Redistribution de données entre deux grappes d'ordinateurs

Nicolas Padoy

Redistribution de données entre deux grappes d'ordinateurs

Equipe Resedas

Nicolas Padoy

Encadrants : Johanne Cohen
Emmanuel Jeannot

Laboratoire Lorrain de Recherche en Informatique et ses Applications
Vandœuvre-lès-Nancy

1 Introduction

Dans le cadre d'un calcul distribué sur une grappe d'ordinateurs, il peut éventuellement être nécessaire à un moment donné de déplacer les données vers une autre grappe, afin d'utiliser par exemple d'autres ressources. Il faut alors transférer les données en essayant d'utiliser au mieux le réseau qui interconnecte les deux grappes. Le sujet de mon stage porte sur un cas particulier d'une telle redistribution.

Après avoir présenté (section 2) et modélisé (section 3) le problème, nous parlerons d'abord de sa complexité (section 4.1). Sa complexité nous invitera ensuite à chercher des algorithmes d'approximation. Nous en présenterons un (section 4.6 et 4.7), après avoir étudié au préalable une borne inférieure du coût d'une solution (section 4.2), qui servira à l'évaluer. Finalement, nous terminerons avec quelques heuristiques (section 4.8) qui se révélent efficaces en pratique.

2 Présentation du problème

Nous considérons la connection unidirectionnelle de deux grappes d'ordinateurs, que nous supposerons de même taille n sans perte de généralité, présentée figure 1. Chaque nœud est relié à un backbone[1] de débit D par l'intermédiaire d'un câble de débit d ou d' selon la grappe à laquelle il appartient. Dans la suite nous ne considérerons que le débit limitant, que nous appellerons aussi d. Chaque nœud P_i doit transmettre la quantité de données (éventuellement nulle) $q_{i,j}$ au nœud P'_j, sous les conditions suivantes :

1. Chaque nœud ne peut participer qu'à une seule communication à la fois, aussi bien comme émetteur que comme récepteur. Les nœuds communiquent donc par paires.
2. Plusieurs nœuds peuvent communiquer simultanément, dans la limite du débit D.
3. Les communications s'effectuent par étapes. Avant chaque étape sont décidées quelles sont les paires de nœuds qui vont communiquer, et pour quelle quantité de données. Par *étape*, on désigne la réalisation simultanée de cet ensemble de communications. Aucune nouvelle communication n'est entreprise avant la fin d'une étape.
4. Les données peuvent éventuellement être transmises en plusieurs fois, c'est-à-dire utiliser plusieurs étapes.

Une étape de communication est caractérisée par sa *durée*, définie comme la durée de la plus longue communication, et les paires de nœuds mises en jeu. On appelle *durée effective* de communication la somme des durées de toutes les

[1] câble supposé de gros débit

étapes. La mise en place d'une étape, c'est-à-dire l'établissement des communications entre les paires de nœuds, possède un coût, β, supposé indépendant du nombre de paires en question. On va chercher à minimiser la durée globale, définie comme la somme de la durée effective et du nombre d'étapes multiplié par β.

3 Modélisation du problème

Les données du problème sont donc les suivantes : n, d, D, β, et une matrice $Q = (q_{i,j} \in N)_{1 \leq i \leq n, 1 \leq j \leq n}$ qui correspond aux quantités de données qui doivent être transmises.

Faisons quelques remarques selon la comparaison de d à D :

- Si $d \geq D$: la durée globale de transmission est optimisée si chaque ordinateur transmet successivement ses données, car le backbone est sans cesse saturé. Ce n'est d'ailleurs pas un cas intéressant dans la pratique.
- Si $n \times d \leq D$: les n ordinateurs P_i pouvant éventuellement communiquer en même temps, le backbone ne constitue pas ici une contrainte. C'est un cas qui a déjà été étudié (par exemple dans [5], voir section 3.3).
- Si $1 < \frac{D}{d} < n$: le nombre de paires d'ordinateurs pouvant communiquer est alors limité par $\left\lfloor \frac{D}{d} \right\rfloor$. (On suppose que le débit ne peut pas être abaissé au cours du temps)

Ce qui précéde nous invite à se restreindre au cas où $1 < \frac{D}{d} \leq n$ et à ramener la contrainte 2 concernant les débits à la contrainte suivante : au plus $k = \left\lfloor \frac{D}{d} \right\rfloor$ communications peuvent avoir lieu lors d'une étape. Pour que la notion de débit n'intervienne plus, la matrice Q est remplacée par la matrice $A = (a_{i,j}) = (\frac{q_{i,j}}{d})_{1 \leq i \leq n, 1 \leq j \leq n}$. Cette dernière représente les temps de communication requis entre les ordinateurs.

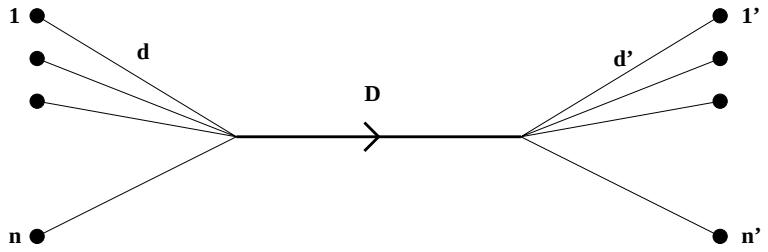


FIG. 1 – Le problème initial

Nous allons formuler ce problème, que nous appelerons KPBS(n, k, β, A), de deux façons différentes : la première utilise la notion de couplage dans un graphe biparti, c'est celle que nous allons utiliser par la suite. La seconde formulation, utilisée dans certains articles, notamment celui de *Bongiovanni et al.* [3], est matricielle.

