Satisfiabilité pour le problème du décodage par syndrome à poids élevé

Carl Berton, Sami Cherif, Claire Delaplace

Université de Picardie Jules Verne, Laboratoire MIS UR 4290, Amiens, France

{carl.berton;sami.cherif;claire.delaplace}@u-picardie.fr

Résumé

Le problème du décodage par syndrome à poids élevé (LWSD, pour Large Weight Syndrome Decoding) est un problème fondamental en théorie des codes. Il consiste à déterminer si un code linéaire donné admet un vecteur de poids de Hamming élevé associé à un syndrome spécifique. Le LWSD est une variante du problème classique de décodage par syndrome qui, à l'inverse, recherche une solution de petit poids de Hamming pour un système linéaire défini sur le corps binaire $\mathbb{F}_2$. Dans cet article, nous étudions une généralisation de ce problème au cas d'un corps fini premier $\mathbb{F}_Z$, désignée sous le nom de LWZSD. Nous proposons plusieurs modélisations utilisant des formules de satisfiabilité booléenne (SAT) et comparons l'efficacité de nos approches en utilisant des solveurs de l'état de l'art.

Mots-clés

Décodage par syndrome à poids élevé, Satisfiabilité, Cryptographie

Abstract

The Large Weight Syndrome Decoding problem LWSD is a fundamental problem in coding theory. It consists in determining whether a given linear code admits a high Hamming weight vector associated with a specific syndrome. LWSD is a variant of the classical syndrome decoding problem, which conversely seeks a low Hamming weight solution for a linear system defined over the binary field $\mathbb{F}_2$. In this paper, we investigate a generalization of this problem to the case of a prime finite field $\mathbb{F}_Z$, referred to as LWZSD. We propose several models using Boolean Satisfiability (SAT) formulas and compare the efficiency of our approaches using state-of-the-art solvers.

Keywords

Large Weight Syndrome Decoding, Satisfiability, Cryptography

1 Introduction

La sécurité de l'information est devenue une préoccupation stratégique majeure pour les états, les entreprises et les citoyens, dans un contexte où la multiplication des échanges numériques rend l'usage de la cryptographie essentiel pour garantir la confidentialité, l'authenticité et l'intégrité des données sensibles. Dans ce cadre, la cryptographie symétrique permet une communication confidentielle entre au moins deux parties partageant une clé secrète commune, en utilisant un algorithme de chiffrement symétrique (tel que AES [13]) pour protéger les communications et permettre l'échange sécurisé de clés ou la signature électronique. Cependant, l'émergence des ordinateurs quantiques remet en question la robustesse des schémas classiques, conduisant l'Institut National des Standards et de la Technologie américain (NIST) à lancer un processus de standardisation pour des algorithmes post-quantiques basés sur des schémas asymétriques. Ces derniers s'appuient sur des paires de clés et des problèmes difficiles, tels que le décodage fondé sur les codes, pour assurer une sécurité robuste face aux adversaires tant classiques que quantiques [10].

Parallèlement, les outils de raisonnement automatisé sont devenus de puissants instruments d'analyse et d'optimisation en cryptographie, particulièrement pour affiner les attaques contre les schémas symétriques classiques [12, 21]. Pourtant, de nombreux algorithmes restent peu étudiés sous cet angle et l'application de ces outils aux schémas asymétriques, et particulièrement à ceux résistants aux attaques quantiques, demeure limitée malgré l'existence de problèmes qui se prêtent naturellement à de telles approches. À cet égard, le problème de satisfaisabilité propositionnelle (SAT) s'impose comme un outil de raisonnement automatisé performant, capable d'aider à l'analyse et à l'exploration de constructions cryptographiques post-quantiques. Le problème SAT consiste à déterminer, pour une formule donnée en Forme Normale Conjonctive (CNF), s'il existe une affectation des variables qui la satisfait [8]. Il constitue un problème de décision fondamental en informatique et en intelligence artificielle, utilisé pour résoudre un large éventail de problèmes dans divers domaines tels que la cryptographie [17, 29, 32], la vérification matérielle et logicielle [16], et la planification [24]. Il fut le premier problème démontré comme étant NP-complet [11].

Dans cet article, nous nous intéressons aux problèmes de décodage. Pour assurer une transmission fiable de l'information sur des canaux bruités, les codes correcteurs d'erreurs, et plus spécifiquement les codes linéaires binaires, sont utilisés pour ajouter de la redondance afin de détecter et de corriger les erreurs [26]. Alors que les codes en théorie de l'information sont généralement conçus pour rendre

ce problème facile à résoudre afin de fluidifier les communications, le décodage d'un code linéaire binaire dans le cas général est un problème NP-complet [4]. Les cryptographes s'intéressent à ce problème depuis les années 1970 [19], lorsque McEliece a introduit le premier schéma de chiffrement à clé publique dont la sécurité repose sur la difficulté du décodage d'un code linéaire binaire. Avec l'avènement de la cryptographie post-quantique, la cryptographie fondée sur les codes a connu un essor important ces dernières années, avec de nombreux efforts tant dans les constructions que dans les attaques [20, 15, 22].

Un problème d'un intérêt particulier pour les cryptographes est le décodage par syndrome, qui consiste à retrouver le bruit à partir de la matrice de contrôle de parité du code [3]. En pratique, ce problème revient à résoudre un système linéaire sous-déterminé modulo 2, sous la contrainte que la solution doit avoir un poids de Hamming faible. Des travaux récents ont exploré la modélisation de ce problème à l'aide de formules CNF et XNF (XOR-CNF) [5, 6] et nous poursuivons ces recherches en étudiant le problème du décodage par syndrome à poids élevé (ou LWSD). Contrairement au cas binaire classique, nous considérons ici une généralisation sur un corps fini premier $\mathbb{F}_Z$ (LWZSD). Le passage à $\mathbb{F}_Z$ empêche l'utilisation directe de formules XNF. Nous explorons quatre encodages propositionnels distincts, basés sur des représentations one-hot, unaires et binaires. Ces approches reformulent les contraintes linéaires modulo Z en utilisant des variables auxiliaires et des contraintes Pseudo-Booléennes, tout en intégrant la contrainte de poids de l'erreur. Nous comparons ces différents modèles pour analyser leurs performances respectives avec des solveurs SAT de l'état de l'art.

