Calcul du n-ème entier de Hamming

Algorithme polyédral

Michel Quercia
querciaOcal.enst.fr
Août 2000

1 Introduction

Les problèmes de Hamming sont les suivants :

> *Un entier de Hamming est un entier naturel dont les seuls diviseurs premiers sont* 2, 3 *et* 5.
> *(1) Écrire un programme imprimant dans l'ordre les* n *premiers entiers de Hamming.*
> *(2) Écrire un programme calculant le* n*-ème entier de Hamming sans calculer les précédents.*

Le problème (1) est présenté comme un exercice de programmation et comme une illustration du calcul paresseux. Il a été introduit par Dijkstra dans [3] pp. 129-134). Le problème (2) est apparu spontanément sur la liste de discussion `ups-info` après la présentation de divers algorithmes résolvant (1). On trouvera dans [4] une description des algorithmes résolvant (1) ou (2), arrêtée en Juin 1999. A cette époque, le meilleur algorithme connu résolvant (2) était la *méthode volumétrique* avec une complexité temporelle $n^{1/3}\ln(n)^2$. Le présent rapport fait suite à ce document et présente un nouvel algorithme, la *méthode polyédrale*, résolvant (2) en un temps expérimentalement proportionnel à $\ln(n)^3$.

A titre de comparaison, sur un PC-Linux équipé d'un processeur AMD Athlon-550 avec 128 Mo de mémoire, l'aglorithme volumétrique de 1999 trouve le 10^{12}-ème entier de Hamming en 54.43 secondes tandis que l'algorithme polyédral le trouve en 1.67 secondes. Dans les deux cas, la bibliothèque multiprécision utilisée est `BigNum` interfacée à Objective Caml. L'algorithme polyédral peut aussi utiliser la bibliothèque `Numerix` en version assembleur, le temps de calcul du 10^{12}-ème entier de Hamming passe alors à 0.53 secondes et le calcul de l'entier de Hamming de rang 10^{100} est effectué en moins de cinq minutes alors qu'il était inenvisageable avec l'algorithme volumétrique de 1999 (temps de calcul estimé à 2×10^{26} années au bas mot).

Ce présent rapport et le code `Ocaml` du programme polyédral sont disponibles sur le serveur de l'Inria à l'adresse :

`http://pauillac.inria.fr/~quercia/papers/hamming-new.ps.gz` (document postscript seul)
`http://pauillac.inria.fr/~quercia/papers/hamming-new.tgz` (sources du programme)

2 Description sommaire de l'algorithme polyédral

Cet algorithme reprend l'idée qui était à l'origine de l'algorithme volumétrique. Pour $a, b, c > 0$ et $N \in \mathbf{N}^*$ on note :

$$V(a, b, c, N) = \operatorname{card}\{(x, y, z) \in \mathbf{N}^3 \ / \ ax + by + cz \leqslant N\}.$$

$V(a, b, c, N)$ est le *volume discret* du tétraèdre de $\mathbf{R}^3$ limité par les plans de coordonnées et le plan d'équation $ax + by + cz = N$. Si $h_n = 2^{x_n}3^{y_n}5^{z_n}$ est le n-ème entier de Hamming, avec la convention $h_0 = 1$ alors on a :

$$V(\ln 2, \ln 3, \ln 5, N) \geqslant n + 1 \iff N \geqslant \ln(h_n).$$

Dans un premier temps on remplace les nombres $\ln(2), \ln(3)$ et $\ln(5)$ par des approximations rationnelles ayant le même dénominateur, soit $(\ln(2), \ln(3), \ln(5)) \approx (a, b, c)/d$ où a, b, c, d sont entiers de l'ordre de $1000n^{2/3}$ et on introduit la *fonction génératrice* $\Phi_{a,b,c,k} : \mathbf{C}^3 \longrightarrow \mathbf{C}$ telle que :

$$\Phi_{a,b,c,k}(x,y,z) = \sum_{\substack{p,q,r\in\mathbf{N},\\ ap+bq+cr\leqslant k}} x^p y^q z^r.$$

Barvinok et Pommersheim montrent dans [2] comment représenter $\Phi_{a,b,c,k}$ par une somme de fractions rationnelles dont le nombre de termes est $O(t^{\ln(3)/\ln(3/2)})$ avec $t = \ln(a)+\ln(b)+\ln(c)+\ln(k)$. Chaque fraction étant évaluable en temps constant, ceci permet de calculer $V(a,b,c,k) = \Phi_{a,b,c,k}(1,1,1)$ en temps $O(t^{\ln(3)/\ln(3/2)}) = O(\ln(n)^{\ln(3)/\ln(3/2)})$ pour k variable, et on peut alors déterminer en $O(\ln n)$ approximations successives deux entiers k', k'' tels que :

$$V(a,b,c,k') = n \quad \text{et} \quad V(a,b,c,k'') = n+1.$$

Le tétraèdre tronqué limité par les plans de coordonnées et les plans d'équation $ax+by+cz = k'$ et $ax+by+cz = k''$ contient donc un unique point (u,v,w) à coordonnées entières que l'on détermine par :

$$\begin{aligned}
u &= \frac{\partial\Phi_{a,b,c,k''}}{\partial x}(1,1,1) - \frac{\partial\Phi_{a,b,c,k'}}{\partial x}(1,1,1),\\
v &= \frac{\partial\Phi_{a,b,c,k''}}{\partial y}(1,1,1) - \frac{\partial\Phi_{a,b,c,k'}}{\partial y}(1,1,1),\\
w &= \frac{\partial\Phi_{a,b,c,k''}}{\partial z}(1,1,1) - \frac{\partial\Phi_{a,b,c,k'}}{\partial z}(1,1,1).
\end{aligned}$$

Connaissant u,v,w on calcule à l'aide d'encadrements rationnels de $\ln(2),\ln(3),\ln(5)$ deux entiers k^+ et k^- tels que les tétraèdres limités par les trois plans de coordonnées et les plans d'équations :

$$ax+by+cz = k^+ \quad \text{et} \quad ax+by+cz = k^-$$

encadrent le tétraèdre de Hamming limité par les trois plans de coordonnées et le plan d'équation :

$$x\ln(2) + y\ln(3) + z\ln(5) = u\ln(2) + v\ln(3) + w\ln(5),$$

puis on vérifie que :

$$V(a,b,c,k^-) = n \quad \text{et} \quad V(a,b,c,k^+) = n+1.$$

Si c'est le cas alors on a ainsi prouvé que le tétraèdre de Hamming précédent contient exactement $n+1$ points à coordonnées entières et que (u,v,w) est le dernier de ces points pour l'ordre induit par la forme linéaire $(x,y,z) \longmapsto x\ln(2)+y\ln(3)+z\ln(5)$ on a donc trouvé $h_n = 2^u 3^v 5^w$. Si $V(a,b,c,k^-) < n$ ou $V(a,b,c,k^+) > n+1$ alors l'approximation $(\ln(2),\ln(3),\ln(5)) \approx (a,b,c)/d$ était insuffisante et l'on recommence tous les calculs en augmentant la précision sur le triplet $(\ln(2),\ln(3),\ln(5))$ (on observe un cas d'échec de cette phase de validation entre autres avec $n = 10^6$).