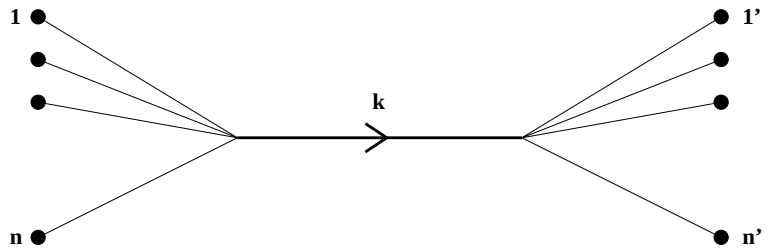


FIG. 2 – KPBS

3.1 Formulation en termes de couplages

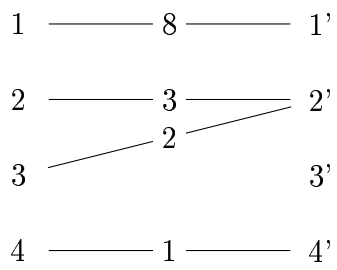
Le problème est modélisé par un graphe biparti G muni d'une fonction de poids entière w :

$$G = \Big(X = \{P_1, \ldots, P_n\}, Y = \{P'_1 \ldots P'_n\}, E = \{(P_i, P'_j) | a_{i,j} > 0\}\Big)$$

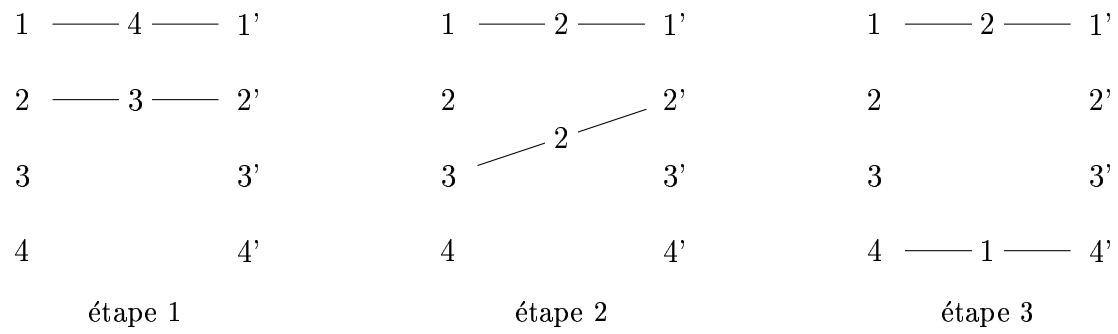
$$w\left((P_i, P'_j)\right) = a_{i,j}$$

On appelle *couplage pondéré* de G un couple (E', w') tel que E' soit un couplage de G et w' une fonction de poids vérifiant $w'(e) \leq w(e)$. Il est dit *valide* si $|E'| \leq k$. On définit la somme de deux couplages pondérés comme le couplage pondéré $(E_1 \cup E_2, w_1 + w_2)$. Pour un couplage pondéré $C = (E', w')$ on définit par extension $w(C) = max\ w'(e \in E')$. Un couplage pondéré valide C correspond à une étape de communication dont $w(C)$ est la durée (la notion de couplage garantit qu'un nœud n'utilise qu'un seul port à la fois, la validité constitue la contrainte du backbone, et la pondération permet de découper les paquets de données). On cherche t couplages C_i pondérés valides de somme égale à (E, w) tel que le coût $\sum_{i=1}^{t} w(C_i) + \beta \times t$ soit minimal.

Exemple :



Voici une solution pour $k = 2$:



Son coût est :

$$4 + 2 + 2 \ + \ 3 \times \beta = 8 + 3 \times \beta$$

On notera dans la suite $m(G) = |E(G)|$, $\Delta(G)$ le maximum des degrés du graphe G, et pour un sommet s, respectivement $\Delta(s)$ et $W(s)$ le degré et la somme des poids incidents à s. Enfin on appelle $W(G)$ le maximum des $W(s)$, et $P(G)$ la somme des poids de toutes les arêtes ($P(G) = \sum a_{i,j} = \sum_{e \in E} w(e)$).

Remarque: Si le graphe G est remplacé par un multigraphe, il est possible de spécifier qu'une communication entre deux nœuds doit être transmise en plusieures étapes.

3.2 Formulation matricielle

On appelle matrice de communication une matrice $B = (b_{i,j})_{1 \leq i \leq n, 1 \leq j \leq n}$ ayant au plus k élèments non nuls (contrainte du backbone), et au plus un élément non nul par ligne et par colonne (contrainte d'un seul port à la fois). On définit $w(B) = max\ b_{i,j}$. On cherche t matrices B_i de somme égale à la matrice $A = (a_{i,j})$ de sorte que le coût $\sum_{i=1}^{t} w(B_i) + \beta \times t$ soit minimal.

Le même exemple que précédemment :

$$\begin{pmatrix} 8 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

La même solution pour $k = 2$:

$$\begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

étape 1 étape 2 étape 3

3.3 Travaux précédents

Le cas où $k = n$ (le problème est alors appelé PBS :"Preemptive Bipartite Scheduling") est considéré par *Crescenzi et al.* [5] et *Afrati et al.* [1]. On trouve en particulier dans [5] des algorithmes d'approximation bornés par un facteur 2 de l'optimum. L'article [1] fournit un algorithme qui améliore ce facteur.

Plusieurs travaux (par exemple [6]) ont prouvé que PBS est NP-complet. Il en résulte que le problème KPBS est lui aussi NP-complet (puisque PBS est un cas particulier de KPBS). Par la suite, nous prouvons qu'il le reste si k est une constante.

Enfin, l'article de *Bongiovanni et al.* [3] propose un algorithme polynomial optimal de KPBS lorsque $\beta = 0$.

4 Contribution

4.1 NP-complétude du problème quand k est constant

Notre problème KBPS se formule comme suit :

Entrée : Un entier n, un graphe biparti G pondéré, un entier k
un entier β, un entier B.
Question : Trouver une suite de couplages pondérés valides $C_1, C_2, \ldots, C_t$ tel que
$G = \sum_{i=1}^{t} C_i$ et $\sum_{i=1}^{t} w(C_i) + t \times \beta \leq B$.

Théorème 1 *Si l'on fixe $k \geq 2$ et $\beta > 0$, le problème reste NP-complet.*

Remarques :
- Si $k = 1$, le problème se résout optimalement et polynomialement.
- En pratique β est une constante non nulle.

preuve du théorème 1 :

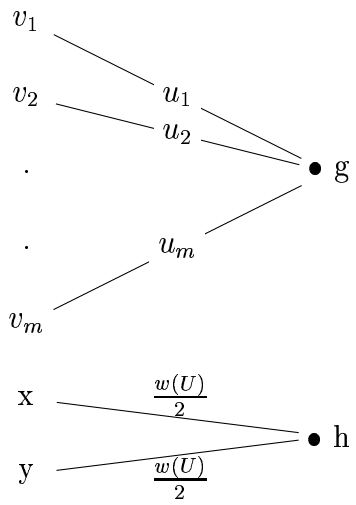
On fixe $k > 1$ et $\beta > 0$. Le problème KPBS est dans NP, pour montrer qu'il est NP-complet, on va effectuer une réduction à partir du problème NP-complet *bipartition* :

Entrée : Un ensemble d'entiers $U = \{u_1, u_2, \ldots, u_m\}$.
On note $w(U) = \sum_1^m u_i$.
Question : Trouver une partition (U_1, U_2) de U telle que $\sum_{u \in U_1} u = \sum_{u \in U_2} u$

On transforme une instance du problème bipartition en une instance de KBPS comme suit :

Soit $U = \{u_1, u_2, \ldots, u_m\}$ un ensemble d'entiers. On pose $n = m + 2$, $B = w(U) + \beta \times m$.

