Les travaux de Madhu Sudan sur les codes correcteurs d'erreurs

Daniel Augot

Résumé

Nous présentons les travaux de Madhu Sudan en théorie des codes correcteurs, qui, parmi d'autres [1], lui valurent de recevoir le prix Nevanlinna en Août 2002 [2], qui est le pendant de la médaille Fields, dans le domaine des aspects mathématiques de l'informatique. La percée majeure de M. Sudan est d'avoir produit un algorithme de décodage des codes de Reed-Solomon bien au delà de la capacité de correction de ces codes, en autorisant de pouvoir décoder en retournant comme résultat une liste de solutions plutôt qu'une unique solution, ce que l'on appelle le "décodage en liste".

Il s'agit d'une avancée théorique fondamentale dont les conséquences pratiques et théoriques ne sont pas encore mesurées. Dans cette présentation, nous introduirons l'arrière-plan pratique de la théorie des codes, puis les codes de Reed-Solomon et les codes géométriques. Enfin, nous présenterons l'algorithme de M. Sudan, comme une généralisation de l'algorithme de Berlekamp-Welsh.

Cet exposé est basé sur la présentation que M. Sudan a lui même faite dans [6].

1 Introduction et Définitions

La théorie des codes est la discipline des mathématiques appliquées dont le sujet est la transmission fiable d'informations sur un canal de transmission bruité, en utilisant des objets combinatoires et algorithmiques appelés *codes correcteurs d'erreurs*. Pour introduire le sujet, il est d'abord nécessaire de préciser les notions de base du codage.

Suivons pour cela un message sur le chemin depuis la source jusqu'au récepteur, et observons les notions intéressantes qui apparaissent. Il y a trois entités impliquées dans le processus : l'émetteur, le récepteur et le canal de transmission. L'objectif de l'émetteur est de communiquer au récepteur un *message*, m, appartenant à $\mathcal{M}$ où $\mathcal{M}$ est un ensemble fini, *l'espace des messages*. Le canal de transmission bruité est capable de communiquer des suites arbitrairement longues de

1. Ses autres travaux portent sur la théorie de la complexité, et il est l'un des auteurs du célébre théorème PCP [8], en rapport avec la conjecture $P \neq NP$.
2. http://www.maa.org/news/fields02.html/

symboles d'un alphabet Σ, qui est "petit" (un des cas les plus intéressants étant $\Sigma = \{0, 1\}$). Alors l'espace des messages à coder est Σ^k, l'ensemble des suites de symboles de longueur k.

Émetteur et récepteur se mettent d'accord sur la longueur n des suites codées transmises, appelée la *longueur du code*, les messages échangés appartenant donc à Σ^n, que l'on appellera l'*espace ambiant*. L'émetteur et le récepteur se mettent aussi d'accord sur une *fonction de codage*, E, injective : $E : \mathcal{M} \to \Sigma^n$, utilisée pour coder les messages avant transmission. L'image $C = \{E(m), \quad m \in \mathcal{M}\}$ est appelé le *code*. Le taux k/n, noté en général R, est appelé le *taux de transmission* ou *rendement* du code, c'est le premier paramètre fondamental d'un code, en théorie des codes.

En ce qui concerne le canal de transmission, il "bruite" les messages transmis. Ce bruit peut être vu comme une application de l'espace ambiant dans lui même. Prescrivons maintenant une structure de corps sur l'alphabet Σ (par exemple Σ est un corps fini, de petite taille), ce qui induit une structure d'espace vectoriel sur Σ^n. Il devient alors plus commonde de considérer des *codes linéaires* c'est-à-dire l'image par une application linéaire de Σ^k dans Σ^n, que l'on supposera toujours non singulière. On spécifiera dorénavant un code linéaire C par sa matrice génératrice (une base de C), ce qui est une manière compacte de décrire un ensemble a priori de taille q^k. En ce qui concerne le canal de transmission, il produit un vecteur de bruit $e \in \Sigma^n$, et que le mot reçu est $y = E(m)+e$, m étant le message. Le récepteur utilise alors une fonction de décodage $D : \Sigma^n \to \mathcal{M}$. Le décodage D doit être rapide, et être tel que $D(y) = m$, avec grande probabilité. Intuitivement, le code introduit une redondance en augmentant la longueur des messages, et cette redondance sera utilisée pour décoder le message transmis, même s'il est bruité. Du point de vue de la fiabilité de la transmission, la question fondamentale de la théorie de codes est

> Étant donnée une distribution de probabilité P sur le canal de transmission (i.e. une distribution de probabilité sur les erreurs de transmission), quelles sont les meilleures fonctions de codage et de décodage, c'est-à-dire quelle est la plus petite probabilité d'erreur
>
> $$\min_{E,D} \left\{ \mathbf{E}_{m \in \mathcal{M}} \left(\Pr_{\eta \in P} [D(E(m) + \eta) \neq m] \right) \right\}$$
>
> où $\mathbf{E}$ désigne l'espérance mathématique.