En pratique on choisit a,b,c,d de sorte que $\max(|a/d-\ln(2)|, |b/d-\ln(3)|, |c/d-\ln(5)|)$ soit de l'ordre de $1/1000n$ (le coefficient 1000 a été déterminé expérimentalement pour limiter les cas d'échec) ce qui entraîne que la mesure de Lebesgue de la partie de $\mathbf{R}^3$ comprise entre les tétraèdres de paramètres a,b,c,k^+ et a,b,c,k^- soit de l'ordre de $(k^+-k^-)/1000$. Ceci rend plausible, mais non certain, l'hypothèse que cette région ne contient pas d'autre point à coordonnées entières que (u,v,w) et donc que l'approximation choisie soit suffisante.

Le temps de calcul est donc théoriquement de l'ordre de $\ln(n)^{1+\ln(3)/\ln(3/2)}$ en supposant que l'on dispose d'une approximation initiale de $(\ln(2),\ln(3),\ln(5))$ suffisante. On parvient à gagner un facteur $\ln(n)$ en effectuant la recherche de k' et k'' à l'aide d'une expression approchée de $V(a,b,c,k)$ calculable en temps constant. Lorsqu'on a trouvé k tel que $V(a,b,c,k) \approx n+1$ on poursuit les approximations sucessives avec la formule exacte $V(a,b,c,k) = \Phi_{a,b,c,k}(1,1,1)$ dont le temps de calcul est $\ln(n)^{\ln(3)/\ln(3/2)}$. La formule approchée utilisée est si précise qu'il ne reste plus, expérimentalement, que quelques itérations à effectuer sur la formule exacte d'où le temps total en $\ln(n)^{\ln(3)/\ln(3/2)} \approx \ln(n)^{2.71}$. En pratique on observe plutôt un temps de calcul de l'ordre de $\ln(n)^3$ mais une estimation précise de cet exposant n'est pas aisée.

Voici un exemple d'exécution pour la recherche du 10^{12}-ème nombre de Hamming :

```
~/src/caml/hamming-new$ ./hamming -e slong -d -n 10^10
    0.01     0.01 Entiers   : Slong
    0.01    -0.00 Réduction : LLL
    0.01    -0.00 2^44*ln(2) = 12193974156572 + 1
    0.01    -0.00 2^44*ln(3) = 19326991772931 + 1
    0.01    -0.00 2^44*ln(5) = 28313531182477 + 1
    0.01    -0.00 Réduction a = 173817877
    0.01    -0.00           b = 275494817
    0.01    -0.00           c = 403592612
    0.37     0.36 Barvinok -> 796 cônes
    0.37     0.00 Approx k = 1050592051267 v ~ 10012181829
    0.37     0.00 Approx k = 1050165792991 v ~ 10000004941
    0.37     0.00 Approx k = 1050165619994 v ~ 10000000001
    0.39     0.02 Exact  k = 1050165619994 v = 9999999997
    0.40     0.01 Exact  k = 1050165620135 v = 10000000001
    0.41     0.01 Exact  k = 1050165620064 v = 9999999999
    0.43     0.02 Exact  k = 1050165620099 v = 10000000000
    0.50     0.07 Dérivée -> x = 4210 y = 98 z = 722
    0.50     0.00 Validation -> OK
```

La première colonne donne le temps total en secondes écoulé depuis le début du programme, la deuxième colonne le temps de la phase en cours décrite dans la troisième colonne.

Les deux premières lignes rappellent quels modules sont utilisés par le programme, ici la bibliothèque `Numerix` en version assembleur (`Slong`) pour les calculs multiprécision et l'algorithme `LLL` pour la réduction des réseaux. On calcule ensuite des encadrements de $\ln(2), \ln(3), \ln(5)$, il faut lire :

$$12193974156572 < 2^{44}\ln(2) < 12193974156572 + 1$$

et de même pour les autres logarithmes. L'étape de réduction déduit de ces encadrements les paramètres a, b, c de l'ordre de $1000n^{2/3}$ qui vont servir pour les calculs suivants. Cette étape n'est pas indispensable, on pourrait travailler avec $a = 12193974156572$, $b = 19326991772931$ et $c = 28313531182477$ mais ces nombres sont inutilement grands et conduiraient à des calculs plus lents dans les étapes suivantes. Les valeurs a, b, c obtenues de l'étape de réduction fournissent une approximation de la direction engendrée par $(\ln(2), \ln(3), \ln(5))$ sensiblement équivalente à celle qui serait déduite des encadrements initiaux :

$$\frac{19326991772931}{12193974156572} - \frac{\ln(3)}{\ln(2)} \approx 1.2 \times 10^{-13} \qquad \frac{275494817}{173817877} - \frac{\ln(3)}{\ln(2)} \approx 2.2 \times 10^{-13}$$
$$\frac{28313531182477}{12193974156572} - \frac{\ln(5)}{\ln(2)} \approx 1.7 \times 10^{-13} \qquad \frac{403592612}{173817877} - \frac{\ln(5)}{\ln(2)} \approx 1.4 \times 10^{-13}$$

L'algorithme de Barvinok appliqué aux nombres a, b, c donne 796 cônes c'est-à-dire que la fraction rationnelle $\Phi_{a,b,c,k}$ contient 796 termes (seuls les numérateurs de ces termes dépendent de k, ils seront déterminés ultérieurement, voir la section 6). Le temps d'exécution de cette phase est 0.36 secondes, c'est la partie prépondérante du temps total d'exécution du programme.