On définit G par :



La transformation est polynomiale, et il faut prouver qu'une instance de bipartition admet une solution ssi l'instance de KBPS correspondante en admet une.

($\Rightarrow$) Si l'instance de bipartition admet une solution, soient U_1 et U_2 les ensembles en question. On considére les couplages valides $\{(v_i, g)(x, h)\}_i$ pour $u_i \in U_1$ avec u_i pour poids sur chaque arête, et les couplages valides $\{(v_j, g)(y, h)\}_j$ pour $u_j \in U_2$ de poids sur chaque arête u_j. Il est clair que la somme de ces couplages est égale à G car $w(U_1) = w(U_2) = \frac{w(U)}{2}$. Le coût associé vaut exactement B.

($\Leftarrow$) Supposons que l'instance de KPBS admet une solution. On remarque d'abord que la durée effective est au moins égale à $w(U)$, et que le nombre d'étapes est au moins égal à m car les arêtes issues des v_i sont toutes adjacentes. Le coût étant inférieur ou égal à B, les deux inégalités précédentes sont donc des égalités, et aucune arête issue d'un v_i n'est découpée. La taille d'un couplage est au plus 2, et par ce qui précéde, chaque couplage C de la solution contient une arête issue d'un v_i et pour ce même i vérifie $w(C) \leq u_i$. On peut donc noter $(C_i)_{1 \leq i \leq m}$ les couplages de la solution. Il est alors nécessaire que dans chaque couplage C_i figure

aussi une arête issue de x ou y et portant poids u_i de l'autre arête du couplage, ceci afin d'épuiser les poids des deux arêtes issues de x et y.

On définit U_1 comme l'ensemble des u_i tel que v_i figure dans un couplage qui contient une arête issue de x, et U_2 par $U \backslash U_1$. On a bien $w(U_1) = \frac{w(U)}{2} = w(U_2)$.

•

4.2 Borne inférieure du coût

Proposition 1 *Soit G un multigraphe biparti pondéré.*

$$\eta(G) = \eta_d(G) + \beta \times \eta_e(G)$$

est une borne inférieure du coût d'une solution, où

$$\eta_d(G) = max\left(W(G), \left\lceil \frac{P(G)}{k} \right\rceil\right) \quad et \quad \eta_e(G) = max\left(\Delta(G), \left\lceil \frac{m(G)}{k} \right\rceil\right)$$

preuve:
$\eta_d(G)$ est une borne inférieure de la durée effective globale. Le premier terme du maximum vient du fait que toutes les arêtes issues d'un même sommet doivent figurer dans des couplages differents, le deuxième vient de la jugulation entraînée par le backbone : il s'agit du temps effectif requis si l'on arrive à le saturer en permanence.

$\eta_e(G)$ est une borne inférieure pour le nombre d'étapes. Elle s'obtient de la même manière. Le coût total est donc minimisé par

$$\eta(G) = \eta_d(G) + \beta \times \eta_e(G)$$

•

Etudions maintenant la qualité de ces bornes inférieures :

4.3 Atteignabilité de la borne inférieure η_e du nombre d'étapes

On donne une preuve existentielle puis constructive de l'atteignabilité de η_e. Remarquons qu'elles fonctionnent aussi dans le cas où G est un multigraphe, ce qui sera utile plus tard.

4.3.1 Cas où $\Delta(G) \leq \left\lceil \frac{m(G)}{k} \right\rceil$

Proposition 2 *Soit G un multigraphe biparti tel que $\Delta(G) \leq \left\lceil \frac{m(G)}{k} \right\rceil$. Alors G peut être décomposé en $\left\lceil \frac{m(G)}{k} \right\rceil$ couplages valides.*

La proposition 2 est une conséquence du théorème 2 (voir par exemple [2]) :

Théorème 2 (Folkman, Fulkerson , 1966) *Soit $G = (X, Y, E)$ un multigraphe biparti de degré maximum $\leq q$. Soit une suite d'entiers*

$$m_1 \geq m_2 \geq \ldots \geq m_q$$

avec

$$\sum_{i=1}^{q} m_i = m = |E|$$

On appelle m_j^* le nombre de m_i qui sont $\geq j$. S'il existe un coloriage des arêtes avec q couleurs $\alpha_1, \ldots, \alpha_q$ tel qu'il y ait exactement m_i arêtes de couleur α_i, pour tout i, alors, notant $m_G(A, B)$ le nombre d'arêtes du graphe G aux extremités dans A et B :

$$\forall (A, B) \subseteq (X, Y) \qquad m_G(A, B) \geq \sum_{j > |X-A|+|Y-B|} m_j^*$$

De plus, dans le cas où il existe k,

$$m_1 = \ldots = m_k \geq m_{k+1} = \ldots = m_q$$

il s'agit d'une condition suffisante.

preuve de la proposition 2 :

Soit $m = \sum \Delta(P_i)$ le nombre d'arêtes de G. On suppose $k|m$ quitte à rajouter éventuellement des arêtes à G sans changer l'inégalité $\Delta(G) \leq \frac{m(G)}{k}$. C'est toujours possible et cela ne change pas le résultat car la borne définie comme la partie entière supérieure de $\frac{m(G)}{k}$ n'a pas changée. On pose $q = \frac{m}{k}$. On a bien par hypothèse $\Delta(G) \leq q$. On pose

$$m_1 = m_2 = \ldots = m_{\frac{m}{k}} = k$$

c'est-à-dire qu'on colore des paquets de k arêtes, ce qui nous place dans le cas de la CNS du théorème 2. On a bien $\sum_1^q m_i = k \times \frac{m}{k} = m = |E|$ Pour les m_j^*, on a dans ce cas :