Shannon a étudié les propriétés asymptotiques de cette quantité quand la distribution du bruit sur Σ^n est le produit cartésien d'une distribution sur Σ. Dans ce contexte, il existe une quantité $C_0 \in [0, 1]$, dépendant du canal, telle que pour tout $R < C_0$ et $\epsilon > 0$, et, pour n assez grand, il existe toujours un couple codage/décodage avec un code de taux R tel que la probabilité d'erreur soit au plus ϵ. Dans le cadre de cet exposé, nous ne considérerons uniquement le cas du *canal q-aire symétrique*, défini de la manière suivante : chaque symbole de Σ transmis est préservé avec une certaine probabilité $1 - \delta$, ou bien est transformé

en autre symbole parmi les $q-1$ autres possible avec probabilité $\delta/(q-1)$, les événements étant indépendants d'un symbole à l'autre.

D'un autre coté, Hamming a défini les notions de *code correcteur d'erreur* et de code *détecteur d'erreur*. Définissons le *poids de Hamming* d'une séquence $x \in \Sigma^n$ comme le nombre de composantes non nulles de x, et la *distance de Hamming* entre x et y comme le poids de la différence $x - y$ (c'est-à-dire le nombre de composantes où x et y diffèrent). C'est bien une distance. On définit alors la distance minimale d'un code C comme la plus petite distance entre deux mots du code C. Le canal de transmission crée en général un vecteur η de petit poids, par exemple de poids borné par e. On dira qu'un code correcteur corrige e erreurs si les boules de rayon e centrées sur les mots de code ne s'intersectent pas. En effet si le poids de l'erreur est inférieur à e, alors, si C est e-correcteur, il y a unicité du mot de code le plus proche. Une capacité de correction e implique que la distance minimale entre deux mots distincts du code est supérieure à $2e+1$. La distance minimale est le deuxième paramètre fondamental d'un code. On parlera d'un code $[n, k, d]$ pour un code de longueur n, de dimension k et de distance minimale d. Du point de vue de Hamming, la question fondamentale est

> Étant donné un alphabet Σ de taille q, et deux entiers n et k, $k < n$, quelle est la plus grande distance minimale d d'un code $C \subseteq \Sigma^n$ de taux de transmission k/n?

En effet, une distance minimale élevée induit que le code est capable de corriger des erreurs de poids élevé. Signalons immédiatement que le problème de Hamming n'est pas résolu quand la taille de l'alphabet est petite. Il y a une réponse satisfaisante à la question quand $q \geq n$ (voir les codes de Reed-Solomon dans la section 2).

2 Constructions de codes

Nous commencerons par la famille des codes aléatoires linéaires (en fait une *non-construction*). La borne de Varshamov-Gilbert indique qu'il existe des codes de paramètres $[n, k, d]$ si n, k et d vérifient :

$$q^k V_q(n, d) \leq q^n,$$

où $V_q(n, d)$ est le volume de la sphère de Hamming de rayon d (c'est-à-dire son cardinal). Donc il existe des codes tels que $q^k V_q(n, d) \geq q^n$. En prenant le logarithme et en approchant $V_q(n, \delta n)$ par $q^{H_q(\delta)n}$ (où $H_q(x)$ désigne la fonction d'entropie q-aire : $H_q(\delta) = -\delta \log_q(\frac{\delta}{q-1}) - (1-\delta)\log_q(1-\delta)$), on obtient l'existence de codes sur la borne suivante :

$$R \geq 1 - H_q(\delta), \text{ avec } R = \frac{k}{n} \text{ et } \delta = \frac{k}{n}.$$

Ce résultat s'étend "particulièrement" aux codes aléatoires : avec une probabilité tendant vers 1 quand la longueur n croît, les codes aléatoires se trouvent sur la

borne de Varshamov-Gilbert. La question qui en découle est de savoir s'il existe des codes dépassant la borne de Varshamov-Gilbert.

