROLLAIRE 3

$$c(k_n, n) = 1/3 \times (2^{n+1} - 2 + (n \text{ modulo } 2)).$$

Preuve : D'après le lemme 4, $2|E_{m_{2n}}| = m(G_{k_{2n}}) = 2/3 \times (n-1) \times (4^n - 1)$, or $c(k_{2n}, 2n) = 2n \times k_{2n} - m(G_{k_{2n}})$, donc

$$\begin{aligned} c(k_{2n}, 2n) &= 2n \times 1/3 \times (4^n - 1) - 2/3 \times (n-1) \times (4^n - 1) = 2/3 \times (4^n - 1). \\ c(k_{2n+1}, 2n+1) &= c(4^n - k_{2n}, 2n+1) \\ &= c(2^{2n} - k_{2n}, 2n) + 2^{2n} - k_{2n} \\ &= c(k_{2n}, 2n) + 4^n - 1/3(4^n - 1). \end{aligned}$$

D'où $c(k_{2n+1}, 2n+1) = 1/3 \times (4^{n+1} - 1)$. On a donc :

$$c(k_n, n) = 1/3 \times (2^{n+1} - 2 + (n \text{ modulo } 2)). \quad \square$$

Preuve du théorème : D'après le lemme 1, on a

$$\text{cong}(H(n), P(2^n)) = \max_{1 \leq i \leq 2^n - 1} c(i, n).$$

Corollaire 3 $\Rightarrow \max_{1 \leq i \leq 2^n - 1} c(i, n) = 1/3 \times (2^{n+1} - 2 + (n \text{ modulo } 2))$. D'où le théorème.

4. COMPARAISON DU PLONGEMENT DE HARPER ET PLONGEMENT A CONGESTION OPTIMALE

Soit $f_{bw, n}$ le plongement de $H(n)$ dans $P(2^n)$ défini sur $[0, 2^n - 1]$ par :

$$\forall i, j, \quad 0 \leq i, j \leq 2^n - 1 \quad \left\{ \begin{array}{l} f_{bw}(0) = 0, \\ f_{bw}(2^n - 1) = 2^n - 1, \\ f_{bw}(i) < f_{bw}(j) \quad \text{si } p_i < p_j, \\ f_{bw}(i) < f_{bw}(j) \quad \text{si } ((p_i = p_j) \text{ et } (i > j)). \end{array} \right.$$

Où p_i représente le poids de i , c'est-à-dire le nombre de bit à 1 dans la représentation binaire de i .

Ce plongement $f_{bw, n}$ est appelé plongement par poids. Une illustration de ce plongement est donnée par la figure 5 ci-après, pour $n = 3$.

PROPOSITION 1 (Harper [6]) : 1. La dilatation de $f_{bw, n}$ est $\lambda_n = \sum_{k=0}^{n-1} \binom{k}{\lfloor k/2 \rfloor}$.

2. Pour tout plongement f de $H(n)$ dans $P(2^n)$, la dilatation est au moins λ_n .

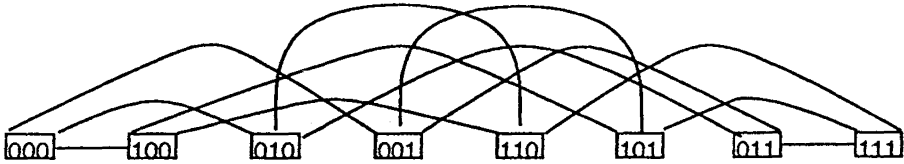


Figure 5. – Plongement de Harper pour $n=3$ ($f_{bw, 3}$).

Congestion du plongement $f_{bw, n}$

Pour tout $r, 0 \leq r \leq n$, on note x_r le nombre $x_r = \sum_{i=0}^r \binom{n}{i}$.

Soit $V_{x_r} = \{v \in V_{H(n)} : \text{le poids de } v, p_v \leq r\}$.

Le cardinal de V_{x_r} est $|V_{x_r}| = x_r$.

Soit $K_{f_{bw, n}, k}$ le sous-graphe de l'hypercube $H(n)$, induit par l'ensemble de sommets $f_{bw, n}^{-1}(V_k)$ où $f_{bw, n}^{-1}(V_k) = \{f_{bw, n}^{-1}(0), f_{bw, n}^{-1}(1), \dots, f_{bw, n}^{-1}(k-1)\}$.