- si $j \leq k$ $m_j^* = \frac{m}{k}$
- si $j > k$ $m_j^* = 0$

Fixons $(A, B) \subseteq (X, Y)$. Il faut montrer que

$$m_G(A, B) \geq \sum_{j > |X-A|+|Y-B|} m_j^*$$

c'est-à-dire

$$m_G(A, B) \geq \frac{m}{k} \sum_{j=|X-A|+|Y-B|+1}^{k} 1$$

Or

$$|X - A| + |Y - B| + 1 = 2n + 1 - |A| - |B|$$

Si $2n+1-|A|-|B| \geq k+1$, c'est-à-dire $|A|+|B| \leq 2n-k$, le résultat est prouvé (on a toujours $m_G(A,B) \geq 0$). Sinon, si $|A| + |B| > 2n - k$, il faut montrer que

$$m_G(A,B) \geq \frac{m}{k} \times (k - 2n + |A| + |B|)$$

- Si $|A| + |B| = 2n$, le résultat est prouvé (A et B forment le graphe entier).
- Si $|A|+|B| = 2n-i$ alors $\frac{m}{k} \times (k-2n+|A|+|B|) = \frac{(k-i)m}{k}$ et par hypothèse $m_G(A,B) \geq m - i \times \Delta(G) \geq m - i \times \frac{m}{k} \geq \frac{(k-i)m}{k}$. D'où le résultat.

•

Voici maintenant une preuve constructive. On suppose toujours $k|m$, quitte à rajouter quelques arêtes. Voici un lemme, dont la preuve est inspirée d'une analogie matricielle de [3].

Lemme 1 *Soit G un multigraphe biparti. Si $\Delta(G) \leq \frac{m}{k}$, alors G admet un couplage de taille k qui sature les éventuels sommets de degrés $\frac{m}{k}$*

preuve:
On rajoute $n - k$ sommets à X et $n - k$ sommets à Y. On rajoute ensuite des arêtes pour obtenir un multigraphe $\frac{m}{k}$-régulier[2]. C'est possible car comme le degré sortant de X est égal à m, il faut donc rajouter $n \times \frac{m}{k} - m$ arêtes pour obtenir la $\frac{m}{k}$-régularité des sommets initiaux de X, ce qui vaut $(n - k) \times \frac{m}{k}$, c'est-à-dire exactement le nombre d'arêtes qu'il est possible et qu'il faut placer sur les $n - k$ nouveaux sommets de Y pour obtenir leur $\frac{m}{k}$-régularité. Le raisonnement est identique pour les sommets restants.

Ensuite, de ce graphe biparti régulier on peut extraire un couplage parfait [3]. Dans ce couplage, $2 \times (n - k)$ arêtes saturent les sommets rajoutés (car il n'y a pas d'arête entre deux des nouveaux sommets). Les autres arêtes appartiennent au graphe initial. Il y en a $2 \times n - k - 2 \times (n - k) = k$. Elles forment le couplage valide cherché. Il reste à voir que les sommets de degré $\frac{m}{k}$ sont couverts. On n'a pas rajouté d'arête à un tel sommet, donc il est couvert dans le couplage par une arête du graphe initial.

•

[2]Un graphe ou un multigraphe est dit *régulier* lorsque tous ses sommets ont le même degré.

[3]Tout graphe ou multigraphe biparti régulier admet un couplage parfait (voir par exemple [2])

Remarque: Trouver un couplage maximal dans ce multigraphe biparti se fait en $\mathcal{O}(n^2 \times m)$ (voir [4]), c'est la complexité pour trouver le couplage du lemme 1.

preuve constructive de la proposition 2 :
Deux cas :

- $\Delta(G) = \frac{m}{k}$, alors on retire de G un couplage de taille k comme dans la proposition précédente pour obtenir un graphe G'. On a $\frac{m(G')}{k} = \frac{m(G)}{k} - 1$, et comme il y a au plus k sommets de degré $\frac{m}{k}$, le degré maximal de G' a baissé, donc G' vérifie $\Delta(G') \leq \frac{m(G')}{k}$.
- $\Delta(G) < \frac{m}{k}$, on retire de G un couplage de taille k du lemme 1 pour obtenir un graphe G' vérifiant $\Delta(G') \leq \frac{m(G')}{k}$.

On peut ainsi conclure par récurrence sur $\frac{m}{k}$, l'initialisation se faisant sans difficulté.

•

Remarque: La décomposition de la proposition 2 s'effectue donc avec la complexité $\mathcal{O}(n^2 \times m^2)$.

4.3.2 Cas où $\Delta(G) > \frac{m}{k}$

On peut s'arranger pour rajouter des arêtes aux sommets de plus faibles degrés, obtenir un multigraphe G' vérifiant $\Delta(G') = \frac{m(G')}{k}$ et appliquer le résultat précédent, puisque la borne de G' reste celle de G.

Remarque: Le rajout des arêtes se fait en $\mathcal{O}(n^2)$.

Nous pouvons aussi faire le raisonnement suivant :

G se décompose en $\Delta(G)$ couplages (voir [2]). Si les $\Delta(G)$ couplages sont de taille $\leq k$, c'est terminé. Sinon, il existe un couplage de taille $> k$, alors on prend k arêtes de ce couplage, de sorte que tous les sommets de degrés maximaux soient couverts. C'est possible car il y en a au plus k et nécessairement chaque couplage les sature. Cela donne un graphe G' qui vérifie $\Delta(G') = \Delta(G) - 1$, et $\frac{m(G')}{k} = \frac{m(G)}{k} - 1$. Donc on peut poursuivre par récurrence sur $\Delta(G)$. (Initialisation : $\Delta(G) = 1$ implique que toutes les arêtes sont indépendantes deux à deux, et qu'il y en a au plus k.)

4.4 La borne inférieure η_d de la durée effective est atteignable

Ceci est prouvé dans [3] dans une version matricielle. En fait il y a une analogie entre la durée effective et le nombre d'étapes. En effet, si l'on considère un graphe

biparti dont tous les poids valent 1, optimiser la durée équivaut à optimiser le nombre d'étapes. Donc la preuve de l'atteignabilité de η_e avait déjà été faite dans le cas d'un *graphe* biparti.