Ensuite, le programme recherche k tel que $V(a, b, c, k) \approx n+1$ par itération de la méthode de Newton sur une formule approchée. On trouve $k = 1050165619994$ en trois itérations et on continue les calculs avec la formule exacte calculée à l'étape Barvinok. En quatre itérations on obtient $k' = 1050165620099$ et $k'' = 1050165620135$ ce qui permet de calculer u, v, w (notés ici x, y, z) par dérivation. Il ne reste plus qu'à calculer k^+ et k^- et vérifier que l'on a bien $V(a, b, c, k-) = n$ et $V(a, b, c, k^+) = n + 1$, c'est l'objet de la dernière phase (ligne Validation).

3 Calcul approché de $\ln(2)$, $\ln(3)$ et $\ln(5)$

On utilise le développement en série entière :

$$2\operatorname{argth}(x) = \ln\Big(\frac{1+x}{1-x}\Big) = 2\sum_{k=0}^{\infty}\frac{x^{2k+1}}{2k+1}$$

où x est un réel choisi suffisamment proche de zéro pour que la convergence de cette série soit rapide.

En étudiant les 1000 premiers entiers de Hamming et en sélectionnant les rapports h_{k+1}/h_k les plus proches de 1 on découvre des combinaisons simples d'entiers de Hamming fournissant des valeurs pour x de l'ordre de 10^{-3}. Les trois rapports retenus sont :

$$\begin{aligned}
\frac{2\times 5^{18}}{3^{27}} &= \frac{1+u}{1-u} \text{ avec } u \approx 0.25\times 10^{-3},\\
\frac{2^{38}}{3^2\times 5^{15}} &= \frac{1+v}{1-v} \text{ avec } v \approx 0.40\times 10^{-3},\\
\frac{3^8\times 5}{2^{15}} &= \frac{1+w}{1-w} \text{ avec } w \approx 0.56\times 10^{-3}.
\end{aligned}$$

On a alors :

$$\begin{pmatrix}\ln 2\\ \ln 3\\ \ln 5\end{pmatrix} = \begin{pmatrix}118 & 171 & 441\\ 187 & 271 & 699\\ 274 & 397 & 1024\end{pmatrix}\times\begin{pmatrix}2\operatorname{argth}(u)\\ 2\operatorname{argth}(v)\\ 2\operatorname{argth}(w)\end{pmatrix}$$

donc il suffit de calculer $2\operatorname{argth}(u), 2\operatorname{argth}(v), 2\operatorname{argth}(w)$ avec 12 bits de précision supplémentaire pour obtenir des encadrements de $\ln(2), \ln(3), \ln(5)$ avec une précision fixée.

On veut des approximations de $\ln(2), \ln(3), \ln(5)$ à $1/1000n$ près pour rechercher le n-ème entier de Hamming, soit pour $n \leqslant 10^{100}$ une précision pour $2\operatorname{argth}(u), 2\operatorname{argth}(v), 2\operatorname{argth}(w)$ de l'ordre de 2^{-345} dans le pire des cas. Compte tenu des choix de u, v, w les termes généraux des séries $2\operatorname{argth}(u)$, $2\operatorname{argth}(v)$, $2\operatorname{argth}(w)$ décroissent au moins aussi vite que 2^{-20k} donc il suffit de calculer 17 termes dans le pire des cas pour chaque série. Les calculs sont conduits en nombres rationnels avec majoration naïve du reste de la série et arrêt quand la précision requise est atteinte.

4 Calcul approché de $V(a, b, c, k)$

On rappelle que $V(a, b, c, k) = \operatorname{card}\{(x, y, z) \in \mathbf{N}^3 \ / \ ax + by + cz \leqslant k\}$. La série génératrice par rapport à k associée est :

$$F_{a,b,c}(x) = \sum_{k=0}^{\infty} V(a, b, c, k)x^k = \frac{1}{(1-x)(1-x^a)(1-x^b)(1-x^c)}.$$

Il suffit donc de développer théoriquement cette fraction en série entière pour obtenir une formule exacte pour $V(a, b, c, k)$. Supposons que a, b, c sont deux à deux premiers entre eux :

$$\frac{1}{(1-x)(1-x^a)(1-x^b)(1-x^c)} = \frac{R(x)}{(1-x)^4} + \frac{A(x)}{(1-x^a)/(1-x)} + \frac{B(x)}{(1-x^b)/(1-x)} + \frac{C(x)}{(1-x^c)/(1-x)}$$

où R, A, B, C sont des polynômes vérifiant $\deg(R) < 4$, $\deg(A) < a-1$, $\deg(B) < b-1$, $\deg(C) < c-1$, car les polynômes $(1-x)^4$, $(1-x^a)/(1-x)$, $(1-x^b)/(1-x)$ et $(1-x^c)/(1-x)$ sont alors sans racine commune donc deux à deux premiers entre eux. Si l'on écrit :

$$\begin{aligned}
R(x) &= r_0 + r_1(1-x) + r_2(1-x)^2 + r_3(1-x)^3,\\
(1-x)A(x) &= \alpha_0 + \alpha_1 x + \ldots + \alpha_{a-1}x^{a-1},\\
(1-x)B(x) &= \beta_0 + \beta_1 x + \ldots + \beta_{b-1}x^{b-1},\\
(1-x)C(x) &= \gamma_0 + \gamma_1 x + \ldots + \gamma_{c-1}x^{c-1},
\end{aligned}$$

alors on obtient :

$$V(a,b,c,k) = r_0\binom{k+3}{3} + r_1\binom{k+2}{2} + r_2\binom{k+1}{1} + r_3 + \alpha_{k \bmod a} + \beta_{k \bmod b} + \gamma_{k \bmod c}.$$

Un calcul fastidieux mais sans difficulté particulière donne :

$$\begin{aligned}
r_0 &= 1/abc,\\
r_1 &= (a+b+c-3)/2abc,\\
r_2 &= (a^2+b^2+c^2+3ab+3ac+3bc-6a-6b-6c)/12abc,\\
r_3 &= (a^2b+ab^2+a^2c+ac^2+b^2c+bc^2+3abc-a^2-b^2-c^2-3ab-3ac-3bc+a+b+c)/24abc,\\
\alpha_p &= \frac{1}{a^3}\sum_{i+bj+ck\equiv p \bmod a}\Big(\frac{a-1}{2}-i\Big)\Big(\frac{a-1}{2}-j\Big)\Big(\frac{a-1}{2}-k\Big),
\end{aligned}$$

et des expressions similaires pour β_p et γ_p (la somme porte sur tous les triplets (i,j,k) vérifiant la relation de congruence indiquée). Ces sommes sont incalculables en pratique car elles comportent a^2 ou b^2 ou c^2 termes et a,b,c sont de l'ordre de $1000n^{2/3}$. En ne conservant que la partie polynomiale,

$$P_{a,b,c}(k) = r_0\binom{k+3}{3} + r_1\binom{k+2}{2} + r_2\binom{k+1}{1} + r_3$$

on obtient une formule calculable en temps constant et qui ne diffère de $V(a,b,c,k)$ que d'une quantité bornée lorsque a,b,c sont fixés et k est variable.