Cet article est organisé comme suit. Dans la section 2, nous rappelons les concepts fondamentaux nécessaires à notre étude, incluant le problème LWSD et le problème SAT. Nous introduisons nos encodages CNF pour ce problème dans la section 3. Nous analysons ensuite les résultats expérimentaux obtenus pour comparer ces différentes approches dans la section 4. Enfin, nous concluons et discutons des perspectives de travaux futurs dans la section 5.

2 Préliminaires

2.1 Satisfaisabilité

Soit $X = \{x_1, x_2, \ldots, x_n\}$ un ensemble de variables booléennes pouvant prendre les valeurs Vrai ($\top$ ou 1) ou Faux ($\bot$ ou 0). Un littéral est soit une variable $x \in X$, soit sa négation $\overline{x}$. Une formule CNF ϕ est une conjonction de clauses $\phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, où chaque clause C_j est une disjonction ($\vee$) de littéraux. Une affectation $\alpha : X \to \{\text{Vrai, Faux}\}$ est une fonction associant à chaque variable de X une valeur de vérité. Ainsi, une clause est satisfaite par une affectation α si au moins l'un de ses littéraux est satisfait par α. Une formule CNF ϕ est satisfaite par une affectation α si toutes ses clauses sont satisfaites par α. Dans ce cas, la formule est dite satisfaisable et α est un modèle de ϕ.

Le problème de satisfaisabilité propositionnelle (SAT) consiste à déterminer si une formule CNF donnée est satisfaisable. Les solveurs SAT modernes sont extrêmement efficaces et parviennent à résoudre des formules comportant un grand nombre de variables et de clauses dans un temps remarquablement court, malgré la difficulté théorique établie du problème. En effet, ces solveurs s'appuient sur l'algorithme CDCL (*Conflict-Driven Clause Learning*) [28] et emploient des mécanismes puissants tels que l'analyse de conflits, le retour arrière non-chronologique (backjumping) et des heuristiques de branchement dédiées [8].

Dans nos modèles, nous utiliserons également des contraintes Pseudo-Booléennes (PB) qui imposent une borne sur une somme pondérée de littéraux $\left(\sum_{i=1}^{h} a_i \times l_i\right) \circ k$ où $a_i, k \in \mathbb{N}$ et $\circ \in \{\leq, =, \geq\}$. Dans le cas où tous les poids sont fixés à 1 (et peuvent donc être omis), de telles contraintes sont plus couramment appelées contraintes de cardinalité. Les contraintes PB et de cardinalité sont typiquement utilisées dans les solveurs pour imposer des bornes pertinentes durant la recherche et peuvent être efficacement encodées sous forme clausale [25].

2.2 Décodage par syndrome à poids élevé

Le problème du décodage par syndrome à poids élevé (LWSD) est une variante fondamentale du problème de décodage par syndrome qui, étant donné un code linéaire binaire, consiste à déterminer s'il existe un mot de code ayant un poids de Hamming élevé. La généralisation du LWSD au cas d'un corps fini premier $\mathbb{F}_Z$ conduit au LWZSD. Nous nous intéressons ainsi à déterminer, pour un code linéaire donné défini sur $\mathbb{F}_Z$, l'existence d'un mot de code à grand poids.

Définition 1 (Poids de Hamming) *Le poids de Hamming* $\mathrm{wt}(\mathbf{e})$ *d'un vecteur* $\mathbf{e} = (e_0, \ldots, e_{n-1}) \in \mathbb{F}_Z^n$ *est défini comme le nombre de ses composantes non nulles, c'est-à-dire :*

$$\mathrm{wt}(\mathbf{e}) := \left|\{\, j \in \{0, \ldots, n-1\} \mid e_j \neq 0 \,\}\right|$$

Définition 2 (LWZSD) *Soit* $\mathbb{F}_Z$ *le corps fini à* Z *éléments, où* Z *est premier, et soient* n*,* k*, et* t *des entiers tels que* $k \leq n$ *et* $t \leq n$*. Une instance du problème* $\mathsf{LWZSD}(n, k, t)$ *consiste en une matrice de contrôle de parité* $\mathbf{H} \in \mathbb{F}_Z^{(n-k)\times n}$ *et un vecteur* $\mathbf{s} \in \mathbb{F}_Z^{n-k}$ *(appelé le syndrome). Une solution à ce problème est un vecteur* $\mathbf{e} \in \mathbb{F}_Z^n$ *tel que* $\mathrm{He}^\top \equiv \mathrm{s}^\top$ *avec* $wt(e) \geq t$*.*

Exemple 1 (Décodage par syndrome à poids élevé) *Considérons un code linéaire sur* $\mathbb{F}_3$ *avec les paramètres* $n = 5$*,* $k = 2$*, et un seuil de poids minimal* $t = 3$*. La matrice de contrôle de parité* $H \in \mathbb{F}_3^{3\times 5}$ *est donnée sous forme systématique* $[I_3 \mid A]$ *comme suit :*

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 2 & 1 \\ 0 & 0 & 1 & 1 & 2 \end{pmatrix}$$

Considérons le syndrome $s = (0, 1, 1)^\top \in \mathbb{F}_3^3$*. Nous cherchons un vecteur* $e \in \mathbb{F}_3^5$ *vérifiant* $He^\top \equiv s^\top \pmod 3$ *avec* $\mathrm{wt}(e) \geq t = 3$*.*

1. ***Proposer un vecteur candidat*** e ***:*** *Contrairement au décodage par syndrome classique qui recherche un vecteur d'erreur de poids faible, il s'agit ici de trouver une solution de poids élevé. Considérons le candidat :*

$$e = (1, 0, 2, 1, 2)^\top$$

qui est de poids $\mathrm{wt}(e) = 4 \geq 3$*, satisfaisant ainsi la contrainte de poids.*

2. ***Vérifier l'équation de syndrome :*** *On calcule* $He^\top \pmod 3$ *:*

$$He^\top = \begin{pmatrix} 6 \\ 4 \\ 7 \end{pmatrix} \equiv \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \pmod 3$$

On obtient bien $He^\top \equiv s^\top \pmod 3$*, ce qui confirme que* e *est une solution valide.*

Le vecteur $e = (1, 0, 2, 1, 2)^\top$ *satisfait les deux conditions du problème* $\mathsf{LWZSD}(5, 2, 3)$ *: il est cohérent avec le syndrome* s *et son poids* $4 \geq t = 3$*.*

Si l'on note $H_{i,j}$ le coefficient à la i-ème ligne et j-ème colonne de la matrice de contrôle H, et s_i la i-ème composante du syndrome, $\mathsf{LWZSD}(n, k, t)$ peut alors être défini comme suit :

$$E_i : \sum_{\substack{0 \leq j \leq n-1 \\ H_{i,j} \neq 0}} H_{i,j} \cdot e_j \equiv s_i \mod Z, \quad \forall\, 0 \leq i \leq n-k-1 \tag{1a}$$

$$\mathrm{wt}(\mathbf{e}) \geq t \tag{1b}$$

L'équation (1a) exige que, pour chaque E_i, la somme pondérée des composantes du vecteur d'erreur $\mathbf{e}$ soit congrue au syndrome s_i modulo Z, tandis que l'équation (1b) garantit que $\mathbf{e}$ a un poids de Hamming supérieur ou égal à t.