En dehors des codes aléatoires, la théorie des codes s'est donc appliquée depuis ses fondements à produire des familles explicites de bons codes, dont la dimension et la distance puisse être déterminées à l'avance. Cela a conduit à toute une "botanique" de codes, diversement utilisés en pratique.

Citons une autre borne, celle de Singleton. Suivant cette borne, tout code C linéaire voit ses paramètres $[n, k, d]$ vérifier $k + d \leq n + 1$. Pour la démontrer, considérons une *matrice de parité* de C, il s'agit d'une matrice H de taille $(n-k) \times n$ dans laquelle sont écrites $n-k$ formes linéaires qui s'annulent sur le code C : tout mot c du code vérifie $Hc = 0$. Le rang de la matrice H est $n-k$, et comme le code ne contient pas de mot de poids strictement inférieur à d, il n'y a pas de relations linéaires entre moins de d colonnes de H : $d - 1 \leq n - k$.

Nous nous contenterons de décrire les codes de Reed-Solomon qui sont optimaux pour la borne de Singleton et les codes de Goppa géométriques.

Codes de Reed-Solomon Un code de Reed-Solomon de dimension k et de longueur n est défini par la donnée de n éléments distincts $\alpha_1, \ldots, \alpha_n$ de $\mathbf{F}_q$, où $\mathbf{F}_q$ est un corps fini. Nous appellerons *points* ces éléments $\alpha_1, \ldots, \alpha_n$. Soit maintenant la fonction ev définie par

$$\begin{array}{rccl} \text{ev} : & \mathbf{F}_q[x] & \to & \mathbf{F}_q^n \\ & f(x) & \mapsto & \text{ev}(f(x)) = (f(\alpha_1), \ldots, f(\alpha_n)). \end{array}$$

où $\mathbf{F}_q[x]$ est l'algèbre des polynômes classique avec l'indéterminée canonique x. Le code de Reed-Solomon RS_k de dimension k est défini par $\alpha_1, \ldots, \alpha_n$, est

$$RS_k = \{\text{ev}(f(x));\ \deg(f(x)) < k\}\,.$$

Il est aisé de voir que sa dimension est k, et que sa distance minimale est $n-k+1$. En effet tout polynôme $f(x)$ de degré strictement inférieur à k a au plus $k-1$ zéros, donc le vecteur $\text{ev}(f(x))$ a au moins $n-k+1$ composantes non nulles. On a bien $k + d = n + 1$ pour les codes de Reed-Solomon, ce qui correspond à la borne de Singleton.

En divisant par n, le code de Reed-Solomon de taux de transmission R a donc une distance minimale aproximativement de $1 - R$. Notons aussi que si ces paramètres sont bons, on ne peut faire croître la longueur n en conservant un alphabet de taille fixée q, ce qui est un inconvénient majeur.

Codes de Goppa géométriques Les codes de Goppa géométriques ont été introduis par Goppa dans [1]. Ils sont une généralisation naturelle des codes de Reed-Solomon. Nous présentons ici une version simplifiée dite des codes "à un point".

Soit C une courbe algébrique irréductible lisse [3] définie sur $\mathbf{F}_q$. Soit $P_1, \ldots, P_n$ n points distincts rationnels de C et soit P_∞ un autre point distinct de $P_1, \ldots, P_n$.

3. Le lecteur ne connaissant pas la théorie des courbes algébriques peut sauter ce paragraphe.

Soit encore ev la fonction d'évaluation suivante :

$$\begin{array}{rccl} \text{ev}: & L(kP_\infty) & \to & \mathbf{F}_q^n \\ & f & \mapsto & \text{ev}(f) = (f(P_1), \ldots, f(P_n)), \end{array}$$

où $L(kP_\infty)$ désigne l'espace de fonctions de $\mathbf{F}_q(C)$ associé au diviseur kP_∞. Alors, de manière similaire aux codes de Reed-Solomon, on définit le code de Goppa géométrique $\Gamma(P_1, \ldots, P_n, kP_\infty)$ comme suit

$$\Gamma(P_1, \ldots, P_n, kP_\infty) = \{\text{ev}(f);\ f \in L(kP_\infty)\}\,.$$

Si on a choisi $k \geq 2g-2$, où g est le genre de C, alors le théorème de Riemann-Roch nous assure que la dimension de $\Gamma(P_1, \ldots, P_n, kP_\infty)$ est $k-g+1$. De même il est facile de prouver que la distance minimale d est telle $d \geq n-k$. Pour ces codes, on a $k+d = n-g+1$, et on voit que le "défaut" par rapport aux codes de Reed-Solomon et à la borne de Singleton est g, le genre de la courbe. En revanche, on peut faire croître la longueur de ces codes, *à alphabet fixé*, en augmentant le nombre de points des courbes sur lesquelles est fondée la construction.