Dans les conditions de recherche du n-ème nombre de Hamming, il n'est pas garanti que a,b,c soient effectivement deux à deux premiers entre eux, et de plus le majorant naïf que l'on obtient pour $|V(a,b,c,k) - P_{a,b,c}(k)|$ est $(a^2+b^2+c^2)/8 \approx 4\times 10^5 n^{4/3}$ soit considérablement plus que la valeur que l'on veut approcher puisque $V(a,b,c,k)\approx n$. Ainsi *il est déraisonnable* d'utiliser $P_{a,b,c}(k)$ comme approximation de $V(a,b,c,k)$ mais pourtant des expérimentations pratiques montrent que cette approximation est inexplicablement précise. Par exemple dans le cas de la recherche du 10^{10}-ème entier de Hamming on a :

$$P_{a,b,c}(1050165619994) = 10000000001 \quad \text{et} \quad V(a,b,c,1050165619994) = 9999999997$$

soit une erreur de 4. Pour la recherche du 10^{100}-ème entier de Hamming, on constate de même une différence entre l'approximation et la valeur exacte valant 167 pour un volume exact de l'ordre de 10^{100} (voir le fichier `h100.txt` dans l'archive source). Cette précision reste à expliquer ou à infirmer, en attendant elle est utilisée dans le programme de recherche pour accélérer la phase d'approximations successives.

5 Réduction des réseaux

5.1 Algorithmes de réduction

La théorie de la réduction des réseaux est utilisée à la fois pour obtenir une approximation de la direction du triplet $(\ln(2), \ln(3), \ln(5))$ avec des nombres a, b, c plus petits que ceux fournis par l'encadrement initial et aussi en tant que composant dans l'algorithme de Barvinok. Dans les deux cas il s'agit, étant donnés trois vecteurs $\mathbf{u}, \mathbf{v}, \mathbf{w}$ de $\mathbf{Z}^3$ linéairement indépendants sur $\mathbf{R}$ de trouver une base $(\mathbf{u}', \mathbf{v}', \mathbf{w}')$ du groupe additif G engendré par $\mathbf{u}, \mathbf{v}, \mathbf{w}$ « réduite » pour un certain produit scalaire. Il y a deux notions de base réduite :

Réduction au sens de Minkowski : $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ est une base réduite de G au sens de Minkowski si et seulement si c'est une base de G et si pour toute base $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ de G avec $||\mathbf{x}|| \leqslant ||\mathbf{y}|| \leqslant ||\mathbf{z}||$ on a $||\mathbf{u}|| \leqslant ||\mathbf{x}||$, $||\mathbf{v}|| \leqslant ||\mathbf{y}||$ et $||\mathbf{w}|| \leqslant ||\mathbf{z}||$.

Réduction au sens de Lovász : $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ est une base réduite de G au sens de Lovász si et seulement si c'est une base de G et si les angles entre deux vecteurs valent au moins 60° et si les projetés $\mathbf{v}^*$ et $\mathbf{w}^*$ de $\mathbf{v}$ et $\mathbf{w}$ sur le plan normal à $\mathbf{u}$ vérifient : $||\mathbf{w}^*||^2 \geqslant \frac{3}{4}||\mathbf{v}^*||^2$.

L'intérêt des bases réduites est de permettre une énumération rapide des vecteurs de G ayant une petite norme. En effet, on démontre que si $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ est réduite au sens de Minkowski alors les coordonnées d'un vecteur $\mathbf{x} \in \mathbf{G}$ de norme inférieure ou égale à M sont majorées en valeur absolue par $M\sqrt{2}/||\mathbf{u}||$ donc on peut énumérer ces vecteurs par exploration systématique quand $M/||\mathbf{u}||$ est petit. Pour une base réduite au sens de Lovász les coordonnées de $\mathbf{x}$ sont majorées par $2M\sqrt{2}/\min(||\mathbf{u}||, ||\mathbf{v}||, ||\mathbf{w}||)$. Par ailleurs, la norme minimale d'un vecteur non nul de G est égale à $||\mathbf{u}||$ si $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ est réduite au sens de Minkowski et est minorée par $\frac{1}{2}||\mathbf{u}||$ si $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ est réduite au sens de Lovász.

Dans le cadre de la recherche du n-ème entier de Hamming on a testé deux algorithmes de réduction :

– l'algorithme AIGU de Brigitte Vallée qui détermine une base réduite au sens de Minkowski pour un groupe G spécifié par la matrice de Gram, entière ou rationnelle, d'une base arbitraire de G. Cet algorithme est spécifique à la dimension 3, il est décrit dans [5].

– l'algorithme LLL en nombres entiers de Arjen K. Lenstra, Hendrik W. Lenstra et L. Lovász qui détermine une base réduite au sens de Lovász du groupe G à partir de la matrice de Gram, entière ou rationnelle, d'une base arbitraire de G. Cet algorithme est applicable en toute dimension et est décrit dans [1] et [5].

Ces deux algorithmes ont une complexité proportionnelle à la taille en bits de la matrice de Gram fournie en argument (lorsque la dimension est fixée). Expérimentalement l'algorithme AIGU est plus rapide que l'algorithme LLL pour la recherche d'une approximation réduite de $(\ln 2, \ln 3, \ln 5)$ mais moins rapide que LLL dans le cadre de la décomposition de Barvinok.