$\forall e_k \in E_{\mathbb{P}(2^n)}$ on a $c_{f_{bw, n}}(e_k) = k \cdot n - 2 |E_{f_{bw, n}, k}|$, où $E_{f_{bw, n}, k}$ est l'ensemble des arêtes du graphe $K_{f_{bw, n}, k}$.

PROPOSITION 2 : (1) Pour tout $k, k = x_r + r'$, avec $0 < r' < \binom{n}{r+1}$:

$$c_{f_{bw, n}}(e_k) = k \times n - 2 \left[(k - x_r)(r + 1) + \sum_{i=0}^r i \binom{n}{i} \right].$$

(2) $\text{cong}(f_{bw, n}) = c_{f_{bw, n}}(e_{\lceil n/2 \rceil - 1}) = \lceil n/2 \rceil \binom{n}{\lceil n/2 \rceil}$.

Preuve : (1) L'ensemble de sommets $V_{f_{bw, n}, k}$ contient $(k - x_r)$ sommets de poids $r + 1$ et $\binom{n}{i}$ sommets de poids i ($\forall i, 0 \leq i \leq r$).

Si un sommet de $f_{bw, n}^{-1}(V_k)$ est de poids i , alors il a i voisins (de poids $i - 1$) dans le graphe $K_{f_{bw, n}, k}$. D'où $|E_{f_{bw, n}, k}| = (k - x_r)(r + 1) + \sum_{i=0}^r i \binom{n}{i}$.

Or, $c_{f_{bw, n}}(e_k) = k \cdot n - 2 |E_{f_{bw, n}, k}|$, d'où (1).

(2) Si $\max_{1 \leq k \leq 2^n - 1} c_{f_{bw, n}}(e_k) = c_{f_{bw, n}}(e_m)$, alors $m = x_r$, où $0 \leq r \leq n$.

En effet, supposons que $m = x_r + r'$ avec $0 < r' < \binom{n}{r+1}$,

$$c_{f_{bw,n}}(e_{x_{r+1}}) = c_{f_{bw,n}}(e_m) + (x_{r+1} - m)(r+1) > c_{f_{bw,n}}(e_m).$$

Or $c_{f_{bw,n}}(e_{x_r}) = (r+1) \binom{n}{r+1}$.

D'où $\text{cong}(f_{bw,n}) = \max_{0 \leq r \leq n} (r+1) \binom{n}{r+1} = \lceil n/2 \rceil \binom{n}{\lceil n/2 \rceil}$. \square

PROPOSITION 3 (Leighton [9]) : *Si G peut être plongé dans H avec une dilatation D et une congestion C , alors l'architecture H peut simuler T étapes d'un algorithme modélisé par l'architecture G (pour un algorithme général) en $O(C+D) T$ étapes.*

Pour le plongement à congestion optimale $f_{c,n} = I$ (plongement identité), on a la dilatation $D = 2^{n-1}$, le facteur $(C+D)$ a pour valeur

$$(C+D)_{f_{c,n}} = 2^{n-1} + 1/3 \times (2^{n+1} - 2 + (n \text{ modulo } 2)).$$

Pour le plongement à dilatation optimale $f_{bw,n}$, ce facteur a pour valeur

$$(C+D)_{f_{bw,n}} = \lambda_n + \lceil n/2 \rceil \binom{n}{\lceil n/2 \rceil}.$$

L'article de Ten-Hwang Lai [8] montre que :

$$\lambda_n \sim \binom{n}{\lfloor n/2 \rfloor} \sim 2^n \sqrt{2/(n\pi)}.$$

$$\lceil n/2 \rceil = \lfloor (n+1)/2 \rfloor, \text{ donc } \binom{n}{\lceil n/2 \rceil} \sim 2^{n+1} \sqrt{2/((n+1)\pi)}.$$

$$(C+D)_{f_{bw,n}} \sim 2^n \sqrt{2/(n\pi)} + \lceil n/2 \rceil 2^{n+1} \sqrt{2/((n+1)\pi)}.$$

D'où

$$(C+D)_{f_{c,n}} = O(2^n) \quad \text{et} \quad (C+D)_{f_{bw,n}} = O(2^n \sqrt{n}).$$

Ceci montre que pour émuler l'hypercube $H(n)$ par la chaîne $P(2^n)$, le plongement à congestion optimale $f_{c,n}$ est plus intéressant que celui de Harper à dilatation optimale $f_{bw,n}$.