En construisant une famille de courbes dont le genre ne croît pas trop vite, Tsfasmann, Vladut et Zink [7] ont montré l'existence de codes de distance relative δ et de taux de transmission R supérieur ou égal à $1-\delta-\frac{1}{\sqrt{q}-1}$. Ces codes, lorsque l'alphabet est de taille supérieure à 49, sont meilleurs que les codes aléatoires, c'est-à-dire qu'ils dépassent la borne de Varshamov-Gilbert, ce qui constitua une surprise de taille lors de leur découverte, les chercheurs en théorie des codes pensant que cette borne était optimale. Toutefois, dans le cas binaire ($q=2$), qui est le plus intéressant en pratique, on ne sait toujours pas si la borne de Varshamov-Gilbert est optimale ou s'il existe des codes dépassant cette borne.

3 Algorithmes de décodage

La problématique Le problème du décodage est une tâche difficile, pour laquelle les algorithmes naïfs présentent de très mauvaises complexités[4]. Par exemple la *recherche exhaustive* qui consiste à passer en revue tous les mots du code pour trouver le plus proche a une complexité en temps exponentielle en la longueur du code (pour peu que la dimension croisse linéairement avec la longueur du code).

Notons aussi que le problème du décodage n'est pas évident à formuler, et présente de nombreuses variantes dans la littérature. La communauté des chercheurs considère, hélas sans preuve, que le décodage des codes aléatoires est difficile. Cela signifie que si les codes aléatoires ont de bons paramètres, il est impossible de les décoder de manière efficace.

4. La *complexité (en temps)* d'un algorithme est ici le nombre d'opérations élémentaires nécessaires à son exécution. Pour les algorithmes de décodage, nous distinguerons la *performance* d'un algorithme, qui est une mesure de sa capacité à corriger des erreurs de poids plus ou moins élevé, de sa *complexité*, qui mesure son temps d'exécution.

Supposons la famille de codes à décoder fixée (Reed-Solomon, codes géométriques), il reste à définir proprement le problème. Suivant l'article de Madhu Sudan, nous en resterons aux définitions suivantes :

> **NCP** (Nearest Codeword Problem : problème du mot le plus proche) Il s'agit de trouver le mot de code le plus proche du mot reçu au sens de la métrique de Hamming.
>
> **LD** (List Decoding : décodage en liste) Une borne e est donnée. Le problème est de trouver *tous* (éventuellement aucun) les mots de code à distance e du mot reçu.
>
> **BDD** (Bounded Distance Decoding : décodage borné) Une borne e est donnée. Le problème est de trouver *un* mot parmi les mots de code à distance e du mot reçu (s'il en existe).
>
> **UD** (Unambiguous Decoding : décodage non ambigu) Ici on se donne $e = (d-1)/2$, où d est la distance minimale du code, et on cherche le mot de code à distance e du mot reçu (s'il existe) [5]

Classiquement, le problème étudié en théorie des codes est ce dernier problème (décodage non ambigu).

Décodage non ambigu (UD) Ce problème a été résolu pour toutes les classes de codes introduites ici, d'une manière efficace. Il est à noter que chacun des algorithmes est non trivial, et que le progrès le plus spectaculaire a été de réussir à décoder les codes de Goppa géométriques (voir le résumé [3]). Nous ne détaillerons pas ces algorithmes, mais notons que ces algorithmes ont en général une complexité quadratique ou presque quadratique [6] en la longueur, et chacun de ces algorithmes est construit "ad hoc" pour tel ou tel code, un algorithme "générique" de bonne complexité n'existant pas.