Le schéma de signature Wave [14] illustre l'une des premières constructions cryptographiques reposant explicitement sur la difficulté du décodage par syndrome sur $\mathbb{F}_3$ pour des mots d'erreur à grand poids. Bricout et al. [9] ont étudié ce problème en détail, soulignant que la structure combinatoire du décodage ternaire à grand poids diffère significativement de celle du décodage binaire classique. Leur analyse exploite des algorithmes basés sur les techniques de décodage par ensemble d'information (ISD) [23, 31, 2] et montre que le cas du poids élevé constitue un cadre pertinent pour les applications cryptographiques. En effet, ce problème est considéré comme une alternative crédible face à la menace quantique, car les algorithmes de Shor [27] ne s'appliquent pas à la structure des codes correcteurs d'erreurs.

Plus récemment, le décodage par syndrome classique a été étudié sous l'angle de la satisfaisabilité propositionnelle. Des travaux antérieurs ont abouti à des modèles SAT sous forme de formules CNF et XNF (XOR-CNF) [5, 6]. Cependant, ces travaux se focalisent sur le cas $\mathbb{F}_2$ avec un poids de Hamming borné, ce qui permettait d'exploiter naturellement la structure XOR d'origine du problème pour introduire des modèles ensuite traduits en CNF. À l'inverse, le problème LWZSD implique des corps $\mathbb{F}_Z$ où $Z > 2$, ce qui limite l'usage direct et naturel des contraintes XOR. Par conséquent, les modèles XNF spécifiques au cas binaire ne sont pas directement transposables à notre cadre, nécessitant l'étude d'encodages CNF dédiés, qui seront présentés dans la section suivante.

3 Modélisation du décodage par syndrome à poids élevé

3.1 Variables et notations

Dans cette section, nous introduisons les variables, ensembles et notations qui seront utilisés dans les sections suivantes pour décrire les différents modèles. Nous notons E_i la i-ème équation du système, pour $0 \leq i \leq n-k-1$. Pour simplifier les notations, posons $m = n-k-1$. Pour encoder chaque $e_j \in \{0, \ldots, Z-1\}$, nous introduisons les variables booléennes $e_{j,1}, e_{j,2}, \ldots, e_{j,Z-1}$, pour $0 \leq j \leq n-1$, où $e_{j,k} = 1 \Leftrightarrow e_j \geq k$. En d'autres termes, si $e_j = v$, alors $e_{j,1} = \cdots = e_{j,v} = 1$ et $e_{j,v+1} = \cdots = e_{j,Z-1} = 0$. Pour chaque équation E_i, nous définissons l'ensemble V_{E_i} comme l'ensemble des indices des variables apparaissant dans E_i. Plus formellement, pour tout $i \in \{0, \ldots, m\}$, nous avons :

$$V_{E_i} = \{j \in \{0, \ldots, n-1\} \mid H_{i,j} \neq 0\}$$

Pour chaque équation E_i, nous considérons la somme pondérée des variables qui y apparaissent, à savoir $\sum_{j \in V_{E_i}} H_{i,j} e_j$. Dans ce contexte, nous introduisons pour chaque équation un ensemble noté K_{E_i}, correspondant à l'ensemble des valeurs entières que cette somme peut prendre tout en satisfaisant la congruence imposée par s_i. Pour caractériser cet ensemble, nous commençons par déterminer une borne inférieure. Puisque le vecteur d'erreur e doit satisfaire $\mathrm{wt}(e) \geq t$, ce qui signifie qu'il contient au moins t composantes non nulles, cela implique qu'au plus $n-t$ composantes de e sont nulles parmi les variables participantes. Par conséquent, nous posons $m_i = \max\big(0,\ |V_{E_i}| - (n-t)\big)$, ce qui représente le nombre minimal de composantes non nulles devant figurer dans E_i. Nous définissons ensuite $V_{E_i}^{\min}$ comme l'ensemble des indices correspondant aux m_i plus petits coefficients $H_{i,j}$ parmi $j \in V_{E_i}$. Ainsi, nous pouvons définir formellement la borne inférieure $v_{\min}^{(i)} \in \mathbb{N}$ par :

$$v_{\min}^{(i)} = \min\Big\{v \geq \sum_{j \in V_{E_i}^{\min}} H_{i,j} \Big|\ v \equiv s_i \pmod Z\Big\}$$

En d'autres termes, $v_{\min}^{(i)}$ est la plus petite valeur entière congrue à s_i modulo Z et supérieure ou égale à la somme minimale réalisable. De même, nous pouvons déterminer une borne supérieure pour l'ensemble K_{E_i}. En effet, chaque variable e_j prend une valeur entière entre 0 et $Z-1$. Ainsi, la somme pondérée $\sum_{j \in V_{E_i}} H_{i,j} e_j$ atteint sa valeur maximale lorsque toutes les variables e_j sont égales à $Z-1$. Pour que cette valeur appartienne à l'ensemble K_{E_i}, elle

doit également respecter la congruence imposée par s_i modulo Z. Nous définissons donc formellement $v_{\max}^{(i)} \in \mathbb{N}$ par :

$$v_{\max}^{(i)} = \max\Big\{v \leq (Z-1) \sum_{j \in V_{E_i}} H_{i,j} \Big|\, v \equiv s_i \pmod{Z}\Big\}$$

En d'autres termes, $v_{\max}^{(i)}$ est la plus grande valeur entière inférieure ou égale à la somme maximale possible et congrue à s_i modulo Z. Nous pouvons alors définir l'ensemble des valeurs possibles que la somme pondérée $\sum_{j \in V_{E_i}} H_{i,j} e_j$ peut prendre dans l'équation E_i, en tenant compte de la parité imposée par s_i dans $\mathbb{F}_Z$ comme suit :

$$K_{E_i} = \Big\{v \in \{\, v_{\min}^{(i)}, \ldots, v_{\max}^{(i)}\} \Big|\, v \equiv s_i \pmod{Z}\Big\}.$$

D'autres variables seront introduites dans les sections suivantes, leur sémantique dépendant du contexte spécifique de chaque modèle.