Pour tout plongement f_n de $H(n)$ dans $P(2^n)$ on a

$$\text{cong}(f_n) \geq 1/3 \times (2^{n+1} - 2 + (n \text{ modulo } 2)),$$

donc $\text{cong}(f_n)$ est au moins de l'ordre $\Omega(2^n)$. Ce qui implique que le facteur $(C+D)_{f_n}$ de ce plongement est minoré par $O(2^n)$.

On peut dire que le plongement de l'hypercube $H(n)$ dans un graphe contenant une chaîne hamiltonienne peut se faire de manière à avoir $(C+D) = O(2^n)$.

5. CONCLUSION

Le but de la plupart des travaux sur les plongements est de chercher un plongement qui permet de minimiser la dilatation, puis de calculer la congestion associée à ce plongement. Ceci nous donne dans beaucoup de cas une congestion assez grande et par conséquent une mauvaise valeur du facteur (*congestion + dilatation*).

Dans cet article, nous avons calculé la valeur exacte de la congestion optimale du plongement de l'hypercube dans la chaîne (*cutwidth* de l'hypercube) et la valeur de la congestion pour le plongement de Harper. Ceci nous permet de voir que dans certains cas, il est plus intéressant de chercher à minimiser la congestion plutôt que la dilatation. Cette remise en cause de l'approche classique, dans laquelle on cherchait avant tout à minimiser la valeur de la dilatation, même si cela entraîne une mauvaise congestion pour le plongement, permet d'espérer de meilleurs plongements pour un grand nombre de graphes. Par exemple, on peut espérer plonger l'hypercube dans le graphe de De Bruijn avec une dilatation et une congestion linéaire (ce qui serait mieux que la plongement à dilatation $n/2$ et une congestion de l'ordre $O(n^2)$). Notre résultat sur le calcul de la congestion optimale du plongement de l'hypercube $H(n)$ dans la chaîne $P(2^n)$ nous permet d'énoncer la conjecture suivante :

La congestion optimale de l'hypercube $H(n)$ dans l'anneau $C(2^n)$ vérifie :

$$\text{cong}(H(n), C(2^n)) = 1/3 \times (5 \times 2^{n-2} - 2 + (n \text{ modulo } 2)).$$

Nous avons vérifié cette conjecture pour certaines valeurs de n et cherchons actuellement à la démontrer pour toute valeur de n .

BIBLIOGRAPHIE

1. S. N. BHATT et F. T. LEIGHTON, A framework for solving VLSI graph layout problems, *J. Comput. System Sci.*, 28, 1984, p. 300-343.

2. P. Z. CHINN, J. CHVÁTALOVÁ, A. K. DEWDNEY et N. E. GIBBS, The bandwidth problem for graphs and matrices-A survey, *J. Graph Theory*, 6, 1982, p. 223-254.
3. F. R. K. CHUNG, Labelings of graphs, in *Selected Topics in Graph Theory, III* (L. Beineke and R. Wilson, Eds.), Academic Press, 1988, p. 151-168.
4. F. R. K. CHUNG et P. D. SEYMOUR, manuscript, Bell Communication Research, *Some results on the bandwidth and the cutwidth of a graph*, 1987.
5. L. H. HARPER, Optimal assignments of numbers to vertices, *J. Soc. Indust. Appl. Math.*, 12, 1964, p. 131-135.
6. L. H. HARPER, Optimal numberings and isoperimetric problems on graphs, *J. of Combinatorial Theory*, 1, 1966, p. 385-393.
7. J. HROMKOVIČ, V. MULLER, O. SYKORA et I. VRTO, *On embedding in cycles* (to appear).
8. TEN-HWANG LAI et Alan P. SPRAGUE, Placement of the Processors of a Hypercube, *IEEE-Trans.-Comput.* 40, 6, 1991, p. 714-722.
9. LEIGHTON, MAGGO, RAO, Universal packet routing algorithms, *29th FOCS*, 1988, p. 256-271.
10. F. MAKEDON, C. H. PAPADIMITRIOU et I. H. SUDBROUGH, Topological bandwidth, *SIAM J. Algebraic Discrete Methods*, 6, 1985, p. 418-444.
11. B. MONIEN et I. H. SUDBROUGH, Comparing Interconnection Networks, *Proceedings of the 13th Symposium on mathematical Foundations of Computer Science*, 1988.
12. B. MONIEN et I. H. SUDBROUGH, Embedding one Interconnection Network in Another, *Computing Suppl.*, 7, 1990, p. 257-282.