A titre d'exemple, nous présentons l'algorithme de Berlekamp-Welsh pour le décodage des codes de Reed-Solomon tels que nous les avons présentés. Le problème est le suivant : étant donnés des points $\alpha_1, \ldots, \alpha_n$ dans $\mathbf{F}_q$, des valeurs $y_1, \ldots, y_n$ dans $\mathbf{F}_q n$, et un entier $e < \frac{n-k}{2}$, trouver tous les polynômes $f(x)$ dans $\mathbf{F}_q[x]$ de degré strictement inférieur à k tels que $f(\alpha_i) = y_i$ pour au moins $n-e$ valeurs de i. Pour cette valeur de e, on sait qu'il y a au plus une solution $f(x)$.

Soit donc $f(x)$ le polynôme solution du problème posé, et onsidérons le polynôme unitaire $E(x)$ qui est de degré e (e étant le poids de l'erreur), tel que $E(\alpha_i) = 0$ s'il y a une erreur à la position i. Alors, à coup sûr, on a, pour tout i, $y_i = f(\alpha_i)$ ou $E(\alpha_i) = 0$, ce qui se traduit algébriquement en $E(\alpha_i)y_i = E(\alpha_i)f(\alpha_i)$, pour tout i.

L'algorithme de Berlekamp-Welsh consiste en les deux étapes suivantes : trouver deux polynômes $E(x)$ et $N(x)$ de degrés respectifs au plus e et $k+e-1$,

5. Signalons, pour être complet, une grande classe de codes et d'algorithmes de décodage, très performants en pratique, qui est celle des turbo-codes et des codes LDPC, conjointement avec les algorithmes de décodage itératifs. Ces algorithmes de décodages ne font apparaître que marginalement la notion de distance minimale.

6. $O(n^{7/3})$

tels que $E(\alpha_i)y_i = N(\alpha_i)$, pour tout i. On voit qu'il s'agit d'un problème d'algébre linéaire, dont les inconnues sont les coefficients de $E(x)$ et de $N(x)$. En particulier, on est sûr que ce système a une solution quand le nombre d'inconnues est inférieur au nombre d'équations, c'est-à-dire quand $e+k+e < n$, ce qui donne la condition $e < \frac{n-k}{2}$, ce qui est bien la capacité de correction des codes de Reed-Solomon. La deuxième étape est de retourner le résultat $N(x)/E(x)$ (qui est bien un polynôme).

Décodage en liste (LD) Le problème du décodage en liste se pose dès que l'on veut décoder e erreurs avec $e > d/2$, où d est la distance minimale du code à décoder. En effet, il n'y a plus unicité du mot de code à distance e, et il devient alors nécessaire de retourner une liste de candidats.

Le principal progrès dû à Madhu Sudan dans son article [5], concerne le décodage des codes de Reed-Solomon, pour lesquels il a produit un algorithme capable de décoder bien au-delà de la capacité de correction $\lfloor \frac{d-1}{2} \rfloor$ du code. Le problème est toujours le même : étant donnés des points $\alpha_1, \ldots, \alpha_n$ dans $\mathbf{F}_q$, des valeurs $y_1, \ldots, y_n$ dans $\mathbf{F}_q$, et un entier e, trouver tous les polynômes $f(x)$ dans $\mathbf{F}_q[x]$ de degré inférieur à k tels que $f(\alpha_i) = y_i$ pour au moins $n - e$ valeurs de i. Mais dans ce cas précis, on autorise e à être plus grand que $\frac{n-k}{2}$, donc il y a plusieurs solutions possibles. L'algorithme est une généralisation de l'algorithme de Berlekamp-Welsh, et l'idée est la suivante. L'algorithme de Berlekamp-Welsh décrit ci-dessus cherche en fait un polynôme $Q(x, y) = N(x) - yE(x)$ de degré 1 en Y tel que $Q(x_i, y_i) = 0$ pour tout i, avec les conditions $\deg N(x) \leq k + e$ et $\deg E(x) \leq e$.

L'idée de M. Sudan est de chercher un polynôme $Q(x, y) = \sum_i Q_i(x)y^i$ *de degré supérieur à* 1 *en* y tels que $\deg Q_i(x) < n - e - i(k - 1)$, et tel que $Q(x_i, y_i) = 0$ pour tout i. Alors, tout polynôme $f(x)$ solution du problème de décodage vérifie $Q(x, f(x)) = 0$. En effet, le polynôme $Q(x, f(x))$ est degré strictement inférieur à $n-e$ par construction de $Q(x, y)$, de plus comme $f(x_i) = y_i$ pour au moins $n - e$ valeurs de i et que $Q(x_i, y_i) = 0$ pour tout i, on a que le polynôme $Q(x, f(x))$ a plus de racines que son degré, donc il est identiquement nul.