3.2 Encodage one-hot

Pour représenter la valeur prise par la somme des variables dans chaque équation, nous introduisons des variables $x_{i,v}$ pour tout $(i,v) \in \{0,\ldots,m\} \times K_{E_i}$, qui sont affectées à Vrai si et seulement si la somme des variables de l'équation E_i est égale à v, c'est-à-dire $\sum_{j \in V_{E_i}} H_{i,j}\, e_j = v$. Le système $\mathsf{LWZSD}(n,k,t)$ peut alors s'écrire sous la forme suivante :

$$\sum_{j \in V_{E_i}} \sum_{c=1}^{Z-1} H_{i,j}\, e_{j,c} + \sum_{v \in K_{E_i}} v\, \bar{x}_{i,v} = \sum_{v \in K_{E_i}} v, \qquad \forall\, 0 \leq i \leq m \quad \text{(2a)}$$

$$\sum_{v \in K_{E_i}} x_{i,v} = 1, \quad \forall\, 0 \leq i \leq m \quad \text{(2b)}$$

$$\bar{e}_{j,v} \vee e_{j,v-1}, \quad \forall\, 0 \leq j \leq n-1,\ 2 \leq v \leq Z-1 \quad \text{(2c)}$$

$$\sum_{j=0}^{n-1} e_{j,1} \geq t \quad \text{(2d)}$$

La contrainte (2a), écrite sous forme de contrainte PB, impose que la somme pondérée des variables d'erreur e_j dans l'équation E_i soit exactement égale à la valeur v choisie par les variables $x_{i,v}$. La contrainte (2b) garantit qu'une seule valeur de somme est sélectionnée pour chaque équation. La contrainte (2c) assure que si $e_{j,v} = 1$, alors tous les $e_{j,c}$ avec $c < v$ sont également égaux à 1. Enfin, la contrainte (2d) exige que le vecteur d'erreur e possède au moins t composantes non nulles.

Proposition 1 *La formule propositionnelle induite par les contraintes (2) est satisfaisable si et seulement si le système* $\mathsf{LWZSD}(n,k,t)$ *admet une solution.*

Preuve. Tout d'abord, nous établissons que les contraintes (2a), (2b) et (2c) sont équivalentes à l'équation de syndrome (1a). Soit $i \in \{0,\ldots,m\}$, en notant que $x = 1 - \bar{x}$, la contrainte (2a) peut s'écrire :

$$\sum_{j \in V_{E_i}} H_{i,j} \sum_{c=1}^{Z-1} e_{j,c} + \sum_{v \in K_{E_i}} v\,(1 - x_{i,v}) = \sum_{v \in K_{E_i}} v.$$

Cette égalité est équivalente à :

$$\sum_{j \in V_{E_i}} H_{i,j} \left(\sum_{c=1}^{Z-1} e_{j,c} \right) = \sum_{v \in K_{E_i}} v \cdot x_{i,v}$$

Sous la contrainte (2c), les variables $e_{j,c}$ constituent un encodage de la variable e_j, avec $e_j = \sum_{c=1}^{Z-1} e_{j,c}$. Nous obtenons ainsi :

$$\sum_{j \in V_{E_i}} H_{i,j} \cdot e_j = \sum_{v \in K_{E_i}} v \cdot x_{i,v}.$$

De plus, la contrainte (2b) impose que, pour chaque équation E_i, les variables $x_{i,v}$ forment un encodage one-hot des valeurs de somme possibles modulo Z. Compte tenu de la définition de l'ensemble K_{E_i}, cette contrainte est donc équivalente à l'équation de syndrome (1a) du problème $\mathsf{LWZSD}(n,k,t)$.
Il reste à considérer la contrainte de poids. Par construction, la variable $e_{j,1}$ est vraie si et seulement si $e_j \geq 1$. Ainsi, nous avons :

$$\mathrm{wt}(\mathbf{e}) = \big|\{\, j \in \{0,\ldots,n-1\} \mid e_j \neq 0\,\}\big| = \sum_{j=0}^{n-1} e_{j,1}.$$

La contrainte (2d) est donc équivalente à la contrainte (1b). Ceci établit l'équivalence entre les solutions du système $\mathsf{LWZSD}(n,k,t)$ et les affectations satisfaisant les contraintes (2). □

3.3 Encodage unaire

Nous pouvons également observer que, dans le cas général, les éléments de l'ensemble K_{E_i} augmentent par incréments de Z. Pour tirer parti de cette structure, nous redéfinissons la sémantique des variables $x_{i,v}$ comme suit : $x_{i,v} = 1 \Leftrightarrow \sum_{j \in V_{E_i}} H_{i,j}\, e_j \geq v$, $\forall v \in K_{E_i}$. Le système peut alors s'écrire ainsi :

$$\sum_{j \in V_{E_i}} \sum_{c=1}^{Z-1} H_{i,j}\, e_{j,c} + \sum_{v \in K_{E_i} \setminus \{v_{\min}^{(i)}\}} Z \cdot \bar{x}_{i,v} = v_{\max}^{(i)}, \qquad \forall\, 0 \leq i \leq m \quad \text{(3a)}$$

$$\bar{x}_{i,v} \vee x_{i,v-Z}, \quad \forall 0 \leq i \leq m,\ v \in K_{E_i} \setminus \{v_{\min}^{(i)}\} \quad \text{(3b)}$$

$$\bar{e}_{j,v} \vee e_{j,v-1}, \quad \forall 0 \leq j \leq n-1,\ 2 \leq v \leq Z-1 \quad \text{(3c)}$$

$$\sum_{j=0}^{n-1} e_{j,1} \geq t \quad \text{(3d)}$$

La contrainte (3a) exprime, sous forme PB, la relation entre la somme pondérée des variables d'erreur e_j dans l'équation E_i et la valeur sélectionnée parmi les variables $x_{i,v}$. La contrainte (3b) garantit que si $\sum_{j \in V_{E_i}} e_j \geq v$ est vrai,

alors $\sum_{j\in V_{E_i}} e_j \geq v - Z$ est également vrai, assurant ainsi la cohérence de l'encodage. La contrainte (3c) encode la représentation unaire des variables e_j. Elle impose que, dès que $e_{j,v} = 1$, toutes les variables $e_{j,c}$ avec $c < v$ sont également fixées à 1. Enfin, la contrainte de poids (3d) garantit que le vecteur d'erreur e possède au moins t composantes non nulles.