5.2 Utilisation de la réduction pour approcher la direction de $(\ln 2, \ln 3, \ln 5)$

Soient $\alpha 2^{-\delta}, \beta 2^{-\delta}, \gamma 2^{-\delta}$ des approximations rationnelles à $2^{-\delta}$ près de $\ln 2, \ln 3, \ln 5$, obtenues lors de la première phase par calcul approché de logarithmes (voir la section 3). On cherche un vecteur (a, b, c) dans $\mathbf{Z}^3$ non nul tel que $|a|$, $|\alpha b - \beta a|$, $|\alpha c - \gamma a|$ soient petits. Pour cela on prend comme forme quadratique sur $\mathbf{R}^3$ la fonction :

$$Q \ : \ (a, b, c) \longmapsto a^2 + (b\alpha - a\beta)^2 + (c\alpha - a\gamma)^2$$

et on calcule une base réduite $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ de $\mathbf{Z}^3$ pour le produit scalaire associé. Avec la réduction au sens de Minkowski $\mathbf{u}$ est un vecteur non nul de $\mathbf{Z}^3$ de norme minimale donc constitue un choix optimal pour (a, b, c). Si l'on utilise la réduction de Lovász alors $\mathbf{u}$ n'est pas nécessairement de norme minimale, mais c'est souvent le cas et en tout état de cause $||\mathbf{u}||$ est au plus deux fois plus grande que la norme minimale d'un vecteur entier non nul donc il n'est pas crucial de chercher un meilleur vecteur.

D'après le théorème sur les approximations simultanées de Dirichlet (si $\alpha_1, \ldots, \alpha_k$ sont des réels et $N \in \mathbf{N}^*$ alors il existe un entier $p \in [\![1, N^k]\!]$ et des entiers $p_1, \ldots, p_k$ tel que l'on ait pour tout i : $|p_i - p\alpha_i| \leqslant 1/N$), pour tout entier $n \in \mathbf{N}^*$ il existe des entiers p, q, r tels que :

$$1 \leqslant p \leqslant \lceil n^{1/3} \rceil^2, \qquad \left|q - p\frac{\ln 3}{\ln 2}\right| \leqslant \frac{1}{n^{1/3}}, \qquad \left|r - p\frac{\ln 5}{\ln 2}\right| \leqslant \frac{1}{n^{1/3}}.$$

On a alors :

$$\left|q\alpha - p\beta\right| \leqslant \alpha\left|q - p\frac{\ln 3}{\ln 2}\right| + \frac{p}{\ln 2}\left|\alpha \ln 3 - \beta \ln 2\right| \leqslant \frac{2^\delta}{n^{1/3}} \ln 2 + p \ln 3 \leqslant K n^{2/3}$$

et de même $|r\alpha - p\gamma| \leqslant K' n^{2/3}$ donc $Q(a, b, c) \leqslant 4Q(p, q, r) \leqslant K'' n^{4/3}$ en particulier a, b, c sont dominés par $n^{2/3}$. Le temps de calcul de a, b, c est linéaire en $\ln(\alpha) + \ln(\beta) + \ln(\gamma)$ et donc est linéaire en $\ln(n)$.

5.3 Utilisation de la réduction dans l'algorithme de Barvinok

Soient $\mathbf{u}, \mathbf{v}, \mathbf{w}$ trois vecteurs entiers tels que $|\det(\mathbf{u}, \mathbf{v}, \mathbf{w})| = d > 1$. Il s'agit de trouver un vecteur $\mathbf{x}$ entier non nul tel que les trois déterminants $\det(\mathbf{x}, \mathbf{v}, \mathbf{w})$, $\det(\mathbf{u}, \mathbf{x}, \mathbf{w})$, $\det(\mathbf{u}, \mathbf{v}, \mathbf{x})$ soient en valeur absolue majorés par $d^{2/3}$ (l'existence d'un tel vecteur est assurée par le théorème de Minkowski). Pour cela on prend comme forme quadratique sur $\mathbf{R}^3$ la fonction :

$$Q \ : \ \mathbf{x} \longmapsto \det(\mathbf{x}, \mathbf{v}, \mathbf{w})^2 + \det(\mathbf{u}, \mathbf{x}, \mathbf{w})^2 + \det(\mathbf{u}, \mathbf{v}, \mathbf{x})^2$$

et on calcule une base $(\mathbf{u}', \mathbf{v}', \mathbf{w}')$ de $\mathbf{Z}^3$ réduite pour le produit scalaire associé.

Si $\min(\|\mathbf{u}'\|, \|\mathbf{v}'\|, \|\mathbf{w}'\|) \leqslant d^{2/3}$ alors le vecteur $\mathbf{u}'$, $\mathbf{v}'$ ou $\mathbf{w}'$ correspondant au minimum convient. Sinon, les coordonnées dans la base $(\mathbf{u}', \mathbf{v}', \mathbf{w}')$ du vecteur $\mathbf{x}$ cherché sont en valeur absolue majorées par $2\sqrt{6} < 5$ car $\|\mathbf{x}\| \leqslant d^{2/3}\sqrt{3}$ donc il suffit d'examiner au plus $(9^3 - 1)/2$ vecteurs pour trouver $\mathbf{x}$. Le temps de calcul de $\mathbf{x}$ est donc linéaire en la somme des tailles en bits des vecteurs $\mathbf{u}, \mathbf{v}, \mathbf{w}$.

6 Algorithme de Barvinok

6.1 Théorie algébrique des polyèdres

6.1.1 Série génératrice d'un polyèdre et fraction associée

La théorie algébrique des polyèdres est exposée dans [2], on rappelle ici seulement les résultats qui seront utilisés dans la recherche des entiers de Hamming. Un polyèdre rationnel en dimension n est une partie de $\mathbf{R}^n$ définie par un nombre fini d'inéquations à coefficients rationnels. Un cône rationnel de sommet $\mathbf{a} \in \mathbf{R}^n$ est un polyèdre rationnel P pour lequel toutes les inéquations définissant P sont des égalités en $\mathbf{a}$. Un tel cône est de la forme :

$$P = \{\mathbf{a} + \alpha_1 \mathbf{u_1} + \ldots + \alpha_k \mathbf{u_k} \text{ tq } \alpha_1 \geqslant 0, \ldots, \alpha_k \geqslant 0\}$$

où $(\mathbf{u_1}, \ldots, \mathbf{u_k})$ est une famille fixée de vecteurs de $\mathbf{Z}^n$. On dit que P est un cône de dimension k si $(\mathbf{u_1}, \ldots, \mathbf{u_k})$ est libre, et que P est un cône unimodulaire si $k = n$ et $\det(\mathbf{u_1}, \ldots, \mathbf{u_n}) = \pm 1$.