En conséquence, l'algorithme de décodage de M. Sudan se déroule en deux étapes :

1. Trouver un polynôme $Q(x, y)$ satisfaisant les conditions ci-dessus.
2. Trouver les facteurs $y - f(x)$ de $Q(x, y)$.

La première étape est encore une étape d'algèbre linéaire, où l'on cherche $Q(x, y) = \sum_i Q_i(x)y^i$ tel que $(x_i, y_i) = 0$ pour tout i, avec les conditions précédentes sur les degrés des $Q_i(x)$, et les inconnues sont les coefficients de $Q(x, y)$. Pour être assurés d'avoir une solution, on doit avoir $N_Q > n$ où N_Q est le nombre de monômes apparaissant dans $Q(x, y)$. Notons que le degré du polynôme $Q(x, y)$ est au plus $\lfloor \frac{n-e}{k-1} \rfloor$: c'est un majorant du nombre de solutions à l'équation $Q(x, f(x)) = 0$, donc du nombre de solutions à notre problème.

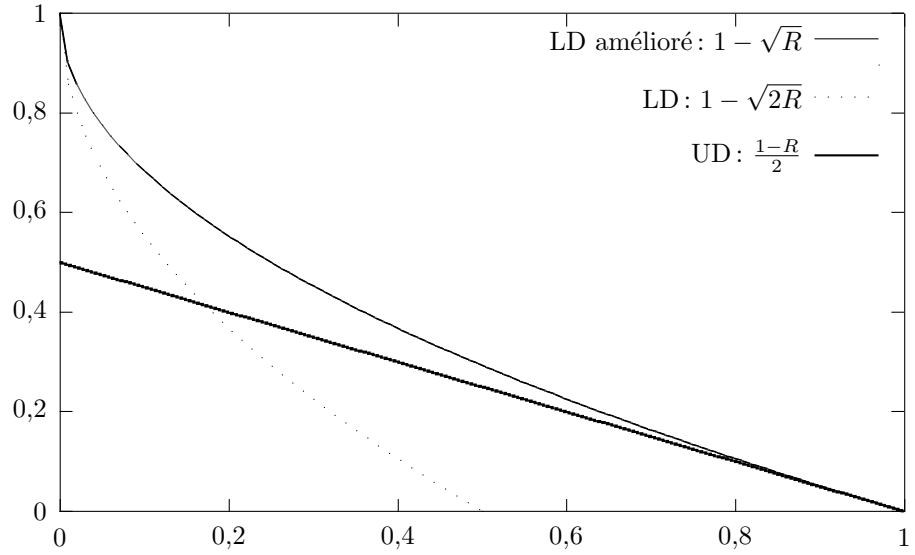

FIG. 1: *Performances des algorithmes de Berlekamp-Welsh, Sudan et Guruswami-Sudan. En abscisse : le taux R de transmission ; en ordonnée : la capacité de correction.*

Un calcul du nombre de termes de $Q(x, y)$ et la condition $N_q > n$ donnent (grossièrement) la relation suivante en e :

$$e < n - \sqrt{2kn} \tag{1}$$

Cette borne est à comparer avec la borne du décodage non ambigu $e < \frac{n-k}{2}$. Elle s'exprime mieux en divisant par n, où nous avons $\epsilon < 1-\sqrt{2R}$, avec $\epsilon = e/n$ er $R = k/n$, et nous comparons les performances sur la figure 1. La première remarque est que la capacité de correction de l'algorithme de Sudan n'est pas toujours plus élevée que celle de l'algorithme de Berlekamp-Welsh, notamment pour les codes de grand rendement k/n. La deuxième remarque est que lorsque le rendement du code est proche de zéro, alors le taux d'erreurs toléré approche 1.

La deuxième étape de l'algorithme de M. Sudan consiste, après avoir trouvé le polynôme $Q(x, y)$ à trouver les solutions $f(x)$ de l'équation $Q(x, f(x)) = 0$, c'est-à-dire de trouver les facteurs $y - f(x)$ de $Q(x, y)$. Notons rapidement qu'il existe un algorithme de complexité polynômiale pour factoriser les polynômes à plusieurs variables sur un corps fini. M. Sudan en conclut donc que son algorithme est de complexité polynômiale. Cela n'est pas suffisant en pratique quand on souhaite obtenir la plus grande efficacité, et beaucoup de chercheurs s'attachent à trouver les meilleurs algorithmes pour accomplir les deux étapes

de l'algorithme de Sudan, afin de le rendre complètement effectif de telle sorte qu'on puisse le programmer dans des circuits électroniques.