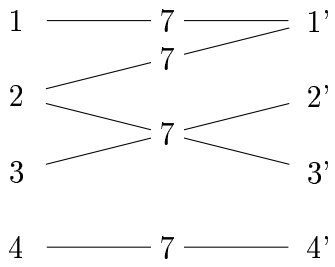
4.5 Exemples de cas où la borne η est atteignable ou non

Nous avons vu que chaque terme (η_d ou η_e) de la borne η est atteignable, qu'en est-il des deux ensembles ?

4.5.1 Cas où la borne η est atteignable

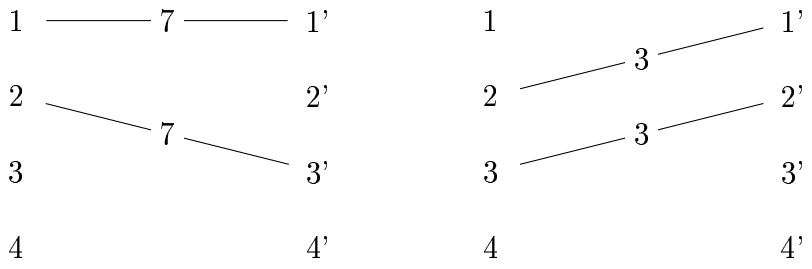
Quand toutes les arêtes portent le même poids α, et que $k|m$, toute décomposition qui atteint la borne inférieure du nombre d'étapes est optimale, car elle atteint aussi la borne inférieure de la durée. Et réciproquement. Cela provient du fait que dans ce cas on a $\eta_d = \alpha \times \eta_e$.

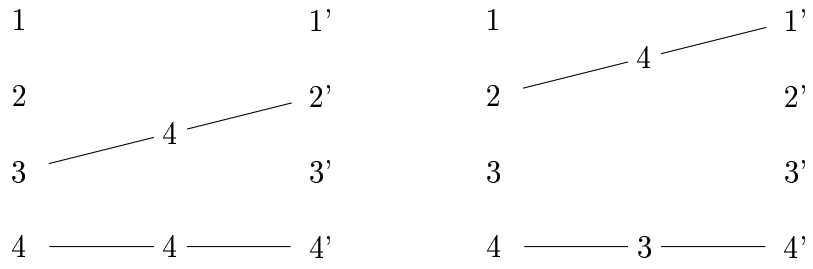
Ce n'est pas nécessairement vrai si k ne divise pas m : Considérons le graphe suivant avec $k = 2$ et $\beta = 1$:



$$\eta_e = 3 \qquad \eta_d = 18$$

Une décomposition en $\eta_e = 3$ étapes posséde une durée effective de 21, donc un coût de 24. Compte tenu du choix $\beta = 1$, la décomposition suivante en 4 étapes, de durée 18 et de coût 22 est meilleure :





Autres graphes où la borne η est atteignable :

Supposons que le graphe biparti G soit la réunion des graphes bipartis $(G_i)_{1 \leq i \leq s}$ vérifiant :

- chaque G_i est uniforme en poids : chaque arête porte le poids α_i
- k divise $m_i = m(G_i)$
- $\Delta(G_i) \leq \frac{m_i}{k}$

Alors il est facile de voir que si l'on décompose chaque G_i optimalement, la réunion des décompositions est optimale pour G. En effet pour chaque G_i, la borne inférieure est atteinte, et $\eta(G) = \sum \eta(G_i)$.

On a un résultat analogue si l'on remplace les deux dernières conditions par les deux suivantes :

- il existe un sommet s tel que pour tout i, $\Delta(s) = \Delta(G_i)$
- $\Delta(G_i) \geq \frac{m_i}{k}$

car $\Delta(G) = \sum \Delta(G_i)$.

4.5.2 Cas où la borne η n'est pas atteignable

Le graphe fourni ci-avant comme contre-exemple dans le cas où k ne divise pas m convient. En effet on a vu que que la borne n'est pas atteignable en $\eta_e = 3$ étapes. Elle ne pourra donc pas l'être non plus avec plus d'étapes (si $\beta > 0$).

4.6 Algorithme pseudo-polynomial de facteur constant

Remarque: A quelle distance sommes nous de la borne inférieure η dans le cas où l'on n'exploite pas le parallélisme ? Supposons qu'on sérialise toutes les communications, c'est à dire qu'elles soient effectuées les unes à la suite des autres. A chaque $a_{i,j}$ correspond une communication, de coût $a_{i,j} + \beta$. Le coût total est donc

$$\sum W(P_i) + \beta \times \sum \Delta(P_i) = k \times \left(\frac{P(G)}{k} + \beta \times \frac{m(G)}{k} \right) \leq k \times \eta$$

Cet algorithme se trouve donc dans le pire des cas à un facteur k de la borne inférieure, donc de l'optimal.

Voici un algorithme qui approxime le problème avec un facteur constant : Intuitivement, pour que la durée effective totale ne soit pas trop élevée, il est souhaitable que dans chaque étape toutes les communications aient la même durée. Pour pouvoir évaluer le nombre d'étapes, nous allons de plus faire en sorte que toutes les étapes aient la même durée. Supposons provisoirement les poids multiples d'une valeur constante α que nous définirons plus loin et considèrons l'algorithme qui décompose chaque arête en arêtes de poids α (procédé inspiré de [1]), ce qui donne un multigraphe H, puis en vertu de ce qui précéde, trouve le nombre optimal de couplages valides de H :

Algorithme

Entrée : un graphe biparti G.
Sortie : une décomposition en couplages pondérés valides.

1. Décomposer les arêtes en arêtes de poids multiples de α, ce qui donne un multigraphe H.
2. Décomposer H en un nombre optimal de couplages.
3. Un couplage représente une étape de durée α.

Complexité de l'algorithme

Elle est en $\mathcal{O}\left(n(H)^2 \times m(H)^2\right) = \mathcal{O}\left(n(G)^2 \times m(G)^2 \times W(G)^2\right)$. Dépendant des poids de G, elle est pseudo-polynomiale.

Evaluation de la solution

On note $\eta_e(H)$ la borne inférieure (atteinte) du nombre d'étapes pour H et $\eta_d(G)$ la borne inférieure en durée effective pour G. On remarque en revenant aux définitions que $\alpha \times \eta_e(H) \leq \eta_d(G) + \alpha$, d'où :

$$\begin{aligned} coût &= \alpha \times \eta_e(H) + \beta \times \eta_e(H) \\ &\leq \frac{\beta}{\alpha} \times \eta_d(G) + \eta_d(G) + \alpha + \beta \\ &\leq \left(\frac{\beta}{\alpha} + 1\right) \times \eta(G) + \alpha + \beta \end{aligned}$$

En prenant $\alpha = \beta$, on obtient un facteur de 2.