A un polyèdre P on associe la série formelle à n variables :

$$S_P(\mathbf{x}) = \sum_{\mathbf{p} \in P \cap \mathbf{Z}^n} \mathbf{x}^{\mathbf{p}}$$

où $\mathbf{x} = (x_1, \dots, x_n)$, $\mathbf{p} = (p_1, \dots, p_n)$ et $\mathbf{x}^{\mathbf{p}} = x_1^{p_1} \dots x_n^{p_n}$. Soit $\mathscr{P}$ l'espace vectoriel engendré par les séries S_P quand P décrit l'ensemble des polyèdres rationnels de $\mathbf{R}^n$. Il existe une unique application $\Phi : \mathscr{P} \longrightarrow \mathbf{Q}[X_1, \dots, X_n]$ telle que :

- Φ est linéaire ;
- pour tout polyèdre P et tout n-uplet $\mathbf{x} \in \mathbf{C}^n$, si la série S_P converge en $\mathbf{x}$ alors sa somme est égale à $\Phi(S_P)(\mathbf{x})$;
- si P contient une droite alors $\Phi(S_P) = 0$.

L'algorithme de Barvinok consiste à calculer explicitement $\Phi(S_P)$ sous forme d'une somme de fractions simples. On se ramène au cas où P est un cône grâce au théorème suivant :

Soit P un polyèdre rationnel. Si $\mathbf{a}$ est un point extrémal de P on note $\text{Cône}(P, \mathbf{a})$ le plus petit cône de sommet $\mathbf{a}$ contenant P. Alors :

$$\Phi(S_P) = \sum_{\mathbf{a} \text{ extrémal dans } P} \Phi(S_{\text{Cône}(P,\mathbf{a})}).$$

6.1.2 Fraction associée à un cône de dimension n

Soit P un cône de sommet $\mathbf{a}$ et de dimension n engendré par la famille $(\mathbf{u_1}, \dots, \mathbf{u_n}) \in (\mathbf{Z}^n)^n$. On note $d = |\det(\mathbf{u_1}, \dots, \mathbf{u_n})|$. Si P est unimodulaire alors $(\mathbf{u_1}, \dots, \mathbf{u_n})$ est une base de $\mathbf{Z}^n$ et on a :

$$\Phi(S_P)(\mathbf{x}) = \frac{\mathbf{x}^{\lfloor \mathbf{a} \rfloor}}{(\mathbf{1} - \mathbf{x}^{\mathbf{u_1}}) \dots (\mathbf{1} - \mathbf{x}^{\mathbf{u_n}})}$$

avec $\mathbf{a} = \alpha_1 \mathbf{u_1} + \dots + \alpha_n \mathbf{u_n}$ et $\lfloor \mathbf{a} \rfloor = \lfloor \alpha_1 \rfloor \mathbf{u_1} + \dots + \lfloor \alpha_n \rfloor \mathbf{u_n}$.

Si P n'est pas unimodulaire, soit si $d > 1$, on introduit un vecteur $\mathbf{x} \in \mathbf{Z}^n$ non nul tel que tous les déterminants :

$$d_i = |\det(\mathbf{u_1}, \dots, \mathbf{u_{i-1}}, \mathbf{x}, \mathbf{u_{i+1}}, \dots \mathbf{u_n})|$$

soient majorés par $d^{2/3}$. Un tel vecteur $\mathbf{x}$ peut être calculé par réduction d'un réseau comme décrit dans la section 5. Soit P_i le cône de sommet $\mathbf{a}$ engendré par les vecteurs $\mathbf{u_1}, \dots, \mathbf{u_{i-1}}, \mathbf{x}, \mathbf{u_{i+1}}, \dots \mathbf{u_n}$. Si l'on a :

$$\mathbf{x} = x_1 \mathbf{u_1} + \dots + x_k \mathbf{u_k} - x_{k+1} \mathbf{u_{k+1}} - \dots - x_n \mathbf{u_n}$$

avec $x_1, \dots, x_n$ réels positifs alors :

$$P \cup P_{k+1} \cup \dots \cup P_n = P_1 \cup \dots \cup P_k$$

d'où, d'après la formule du crible :

$$S_P = S_{P_1} + \dots + S_{P_k} - S_{P_{k+1}} - \dots - S_{P_n} + \sum \pm S_{\text{cône de dimension } < n},$$

et donc :

$$\Phi(S_P) = \Phi(S_{P_1}) + \dots + \Phi(S_{P_k}) - \Phi(S_{P_{k+1}}) - \dots - \Phi(S_{P_n}) + \sum \pm \Phi(S_{\text{cône de dimension } < n}).$$

On peut ainsi calculer $\Phi(S_P)$ par récurrence sur d et sur la dimension n.

6.1.3 Fraction associée à un cône quelconque

Soit P un cône rationnel de sommet $\mathbf{0}$. On note P^* le cône dual défini par :

$$P^* = \{\mathbf{x} \in \mathbf{R}^n \text{ tq } \forall\, \mathbf{y} \in P,\ (\mathbf{x} \mid \mathbf{y}) \geqslant 0\}.$$

P^* est encore un cône rationnel de sommet $\mathbf{0}$ et l'on a $P^{**} = P$. En termes de séries génératrices, on démontre qu'il existe une unique application linéaire $\Delta : \mathscr{P} \longrightarrow \mathscr{P}$ telle que pour tout cône P de sommet $\mathbf{0}$ on a :

$$\Delta(S_P) = S_{\mathbf{R}^n} - S_{P^*}.$$

On peut décomposer le cône P^* en une réunion de cônes de dimension inférieure ou égale à n et décomposer chacun des cônes de dimension n en cônes unimodulaires. On obtient ainsi une relation de la forme :

$$S_{P^*} = \sum_i \varepsilon_i S_{P_i} + \sum \pm S_{\text{cône de dimension } < n}$$

où les P_i sont des cônes unimodulaires et $\varepsilon_i = \pm 1$. Par dualité on a alors :

$$S_{\mathbf{R}^n} - S_P = \sum_i \varepsilon_i (S_{\mathbf{R}^n} - S_{P_i^*}) + \sum \pm (S_{\mathbf{R}^n} - S_{\text{dual de cône de dimension } < n})$$

puis :

$$\Phi(S_P) = \sum_i \varepsilon_i \Phi(S_{P_i^*})$$

car Φ s'annule sur toute série d'un polyèdre contenant une droite. Le cône dual d'un cône unimodulaire engendré par $\mathbf{u_1}, \dots, \mathbf{u_n}$ avec $\det(\mathbf{u_1}, \dots, \mathbf{u_n}) = 1$ est le cône unimodulaire engendré par les colonnes de la comatrice de $(\mathbf{u_1}, \dots, \mathbf{u_n})$ dans la base canonique de $\mathbf{R}^n$. On peut donc calculer $\Phi(S_P)$ en décomposant P^* en cônes unimodulaires et en négligeant tous les cônes de dimension strictement inférieure à n qui apparaissent au cours des calculs. On a de même, pour tout $\mathbf{a} \in \mathbf{R}^n$:

$$\Phi(S_{P+\{\mathbf{a}\}}) = \sum_i \varepsilon_i \Phi(S_{P_i^*+\{\mathbf{a}\}})$$

donc on peut calculer explicitement la fraction rationnelle associée à un cône rationnel quelconque.