Proposition 2 *La formule propositionnelle induite par les contraintes* (3) *est satisfaisable si et seulement si le système* $\mathsf{LWZSD}(n, k, t)$ *admet une solution.*

Preuve. Nous démontrons que, sous les contraintes (3b) et (3c), la contrainte Pseudo-Booléenne (3a) est équivalente à l'équation de syndrome (1a) du système $\mathsf{LWZSD}(n, k, t)$. En notant que $\overline{x} = 1 - x$, la contrainte (3a) peut être réécrite comme suit :

$$\sum_{j\in V_{E_i}} \sum_{c=1}^{Z-1} H_{i,j}\, e_{j,c} = v_{\max}^{(i)} + \sum_{v\in K_{E_i}\setminus\{v_{\min}^{(i)}\}} Z \cdot (x_{i,v} - 1)$$

On note ici que par définition des ensembles K_{E_i}, nous avons $\sum_{v\in K_{E_i}\setminus\{v_{\min}^{(i)}\}} Z = v_{\max}^{(i)} - v_{\min}^{(i)}$. Ainsi, de manière analogue à la démonstration de la proposition 1, nous pouvons déduire la forme équivalente suivante :

$$\sum_{j\in V_{E_i}} H_{i,j} \cdot e_j = v_{\min}^{(i)} + \sum_{v\in K_{E_i}\setminus\{v_{\min}^{(i)}\}} Z \cdot x_{i,v}$$

Sous la contrainte (3b), la cohérence des variables $x_{i,v}$ pour la sélection des valeurs successives de v dans K_{E_i} par incréments de Z est assurée. Par conséquent, la contrainte (3a) est satisfaite si et seulement si $\sum_{j\in V_{E_i}} H_{i,j}\, e_j$ est égal à l'un des éléments de l'ensemble K_{E_i}, ce qui correspond exactement à la condition imposée par l'équation de syndrome (1a) associée à E_i. □

3.4 Encodage binaire

Dans cette section, notre objectif est de représenter en binaire le quotient de la division euclidienne de la somme $\sum_{j\in V_{E_i}} H_{i,j} e_j$, tout en imposant naturellement la contrainte de congruence fixée par s_i. Pour chaque équation E_i, $0 \leq i \leq m$, cette contrainte implique l'existence d'un unique entier $q \in \mathbb{N}$ tel que :

$$\sum_{j\in V_{E_i}} H_{i,j}\, e_j = s_i + Zq.$$

Nous représentons ensuite q en binaire à l'aide de variables auxiliaires $x_{i,\ell} \in \{0, 1\}$, définies par $x_{i,\ell} = 1$ si et seulement si le ℓ-ème bit de la représentation binaire de q est 1, pour $0 \leq \ell \leq L_i - 1$. Nous notons par $b_i(q) := (x_{i,L_i-1}, x_{i,L_i-2}, \ldots, x_{i,0})$ la représentation binaire de q, où L_i est le nombre minimal de bits nécessaires pour encoder la valeur maximale possible de q dans l'équation E_i. On note $b_{i,\ell}(q)$ le ℓ-ième bit de cette représentation, tel que $b_{i,\ell}(q) = x_{i,\ell}$.

Plus précisément, puisque nous avons défini $v_{\max}^{(i)}$ (resp. $v_{\min}^{(i)}$) comme la somme entière maximale (resp. minimale) possible pour l'équation E_i, nous pouvons introduire les notations suivantes :

$$q_{\max}^{(i)} := \frac{v_{\max}^{(i)} - s_i}{Z} \quad \text{et} \quad q_{\min}^{(i)} := \frac{v_{\min}^{(i)} - s_i}{Z},$$

où $q_{\max}^{(i)}$ (resp. $q_{\min}^{(i)}$) représente le quotient de la division euclidienne de $v_{\max}^{(i)}$ (resp. $v_{\min}^{(i)}$) par Z. Ainsi, nous pouvons représenter le quotient q comme suit, avec L_i représentant le nombre de bits requis pour q :

$$q = \sum_{\ell=0}^{L_i-1} 2^\ell\, x_{i,\ell}, \text{ où } L_i = \left\lceil \log_2\left(q_{\max}^{(i)} + 1\right)\right\rceil.$$

Le système $\mathsf{LWZSD}(n, k, t)$ peut donc s'écrire sous la forme suivante :

$$\sum_{j\in V_{E_i}} H_{i,j} e_j + Z \sum_{\ell=0}^{L_i-1} 2^\ell \cdot \bar{x}_{i,\ell} = s_i + Z \sum_{\ell=0}^{L_i-1} 2^\ell,$$

$$\forall\, 0 \leq i \leq m \quad \text{(4a)}$$

$$\bar{e}_{j,v} \vee e_{j,v-1}, \quad \forall 0 \leq j \leq n-1,\ 2 \leq v \leq Z-1 \quad \text{(4b)}$$

$$\sum_{j=0}^{n-1} e_{j,1} \geq t \quad \text{(4c)}$$

La contrainte (4a), écrite sous forme PB, exprime que la somme pondérée des variables d'erreur $\sum_{j\in V_{E_i}} H_{i,j} e_j$ satisfait la relation modulo Z, en utilisant les variables auxiliaires $x_{i,\ell}$ représentant le quotient q en binaire. De nouveau, les contraintes (4b) et (4c) encodent correctement les variables d'erreur et imposent que le vecteur d'erreur possède au moins t composantes non nulles.