Que se passe-t-il quand dans G les poids des arêtes ne sont pas multiples de α ? On appelle G' le graphe issu de G dont toutes les arêtes ont été arrondies au multiple supérieur. On essaye de comparer $\eta(G)$ à $\eta(G')$. Seul η_d change car les

deux graphes ont le même nombre d'arêtes. Or

$$\begin{aligned}
\eta_d(G') &= max\left(W(G'), \left\lceil \frac{P(G')}{k} \right\rceil\right) \\
&\leq max\left(W(G) + (\alpha-1)\Delta(G), \left\lceil \frac{P(G) + (\alpha-1)m(G)}{k} \right\rceil\right) \\
&\leq \eta_d(G) + (\alpha-1) \times \eta_e(G) \\
&\leq \eta(G)
\end{aligned}$$

en prenant toujours $\alpha = \beta$. D'où $\eta(G') \leq 2 \times \eta(G)$, ce qui donne au total un facteur de 4 dans le pire des cas. C'est nettement meilleur si les poids sont grands par rapport à α.

Réutilisons cet algorithme pour obtenir un algorithme polynomial :

4.7 Algorithme polynomial de facteur constant

Il apparaît pertinent de choisir l'algorithme en fonction de la comparaison entre la valeur de β et la durée moyenne d'une étape. Moyennant cette distinction, on obtient un algorithme polynomial en utilisant deux algorithmes différents selon le cas.

Intuitivement, $\frac{P(G)}{k \times m(G)}$ représente la durée moyenne d'une étape. A la place de $m(G)$, on utilise un polynôme d'ordre de grandeur n^2 pour que cela fonctionne.

Soit f un polynôme en n.

Supposons $P(G) \leq \beta \times k \times f(n)$

Dans ce cas, les poids étant bornés, l'algorithme de la section précédente devient polynomial et peut convenir.

Supposons $P(G) > \beta \times k \times f(n)$

Intuitivement, il s'agit du cas où le temps β pour mettre en place une étape de communication est petit devant le temps effectif de chaque étape. Dans ce cas, on va chercher un algorithme qui fournit une durée effective totale proche de la borne.

L'article [3] fournit un algorithme polynomial optimal pour la durée effective, avec dans le pire des cas un *nombre d'étapes* inférieur à $n^2 + n + 1$. Choisissons donc $f(n) = n^2 + n + 1$. Pour cet algorithme, on a alors :

$$\begin{aligned}
coût &\leq \eta_d(G) + \beta \times f(n) \\
&\leq \eta_d(G) + \frac{P(G)}{k} \\
&\leq 2 \times \eta(G)
\end{aligned}$$

Il approxime donc dans ce cas le problème à un facteur 2 de l'optimal.

On dispose donc d'un algorithme d'approximation polynomial pour KPBS.

4.8 Heuristiques

Voici deux heuristiques qui se comportent bien en pratique (voir les courbes ci-après) :

Heuristique sur les poids

Entrée : un graphe biparti G.
Sortie : une décomposition en couplages pondérés valides.

1. Prendre un couplage maximal
2. Le transformer en couplage valide maximal, en conservant les k arêtes de plus gros poids
3. Le soustraire de G
4. Recommencer jusqu'à épuisement des arêtes.

Heuristique sur les degrés

Entrée : un graphe biparti G.
Sortie : une décomposition en couplages pondérés valides.

1. Prendre un couplage maximal
2. Conserver uniquement les k arêtes adjacentes aux sommets de plus gros degré (A chaque arête est associé le maximum des degrés de ses deux sommets, et on prend les arêtes qui ont les plus grands maximums)
3. Les soustraire de G
4. Recommencer jusqu'à épuisement des arêtes.

Ces heuristiques sont polynomiales car à chaque étape une arête du graphe est supprimée (méthode utilisée dans [3],[1] et [5] pour rendre les algorithmes polynomiaux). Par ailleurs, de cette façon toutes les communications d'une étape ont la même durée.

Les simulations ont été effectuées pour n et k donnés sur un échantillon de 100000 graphes aléatoires (sauf pour la figure 4.8 où les premiers points ont été obtenus à partir de 30000 graphes). On a distingué entre des graphes de gros

poids (les arêtes ont alors un poids compris entre 1 et 100000) et des graphes de petits poids (les arêtes ont alors un poids compris entre 1 et 20), sachant que $\beta = 1$. On évalue la solution d'une heuristique grâce au rapport entre le coût de la solution produite et la borne inférieur η d'une solution. Les courbes représentent pour n donné la moyenne ou le maximum des quotients obtenus, en fonction de k.

Pour les tests effectués, le maximum est toujours inférieur à 2.5, voire 1.8 s'il s'agit de gros poids, et la moyenne est toujours inférieure à 2, voire 1.3 s'il s'agit de gros poids.