6.2 Application au calcul du n-ème entier de Hamming

Soient $a, b, c, k \in \mathbf{N}$. Pour calculer $V(a,b,c,k) = \operatorname{card}\{(x,y,z) \in \mathbf{N}^3 \ / \ ax+by+cz \leqslant k\}$ on décompose le tétraèdre P associé en cônes unimodulaires selon l'algorithme de Barvinok. P a quatre points extrémaux : $O = (0,0,0)$, $A = (k/a, 0, 0)$, $B = (0, k/b, 0)$ et $C = (0,0,k/c)$. Le cône circonscrit à P en O est le trièdre engendré par la base canonique de $\mathbf{R}^3$, la fraction associée est :

$$\Phi(S_{\text{Cône}(P,O)})(x,y,z) = \frac{1}{(1-x)(1-y)(1-z)}.$$

Le cône $K + \{A\}$ circonscrit à P en A est engendré par les vecteurs : $\mathbf{u} = (-1,0,0)$, $\mathbf{v} = (-b, a, 0)$ et $\mathbf{w} = (-c, 0, a)$. Le cône dual de K est engendré par les vecteurs : $\mathbf{u}' = (-a,-b,-c)$, $\mathbf{v}' = (0,1,0)$, et $\mathbf{w}' = (0,0,1)$ de déterminant $-a = O(n^{2/3})$. Chaque étape de l'algorithme de Barvinok échange un cône tridimensionnel de déterminant d contre trois cônes tridimensionnels de déterminants inférieurs ou égaux à $d^{2/3}$. Après p étapes de décomposition, on obtient au plus 3^p cônes de déterminants inférieurs ou égaux à $d^{(2/3)^p}$ donc il faut au plus $\lceil \ln(\ln d/\ln 2)/\ln(3/2) \rceil$ étapes pour décomposer $\text{Cône}(P,A)$ en cônes unimodulaires. Le nombre total de cônes unimodulaires obtenus est majoré par $3^{\text{cette quantité}}$ soit $O(\ln(d)^{\ln(3)/\ln(3/2)}) = O(\ln(n)^{2.71})$. Les cônes circonscrits à P en B et en C relèvent de la même analyse donc on obtient finalement une expression de $\Phi(S_P)$ sous la forme :

$$\Phi(S_P)(\mathbf{x}) = \sum_i \frac{\varepsilon_i \mathbf{x}^{\lfloor k\mathbf{t}_i \rfloor}}{(1-\mathbf{x}^{\mathbf{u}_i})(1-\mathbf{x}^{\mathbf{v}_i})(1-\mathbf{x}^{\mathbf{w}_i})}$$

où $\varepsilon_i = \pm 1$ et $\mathbf{u}_i, \mathbf{v}_i, \mathbf{w}_i$ sont des vecteurs entiers et $\mathbf{t}_i$ est un vecteur rationnel, ces vecteurs étant indépendants de k. Cette somme comporte $O(\ln(n)^{2.71})$ termes soit considérablement moins que l'expression développée déduite de la définition :

$$\Phi(S_P)(\mathbf{x}) = \sum_{\mathbf{p} \in P \cap \mathbf{Z}^3} \mathbf{x}^{\mathbf{p}} \qquad \text{(environ } n \text{ termes).}$$

Le temps de calcul est plus délicat à estimer : chaque étape de l'algorithme de Barvinok se ramène essentiellement à une réduction de réseau de complexité linéaire en $\ln(M)$ ou M majore la taille des vecteurs directeurs du cône à décomposer. Cette taille est elle-même bornée par un polynôme en $\ln(d)$ où d est le déterminant du cône initial mais il est difficile de suivre les majorations théoriques dans les méandres des algorithmes de réduction et de l'algorithme de Barvinok pour préciser le degré de ce polynôme. Expérimentalement, on constate que le nombre de cônes croît comme $\ln(n)^{2.5}$ et le temps de calcul croît environ comme $\ln(n)^3$ avec l'algorithme LLL. Avec l'algorithme AIGU le temps de calcul augmente considérablement plus vite :

Décomposition de Barvinok avec réduction par LLL

$\log(n)$	10	20	30	40	50	60	70	80	90	100
nb. cônes	796	2316	5842	11308	18992	26926	40396	60196	81940	108676
temps en ms	360	1340	4250	10340	20850	33520	61130	107870	170340	259890
$\log(\text{cônes})/\log(\log(n))$	2.90	2.59	2.55	2.53	2.52	2.49	2.50	2.51	2.51	2.52
$\text{temps}/\log(n)^3$	0.36	0.17	0.16	0.16	0.17	0.16	0.18	0.21	0.23	0.26

Décomposition de Barvinok avec réduction par l'algorithme AIGU

$\log(n)$	10	20	30	40	50	60	70	80	90	100
nb cônes	756	2326	5706	11350	19068	29604	40560	58970	81300	–
temps en ms	370	1370	4630	15150	170970	131420	3.27e6	44.2e6	35.1e6	–
$\log(\text{cônes})/\log(\log(n))$	2.88	2.59	2.54	2.53	2.52	2.51	2.50	2.51	2.51	–
$\text{temps}/\log(n)^3$	0.37	0.17	0.17	0.24	1.37	0.61	9.55	86.33	48.10	–