Proposition 3 *La formule propositionnelle induite par les contraintes* (4) *est satisfaisable si et seulement si le système* $\mathsf{LWZSD}(n, k, t)$ *admet une solution.*

Preuve. Sous la contrainte (4b), la contrainte Pseudo-Booléenne (4a) est équivalente à l'équation de syndrome (1a) du système $\mathsf{LWZSD}(n, k, t)$. Pour une équation E_i, avec $i \in \{0, \ldots, m\}$, isoler la somme pondérée des variables d'erreur dans (4a) donne :

$$\sum_{j\in V_{E_i}} H_{i,j}\, e_j = s_i + Z \sum_{\ell=0}^{L_i-1} 2^\ell Z + \sum_{\ell=0}^{L_i-1} 2^\ell (x_{i,\ell} - 1)$$

Ainsi, en posant $q := \sum_{\ell=0}^{L_i-1} 2^\ell\, x_{i,\ell}$, nous obtenons la forme suivante :

$$\sum_{j\in V_{E_i}} H_{i,j}\, e_j = s_i + Z \sum_{\ell=0}^{L_i-1} 2^\ell\, x_{i,\ell} = s_i + Zq$$

ce qui correspond à l'équation de syndrome (1a) pour E_i. □

Il convient de noter que l'expression $s_i + Zq$ ne tient pas compte des bornes inférieure et supérieure imposées par le problème initial. En effet, pour chaque équation E_i, seules les valeurs de sommes appartenant à l'ensemble K_{E_i} sont admissibles. Cependant, nous autorisons actuellement également des valeurs qui ne devraient pas l'être, c'est-à-dire des quotients q tels que $s_i + Zq \notin K_{E_i}$. Il est donc nécessaire d'éliminer ces configurations invalides en ajoutant des contraintes complémentaires. Nous présentons ci-dessous deux méthodes pour imposer cette restriction en agissant directement sur le quotient et les variables auxiliaires qui le représentent.

Filtrage exhaustif. Puisque q est représenté à l'aide de L_i bits, la valeur maximale qu'il peut prendre est $2^{L_i}-1$. Selon la construction précédente, les seules valeurs admissibles pour le quotient q sont celles comprises entre $q^{(i)}_{\min}$ et $q^{(i)}_{\max}$. Ainsi, toutes les valeurs situées dans les intervalles suivants doivent être interdites :

$$I^{(i)} = [0, q^{(i)}_{\min}[\ \cup \ ]q^{(i)}_{\max}, 2^{L_i}-1], \quad \forall i \in \{0,\dots,m\}$$

Par conséquent, pour chaque valeur interdite, nous pouvons ajouter une clause qui exclut explicitement l'affectation correspondant à cette valeur. De telles clauses doivent imposer qu'au moins un bit du quotient soit différent et s'écrivent formellement comme suit :

$$\bigvee_{\substack{0\le \ell \le L_i-1 \\ b_{i,\ell}(v)=0}} x_{i,\ell} \ \vee \bigvee_{\substack{0\le \ell \le L_i-1 \\ b_{i,\ell}(v)=1}} \bar{x}_{i,\ell}, \quad \forall i \in \{0,\dots,m\},\ v \in I^{(i)} \tag{5}$$

Exemple 2 *Considérons $v = 12$, dont la représentation binaire sur 4 bits est $b_i(v) = (1,1,0,0)$. La clause résultante pour interdire cette valeur est : $(\bar{x}_{i,3} \vee \bar{x}_{i,2} \vee x_{i,1} \vee x_{i,0})$, ce qui empêche l'affectation $(x_{i,3}, x_{i,2}, x_{i,1}, x_{i,0}) = (1,1,0,0)$.*

Nous obtenons un premier modèle binaire qui combine l'ensemble des contraintes (4) et (5). Pour chaque équation E_i, cette méthode de filtrage exhaustif génère exactement une clause par valeur interdite de q. Afin de réduire le nombre de clauses et d'obtenir un encodage plus compact, nous proposons dans la section suivante une méthode plus judicieuse qui agit directement sur les variables binaires représentant le quotient.

Filtrage compact. Pour commencer, nous définissons l'ensemble $J^{(i)}_{\max}$ (resp. $J^{(i)}_{\min}$) comme l'ensemble des indices où le bit vaut 1 (resp. 0) dans $q^{(i)}_{\max}$ (resp. $q^{(i)}_{\min}$). Dans la suite, nous détaillons uniquement le raisonnement pour la borne supérieure $q^{(i)}_{\max}$. Le traitement de la borne inférieure $q^{(i)}_{\min}$ repose sur un raisonnement symétrique.

Soit $(b_{i,L_i-1}, \dots, b_{i,0})$ la décomposition binaire de $q^{(i)}_{\max}$. Nous identifions l'ensemble des indices de début de bloc, noté $\mathcal{B}^{(i)}_{\max} = \{l \in J^{(i)}_{\max} \mid b_{i,l-1} = 0\}$, qui correspond aux positions minimales de chaque séquence de 1 consécutifs dans la représentation binaire de $q^{(i)}_{\max}$.

Exemple 3 *On considère $q^{(i)}_{\max} = 111001101$. Alors $J^{(i)}_{\max} = \{8,7,6,3,2,0\}$, et l'ensemble $\mathcal{B}^{(i)}_{\max} = \{6,2\}$.*

Nous introduisons aussi des variables auxiliaires $p^{\max}_{i,j}$, qui représentent le préfixe binaire de $q^{(i)}_{\max}$, formé par les bits lus du bit de poids fort vers le bit de poids faible, jusqu'au bit d'indice j inclus. La variable $p^{\max}_{i,j}$ satisfait donc la relation suivante, où j' désigne le premier indice supérieur à j dans $\mathcal{B}^{(i)}_{\max}$:

$$p^{\max}_{i,j} \iff p^{\max}_{i,j'} \wedge \bigwedge_{\substack{j\le l\le j'-1 \\ b_{i,l}=1}} x_{i,l} \wedge \bigwedge_{\substack{j\le k\le j'-1 \\ b_{i,l}=0}} \bar{x}_{i,l}$$

Cette relation est encodée en CNF par les clauses suivantes :

$$\bigwedge_{j\le l\le L_i-1} (\bar{p}^{\max}_{i,j} \vee x_{i,l}),$$

$$\left(p^{\max}_{i,j} \vee \bigvee_{\substack{j\le l\le L_i-1 \\ b_{i,l}=1}} \bar{x}_{i,l}\right), \quad \forall i \in \{0,\dots,m\},$$

$$j = \max(\mathcal{B}^{(i)}_{\max}) \tag{6a}$$

$$(\bar{p}^{\max}_{i,j} \vee p^{\max}_{i,j'}) \wedge \bigwedge_{\substack{j\le l\le j'-1 \\ b_{i,l}=1}} (\bar{p}^{\max}_{i,j} \vee x_{i,l}) \wedge \bigwedge_{\substack{j\le l\le j'-1 \\ b_{i,l}=0}} (\bar{p}^{\max}_{i,j} \vee \bar{x}_{i,l}),$$

$$\left(p^{\max}_{i,j} \vee \bar{p}^{\max}_{i,j'} \vee \bigvee_{\substack{j\le l\le j'-1 \\ b_{i,l}=1}} \bar{x}_{i,l} \vee \bigvee_{\substack{j\le l\le j'-1 \\ b_{i,l}=0}} x_{i,l}\right),$$