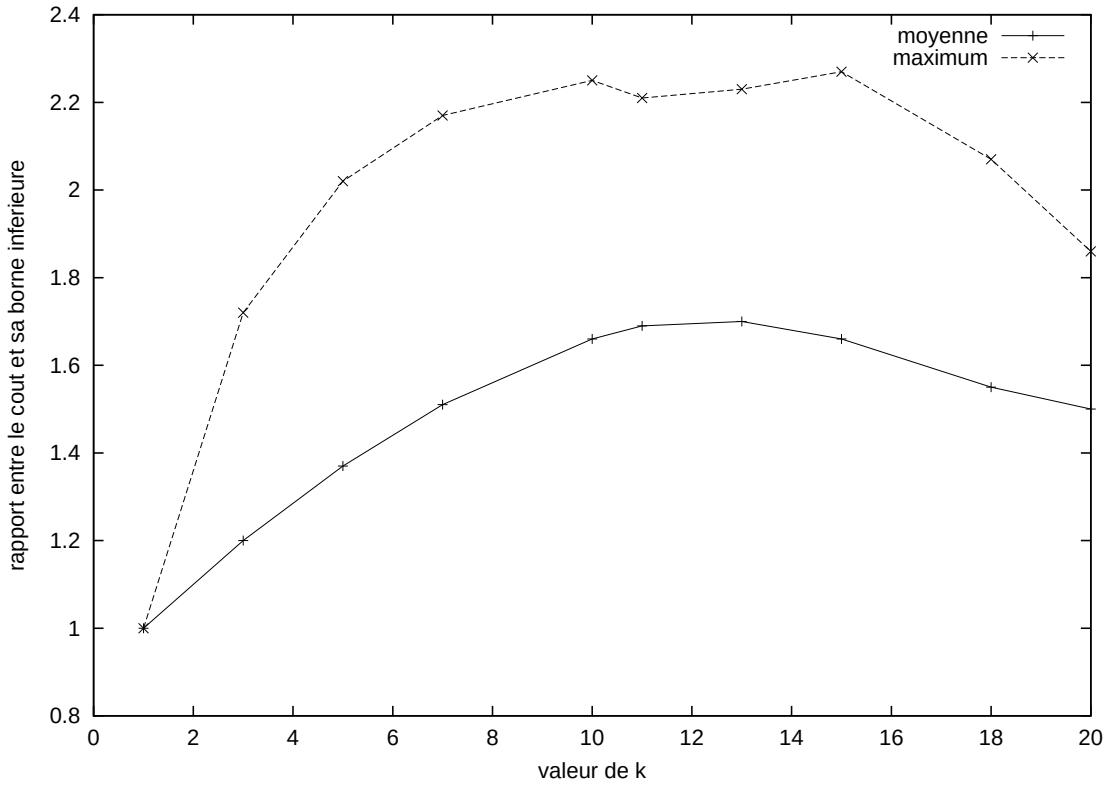


FIG. 3 – Heuristique sur les degrés. $n = 20$. Petits poids. Simulation sur 100000 graphes par points.

5 Conclusion

Nous avons donc étudié un cas particulier de redistribution de données, KPBS. Nous avons prouvé sa NP-complétude à k constant, étudié une borne inférieure du coût d'une solution, et exhibé un algorithme d'approximation polynomial. Fi-

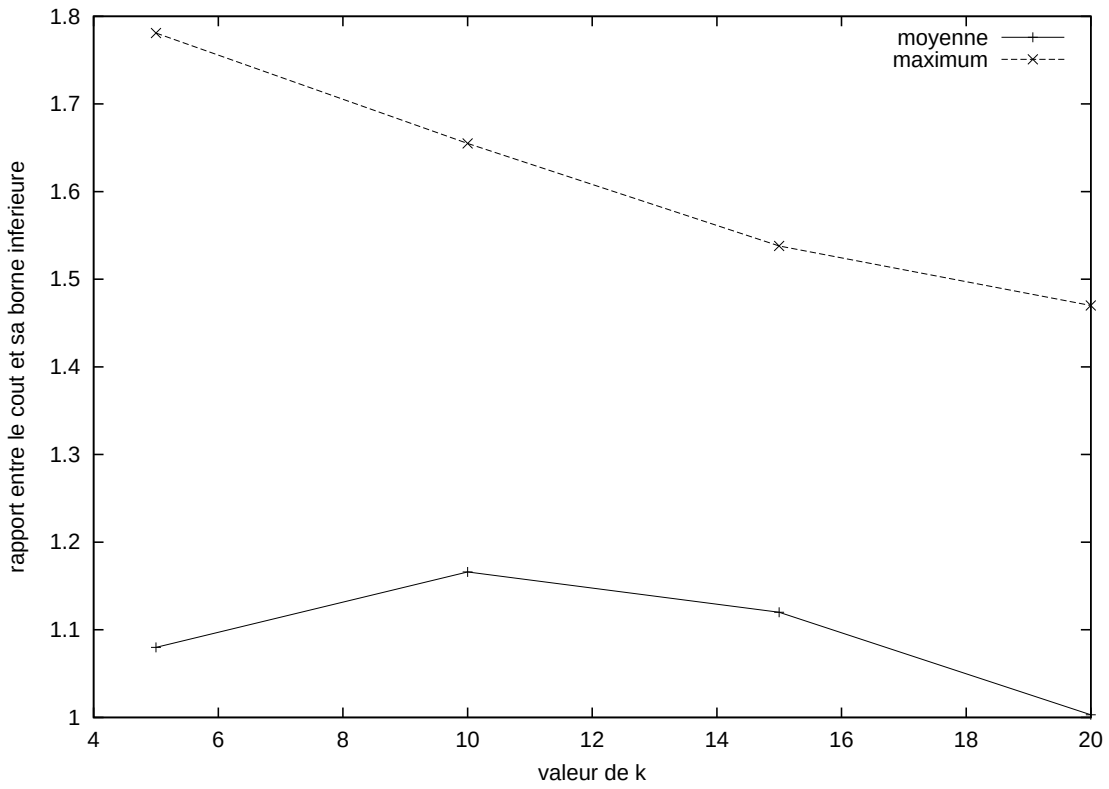


FIG. 4 – Heuristique sur les poids. $n = 20$. Gros poids. Simulation sur 100000 graphes par points.

nalement, nous avons implanté deux heuristiques qui approchent convenablement la solution.

Il reste à expérimenter la redistribution dans la réalité, sachant que l'hypothèse de débits constants ne s'applique pas forcément si d'autres personnes utilisent le réseau en même temps. Nous avons implanté avec la librairie MPI[4] un programme qui prend en entrée une description des étapes pour simuler une redistribution de données, par exemple entre les grappes d'ordinateurs des laboratoires INRIA de Grenoble et de Nice, en passant par les lignes du réseau VTHD[5]. Il reste encore à l'exploiter, et en particulier à mesurer la véritable valeur de β.