6.3 Calcul exact de $V(a, b, c, k)$

La formule :

$$\Phi(S_P)(\mathbf{x}) = \sum_i \frac{\varepsilon_i \mathbf{x}^{\lfloor k\mathbf{t}_i \rfloor}}{(1-\mathbf{x}^{\mathbf{u}_i})(1-\mathbf{x}^{\mathbf{v}_i})(1-\mathbf{x}^{\mathbf{w}_i})}$$

ne permet pas un calcul direct de $V(a, b, c, k) = \Phi(S_P)(1, 1, 1)$ car les dénominateurs s'annulent pour $\mathbf{x} = (1, 1, 1)$. Barvinok propose une approche analytique pour contourner ce problème : $\Phi(S_P)$ est en réalité un polynôme donc est continu en $(1, 1, 1)$. Il suffit de calculer la limite de $\Phi(S_P)(\mathbf{x})$ quand $\mathbf{x}$ tend vers $(1, 1, 1)$ sans annuler aucun dénominateur. Pour cela on pose $\mathbf{x} = (e^{\alpha h}, e^{\beta h}, e^{\gamma h})$ où h est une variable réelle tendant vers 0 et $\mathbf{p} = (\alpha, \beta, \gamma)$ est un vecteur entier choisi de sorte qu'aucun produit scalaire $(\mathbf{p} \mid \mathbf{u_i})$, $(\mathbf{p} \mid \mathbf{v_i})$, $(\mathbf{p} \mid \mathbf{w_i})$ ne soit nul. Il existe un tel vecteur car il n'y a qu'un nombre fini de plans à éviter et $\mathbf{Z}^3$ n'est pas union d'un nombre fini de plans. On peut déterminer $\mathbf{p}$ par exploration systématique en temps proportionnel au carré du nombre de fractions composant $\Phi(S_P)$, mais en pratique on constate qu'une énumération des petits vecteurs de $\mathbf{Z}^3$ fournit un vecteur $\mathbf{p}$ convenable en quelques essais seulement.

Un développement limité théorique donne :

$$\frac{e^{th}}{(1-e^{uh})(1-e^{vh})(1-e^{wh})} = \frac{\lambda_3}{h^3} + \frac{\lambda_2}{h^2} + \frac{\lambda_1}{h} + \lambda_0 + \underset{h \to 0}{o}(h)$$

avec :

$$\lambda_0 = \frac{(u+v+w-2t)(uv+uw+vw-2t(u+v+w)+2t^2)}{24uvw}$$

pour $u, v, w, t \in \mathbf{R}$, u, v, w étant non nuls. Il ne reste donc plus qu'à remplacer u par $(\mathbf{p} \mid \mathbf{u_i})$, v par $(\mathbf{p} \mid \mathbf{v_i})$, w par $(\mathbf{p} \mid \mathbf{w_i})$ et t par $(\mathbf{p} \mid \lfloor k\mathbf{t_i} \rfloor)$ dans l'expression de λ_0 et à additionner/retrancher les termes ainsi obtenus pour toutes les fractions composant $\Phi(S_P)$. Il subsiste une dernière difficulté : pour chaque fraction le λ_0 associé est un nombre rationnel et il y a plus de 100000 tels rationnels à additionner/soustraire pour $n = 10^{100}$. Même en mettant le coefficient 1/24 en facteur, le calcul exact en nombre rationnels est prohibitif car les dénominateurs uvw ne se simplifient « qu'à la fin ». On contourne le problème en calculant λ_0 en virgule fixe en arrondissant chaque fraction à 2^{-32} près. Tant que le nombre de fractions composant $\Phi(S_p)$ est inférieur à 24×2^{31}, l'entier le plus proche de la somme totale est la valeur exacte de $V(a, b, c, k)$. La même méthode de développement limité et de calcul en virgule fixe est utilisée pour calculer les dérivées partielles de $\Phi(S_P)$ en $(1, 1, 1)$ lors de la phase d'identification du n-ème point de Hamming une fois qu'il a été isolé entre deux tétraèdres rationnels.

7 Conclusion

La chasse au n-ème entier de Hamming est-elle terminée ? Si l'on reprend l'historique du problème de Hamming, on constate que l'on est passé en 1998 d'un algorithme paresseux de calcul du n-ème entier de Hamming de complexité n, déduit directement de la définition des nombres de Hamming à un algorithme de complexité $n^{2/3}\ln(n)$ (première méthode géométrique, par comptage) puis $n^{1/3}\ln(n)^2$ (comptage par arbres différentiels et première méthode volumétrique) pour atteindre à présent $n^{0/3}\ln(n)^3$ (cette dernière complexité n'est pas prouvée, elle est seulement constatée expérimentalement). L'étape logique suivante serait un algorithme de complexité $n^{-1/3}\ln(n)^4$ mais il semble peu probable que l'on découvre un jour un algorithme dont le temps d'exécution *décroît* avec n donc les seules améliorations que l'on peut encore espérer porteront sur l'exposant de $\ln(n)$. Mon but en 1998 était de pouvoir calculer le heptantillionnième entier de Hamming en quelques secondes (un heptantillion $= 10^{420}$). Le but n'est pas atteint mais on n'en est pas loin, il suffit avec l'algorithme polyédral de disposer d'une machine cent fois plus rapide et ayant cent fois plus de mémoire que celle ayant servi au calcul de $h_{10^{100}}$ (processeur Athlon 550 Mhz avec 128 Mo de mémoire vive), ce qui est l'affaire de quelques années. On peut aussi chercher un nouvel algorithme de complexité $\ln(n)^2$; l'avenir dira lequel de ces deux évènements se produira le premier.

La recherche des entiers de Hamming peut sembler un exercice puéril et vain, pourtant elle a le mérite de mobiliser sur un même problème à la fois l'arithmétique, l'analyse, l'algèbre, la géométrie, l'algorithmique et la structuration des données. Voici pour terminer les « coordonnées » du 10^{100}-ème entier de Hamming :

$$
\begin{aligned}
h_{10^{100}} &= 2^x 3^y 5^z \text{ avec} \\
x &= 4\,797\,740\,003\,723\,871\,675\,239\,153\,566\,333\,597, \\
y &= 332\,525\,835\,937\,443\,599\,429\,449\,538\,321\,604, \\
z &= 309\,841\,957\,917\,623\,833\,100\,064\,425\,409\,784.
\end{aligned}
$$

8 Bibliographie

[1] Henri Cohen
A Course in Computational Algebraic Number Theory
Springer-Verlag, 2000

[2] Alexander Barvinok, James E. Pommersheim
An algorithmic theory of lattice points in polyhedra
`http://www.math.lsa.umich.edu/~barvinok/papers.html`

[3] Dijkstra, Edsger Wybe
A discipline of programming

Prentice Hall 1976

[4] Michel Quercia
Calcul du n*-ème entier de Hamming*
`http://pauillac.inria.fr/~quercia/papers/hamming.ps.gz`

[5] Brigitte Vallée
Une approche géométrique de la réduction des réseaux en petite dimension
Thèse de doctorat, université de Caen

Table des matières