$$\forall i \in \{0,\dots,m\},\ \forall j \in \mathcal{B}^{(i)}_{\max} \setminus \{\max(\mathcal{B}^{(i)}_{\max})\},$$

$$j' = \min\{l \in \mathcal{B}^{(i)}_{\max} \mid l > j\} \tag{6b}$$

Rappelons que L_i correspond au nombre minimal de bits requis pour représenter $q^{(i)}_{\max}$ tel que $s_i + Z \cdot q^{(i)}_{\max} = v^{(i)}_{\max}$ soit la plus grande somme entière avant réduction modulo Z appartenant à l'ensemble K_{E_i}. Il est clair que le bit de poids fort $b_{i,L_i-1} = 1$. Ainsi, pour obtenir une valeur strictement supérieure à $q^{(i)}_{\max}$, il est nécessaire de modifier au moins l'un des bits de poids inférieur (indices de 0 à L_i-2). Pour éliminer les valeurs supérieures à $q^{(i)}_{\max}$, nous interdisons d'abord les configurations dans lesquelles, entre deux bits fixés à 1, au moins un bit fixé à 0 est basculé, produisant ainsi une valeur strictement supérieure à $q^{(i)}_{\max}$. Pour empêcher cela, nous pouvons utiliser les clauses suivantes :

$$\bigwedge_{j'+1\le k\le j-1} \left(\bar{p}^{\max}_{i,j} \vee \bar{x}_{i,k}\right), \quad \forall i \in \{0,\dots,m\},$$

$$\forall j \in \mathcal{B}^{(i)}_{\max}, j' = \max\{l \in J^{(i)}_{\max} \mid l < j\} \tag{6c}$$

La contrainte ci-dessus permet au préfixe binaire jusqu'à l'indice j (inclus) d'être identique à celui de $q^{(i)}_{\max}$, tout en

interdisant à tout bit situé strictement entre $j-1$ et $j'+1$ de prendre la valeur 1. Cependant, lorsque nous atteignons le dernier élément de l'ensemble $J^{(i)}_{\max}$, les bits de poids inférieur peuvent encore être égaux à 0. Dans ce cas, il serait possible d'obtenir une valeur strictement supérieure à $q^{(i)}_{\max}$ en remplaçant l'un de ces zéros par un 1. Pour éviter cela, nous générons les clauses suivantes :

$$\bigwedge_{0\le k\le j-1} \left(\bar{p}^{\max}_{i,j} \vee \bar{x}_{i,k}\right), \quad \forall i \in \{0,\dots,m\}, \\ j = \min\left(J^{(i)}_{\max}\right) \quad \text{si } 0 \notin J^{(i)}_{\max} \quad \text{(6d)}$$

En combinant les contraintes (4) et les contraintes clausales du filtrage compact, nous obtenons un modèle qui élimine les valeurs strictement supérieures à $q^{(i)}_{\max}$ (et strictement inférieures à $q^{(i)}_{\min}$ par un raisonnement symétrique). Enfin, notons que pour une équation E_i donnée où L_i représente le nombre de bits nécessaires pour le quotient q, le filtrage exhaustif nécessite $O(2^{L_i})$ clauses sans variables additionnelles, alors que le filtrage compact ne nécessite que $O(L_i)$ clauses et variables.

Exemple 4 *Soit* $n = 20$, $Z = 3$, $t = 19$, *et considérons l'équation de parité* $E_0 = [1,0,0,0,0,0,0,0,2,1,2,0,2,2,2,1,1,0,2,1]$ *avec* $s_0 = 1$. *Les indices des variables apparaissant dans l'équation sont* $V_{E_0} = \{0,8,9,10,12,13,14,15,16,18,19\}$, *avec* $|V_{E_0}| = 11$. *Comme la contrainte de poids impose* $\mathrm{wt}(e) \ge t = 19$, *il peut y avoir au plus* $n - t = 1$ *composante nulle dans le vecteur d'erreur. Ainsi, nous posons* $m_0 = \max(0, |V_{E_0}| - (n-t)) = 10$, *ce qui représente le nombre minimal de composantes non nulles devant participer à cette équation.*
Nous pouvons maintenant définir l'ensemble K_{E_0}, *qui correspond aux valeurs entières que la somme des variables d'erreur peut prendre tout en satisfaisant la congruence imposée par* s_0. *Il y a 5 coefficients égaux à 1 et 6 coefficients égaux à 2. Nous définissons alors* $V^{\min}_{E_0} = \{0,9,15,16,19,8,10,12,13,14\}$ *comme l'ensemble des indices correspondant aux* m_0 *plus petits coefficients* $H_{0,j}$ *parmi* $j \in V_{E_0}$. *Cet ensemble nous permet de déterminer la borne inférieure de l'ensemble* K_{E_0}, *qui est* $\sum_{j\in V^{\min}_{E_0}} H_{0,j} = 15$.
Nous savons également que la somme pondérée des variables d'erreur atteint sa valeur maximale lorsque toutes les variables e_j *sont égales à* $Z-1$; *ainsi, la somme entière réalisée par cette équation doit satisfaire* $15 \le \sum_{j\in V_{E_0}} H_{0,j} e_j \le 34$. *Puisque la contrainte modulaire impose* $\sum_{j\in V_{E_0}} H_{0,j} e_j \equiv s_0 = 1 \pmod 3$, *les valeurs admissibles sont exactement* $K_{E_0} = \{v \in [15,34] \mid v \equiv 1 \pmod 3\} = \{16,19,22,25,28,31,34\}$. *Ainsi,* $v^{(0)}_{\min} = 16$ *et* $v^{(0)}_{\max} = 34$.
Rappelons que la somme pondérée des variables d'erreur pour une équation E_i *s'écrit*

$$\sum_{j\in V_{E_i}} H_{i,j}\, e_j = s_i + Z \sum_{\ell=0}^{L_i-1} 2^\ell x_{i,\ell}\,,$$

q	$x_{0,3}$	$x_{0,2}$	$x_{0,1}$	$x_{0,0}$	$q\cdot Z + s_0$	État
0	0	0	0	0	1	✗ (6c)
1	0	0	0	1	4	✗ (6c)
2	0	0	1	0	7	✗ (6c)
3	0	0	1	1	10	✗ (6c)
4	0	1	0	0	13	✗ (6d)
5	0	1	0	1	$v^{(0)}_{\min} = 16$	✓
6	0	1	1	0	19	✓
7	0	1	1	1	22	✓
8	1	0	0	0	25	✓
9	1	0	0	1	28	✓
10	1	0	1	0	31	✓
11	1	0	1	1	$v^{(0)}_{\max} = 34$	✓
12	1	1	0	0	37	✗ (6c)
13	1	1	0	1	40	✗ (6c)
14	1	1	1	0	43	✗ (6c)
15	1	1	1	1	46	✗ (6c)