[4]Message Passing Interface
[5]Vraiment Très Haut Débit

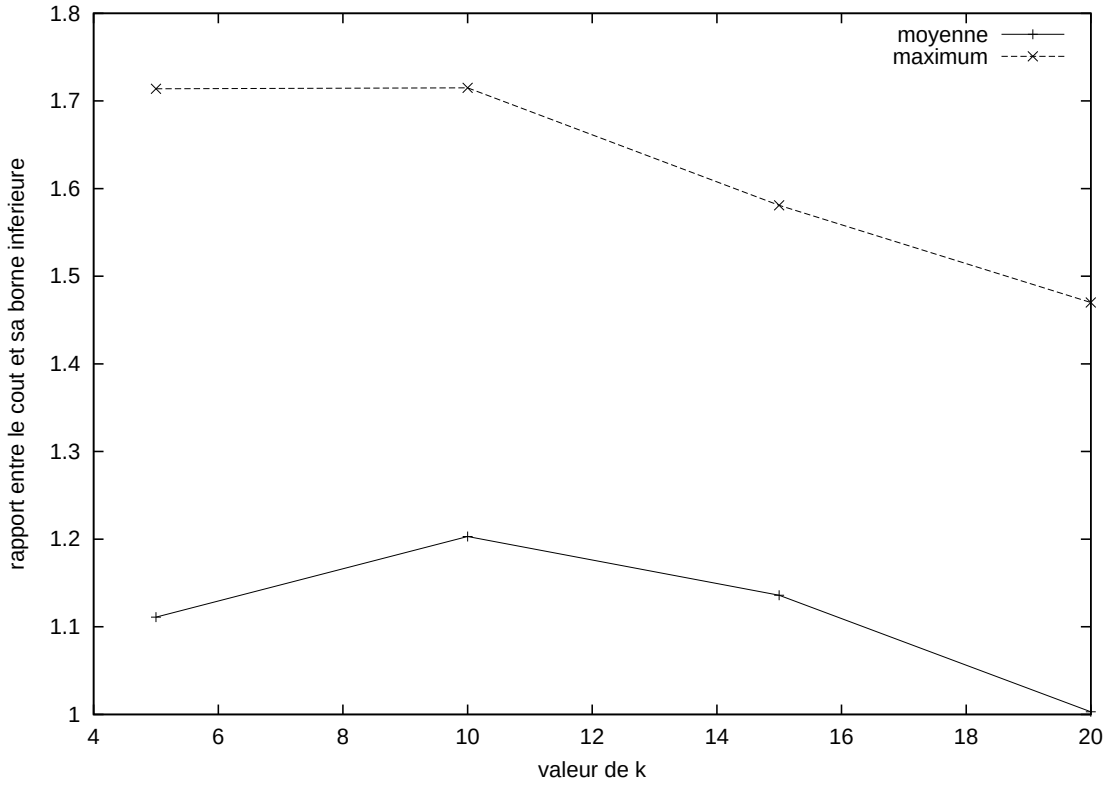


FIG. 5 – Heuristique sur les degrés. $n = 20$. Gros poids. Simulation sur 100000 graphes par points.

Références

[1] F. Afrati, T. Aslanidis, E. Bampis, and I. Milis. Scheduling in switching networks with set-up delays. In *AlgoTel 2002*, Mèze, France, May 2002.

[2] C. Berge. *Graphs*. North-Holland, 1985.

[3] G. Bongiovanni, D. Coppersmith, and C. K. Wong. An Optimum Time Slot Assignment Algorithm for an SS/TDMA System with Variable Number of Transponders. *IEEE Transactions on Communications*, 29(5) :721–726, 1981.

[4] Th. Cormen, C. Leiserson, and R. Rivest. *Introduction à l'algorithmique*. Dunod, 1994.

[5] P. Crescenzi, D. Xiaotie, and C. H. Papadimitriou. On Approximating a Scheduling Problem. *Journal of Combinatorial Optimization*, 5 :287–297, 2001.

[6] S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problem. *SIAM J. Comput.*, 5 :691–703, 1976.

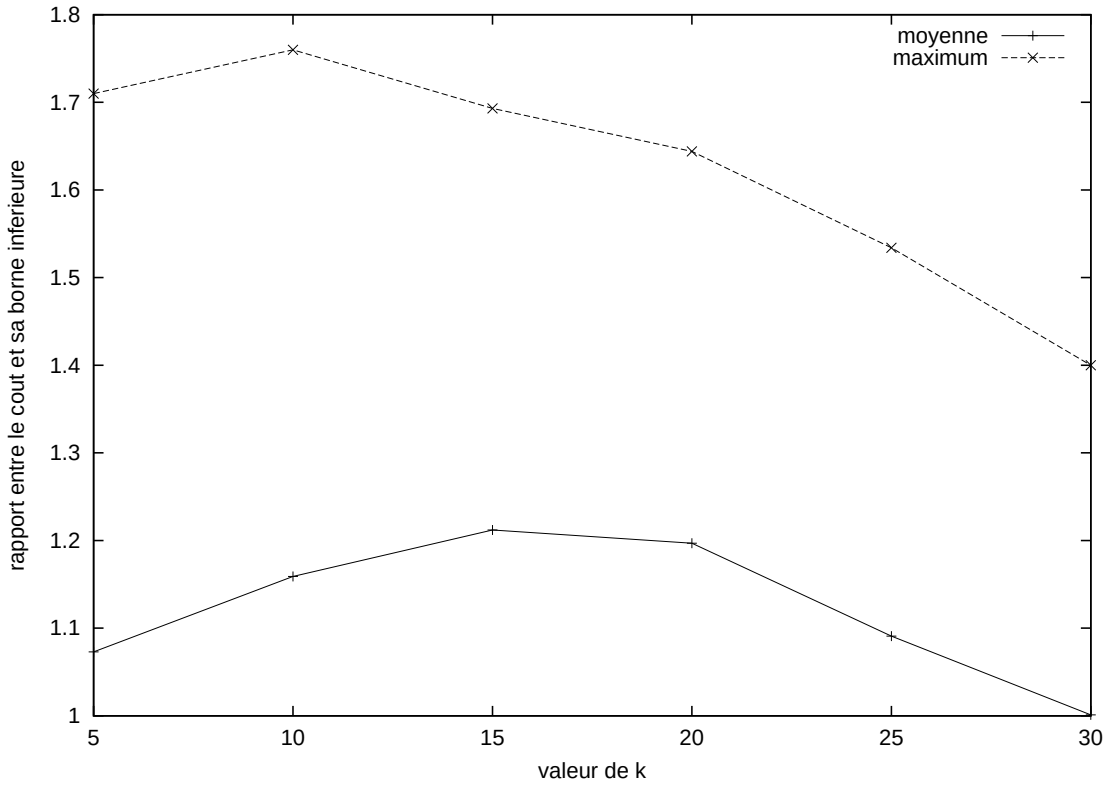


FIG. 6 – Heuristique sur les degrés. $n = 30$. Gros poids. Simulation sur plus de 30000 graphes par points.