Décodage en liste amélioré V. Guruswami et M. Sudan ont rapidement proposé un algorithme étendant la capacité de correction à la borne $1 - \sqrt{R}$, qui est toujours meilleure que la borne classique $\frac{1-R}{2}$ (cf. figure 1). Sans entrer dans les détails de l'article [2], nous faisons la remarque suivante. Si $f_1(x)$ et $f_2(x)$ sont deux solutions au problème de décodage, alors il peut y avoir des indices i tels que l'on ait $y_i = f_1(x_i) = f_2(x_i)$. Sachant que $y - f_1(x)$ et $y - f_2(x)$ divisent tous deux $Q(x, y)$, alors ce dernier polynôme présentera une multiplicité d'ordre au moins deux au point (x_i, y_i). C'est cette remarque qui est à la base de l'algorithme amélioré : on cherche maintenant un polynôme $Q(x, y)$ tel que $Q(x_i, y_i) = 0$ avec une certaine multiplicité r. Le degré de $Q(x, y)$ étant bien choisi ainsi que la multiplicité, alors on aura, pour tout polynôme $f(x)$ solution du problème de décodage, $Q(x, f(x)) = 0$. Une optimisation des paramètres auxiliaires (degré de $Q(x, y)$, ordre r de multiplicité), conduit à la borne $1 - \sqrt{R}$.

Pour conclure cette section, nous citons [2, 4] pour indiquer que ces algorithmes (Sudan, Guruswami-Sudan) se généralisent facilement aux codes géométriques. De plus, d'une manière surprenante, ces généralisations conduisent à des algorithmes conceptuellement plus simples que ceux déjà inventés pour le décodage classique des codes géométriques.

4 Conclusion

La découverte d'un algorithme de décodage des codes de Reed-Solomon avec un tel pouvoir de correction a relancé l'intérêt pratique de ces codes, ainsi que des codes géométriques, puisque leur *performance* s'en trouve grandement améliorée. Si la question se pose de savoir en pratique comment choisir une solution dans la liste des solutions proposées par l'algorithme, force est de constater que l'algorithme retourne une unique solution avec une probabilité quasi égale à un, ce qui est fort intéressant. Reste maintenant à améliorer l'implémentation de cet algorithme pour être performant dans les applications.

Les algorithmes de Sudan et Guruswami-Sudan ont déjà eu des applications en cryptologie (pour cryptanalyser un algorithme de chiffrement) et aussi dans le domaine de la protection des droits d'auteurs. Dans les deux cas, c'est le haut pouvoir de correction, proche de 1 à taux de transmission proche de zéro, qui est utilisé. Nul doute que, du point de vue théorique, cet algorithme aura encore de nombreuses applications dans diverses branches des mathématiques appliquées et deviendra un classique parmi les grands algorithmes de l'informatique.

Références

[1] V.D. Goppa. Codes associated with divisors. *Problems of Information Transmission*, 12(1):22–27, 1977.

[2] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and Algebraic-Geometric codes. *IEEE Transactions on Information Theory*, 45:1757–1767, 1999.

[3] Tom Hoholdt and Ruud Pellikaan. On the decoding of algebraic geometry codes. *IEEE Transactions on Information Theory*, 41(6), 1995.

[4] M. Amin Shokrollahi and Hal Wasserman. Decoding algebraic geometric codes beyond the error-correction bound. *IEEE Transactions on Information Theory*, 45:432–437, 1999.

[5] Madhu Sudan. Decoding of Reed-Solomon codes beyond the error-correction boud. *Journal of Complexity*, 13, 1997.

[6] Madhu Sudan. Coding theory: Tutorial and survey. In *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science*, pages 36–53, 2001.

[7] M. A. Tsfasmann, S. G. Vlădut, and T. Zink. Modular curves, shimura curves, and goppa codes better than varshamov-gilbert bound. *Math. Nachr.*, "109":21–28, "1982".

[8] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy. Proof verification and hardness of approximation problems. *Journal of the ACM*, 45(3):501-555, 1998.