TABLE 1 – Représentation binaire des quotients et état des sommes $q\cdot Z + s_0$ (✓ : autorisé, ✗ : interdit).

où $q = \sum_{\ell=0}^{L_i-1} 2^\ell x_{i,\ell}$ *représente le quotient, et* L_i *est choisi pour encoder toutes les valeurs admissibles de* q. *Nous calculons*

$$L_0 = \left\lceil \log_2\left(\left\lfloor \frac{v^{(0)}_{\max} - s_0}{Z} \right\rfloor + 1 \right) \right\rceil = 4$$

et, par conséquent, seuls quatre bits sont nécessaires pour représenter toutes les valeurs possibles du quotient. Le Tableau 1 illustre le processus de filtrage appliqué aux configurations du quotient q *encodé sur* $L_0 = 4$ *bits.*
Pour éliminer les valeurs strictement supérieures à $q^{(0)}_{\max} = 1011_2$, *nous définissons* $J^{(0)}_{\max} = \{3,1,0\}$ *et* $\mathcal{B}^{(0)}_{\max} = \{3\}$. *Nous introduisons une variable auxiliaire* $p^{\max}_{0,3}$ *en appliquant la contrainte (6a), représentant le préfixe de* $q^{(0)}_{\max}$ *jusqu'à l'indice 3, ce qui donne les clauses suivantes :* $(\bar{p}^{\max}_{0,3} \vee x_{0,3})$ *ainsi que* $(p^{\max}_{0,3} \vee \bar{x}_{0,3})$. *Ensuite, en appliquant la contrainte (6c), nous imposons que la variable* $x_{0,2}$ *ne puisse pas prendre la valeur* 0 *lorsque* $p^{\max}_{0,3}$ *est vraie, conduisant à la clause* $(\bar{p}^{\max}_{0,3} \vee x_{0,2})$. *Nous procédons de manière analogue pour éliminer les valeurs strictement inférieures à* $q^{(0)}_{\min} = 0101_2$.

4 Évaluation expérimentale

4.1 Protocole expérimental

Pour évaluer les modèles introduits dans la section précédente, nous avons testé des instances issues du Large Weight Ternary Syndrome Decoding Challenge[1] (LW3SD), avec des valeurs de n allant de 10 à 100. Nous désignons nos modèles (2), (3), (4;5), et (4;6) respectivement par CNF1, CNF2, CNF3 et CNF4. De plus, nous avons implémenté ces modèles à l'aide du solveur CP-SAT

1. https://decodingchallenge.org/large-weight

Instance		CaDiCaL				CryptoMiniSat				CP-SAT			
n	t	CNF1	CNF2	CNF3	CNF4	CNF1	CNF2	CNF3	CNF4	CP1	CP2	CP3	CP4
10	9	89.26	14.05	78.33	26.47	86.21	15.06	80.4	26.3	**0.44**	0.57	0.55	**0.44**
20	19	49.52	192.04	1.02	1.13	38.01	200.92	10.88	3.02	0.59	0.7	0.55	**0.5**
30	29	17.85	147.73	63.55	**2.09**	6.98	31.64	152.1	66.58	3.20	9.24	2.31	1.89
40	39	52.15	103.14	115.35	20.14	23.91	126.5	26.26	67.19	16.86	33.23	14.71	**11.81**
50	49	90.41	71.71	**69.14**	87.7	173.34	250.39	79.73	145.62	161.79	328.22	82.74	85.5
60	59	807.09	**567.04**	577.33	852.62	1166.52	980.41	782.54	1107.86	1614.22	3806.92	807.07	1300.56
70	69	11772.57	14143.43	**6583.66**	17734.06	11165.61	10752.38	9054.9	13356.76	20704.41 (11)	10295.93 (6)	25763.99 (14)	23997.54 (15)
80	79	7888.75 (4)	10527.77 (7)	6855.34 (3)	8636.05 (7)	14877.03 (11)	**17253.96 (12)**	19644.53 (10)	9592.45 (5)	851.66 (1)	- (0)	- (0)	6984.33 (5)
90	89	**4580.82 (2)**	- (0)	- (0)	- (0)	464.74 (1)	- (0)	1692.33 (1)	2709.72 (1)	- (0)	- (0)	- (0)	2552.23 (1)
100	99	**533.04 (1)**	- (0)	- (0)	- (0)	802.53 (1)	- (0)	- (0)	- (0)	- (0)	- (0)	2309.31 (1)	- (0)
TOTAL		25881.48 (147)	25766.92 (147)	14343.72 (143)	27360.25 (147)	**28804.88 (153)**	29611.25 (152)	31523.67 (151)	27075.5 (146)	23353.18 (132)	14474.82 (126)	28981.23 (135)	34934.8 (141)

TABLE 2 – Résultats en termes de temps de résolution pour chaque taille n et poids t pour les différents modèles du problème LW3SD. Si les problèmes n'ont pas été résolus dans le temps imparti, le nombre d'instances résolues est indiqué entre parenthèses "()". Les meilleurs résultats sont mis en évidence en gras, d'abord sur la base du nombre d'instances résolues, puis, en cas d'égalité, sur la base du temps de résolution.

d'OR-Tools [2], notés CP1, CP2, CP3 et CP4, correspondant à leurs formulations CNF respectives. Les expérimentations ont été menées en utilisant les solveurs SAT CryptoMiniSat [30] et CaDiCaL [7], ainsi que le solveur CP-SAT. Tous les tests ont été effectués sur la plateforme de calcul MatriCS [3], sur la partition "bigmem" équipée de 12 serveurs biprocesseurs Intel Xeon E5-2680 v4 (2,40 GHz) disposant chacun de 512 Go de mémoire, avec une limite de temps fixée à 3600 secondes (1 heure) par instance.

Nous avons utilisé la bibliothèque PySAT [4] pour encoder nos modèles ainsi que les contraintes Pseudo-Booléennes. En particulier, nous utilisons l'encodage CardNetwork [1] pour les contraintes de cardinalité présentes dans tous les modèles CNF. Pour une contrainte de cardinalité de la forme $\sum_{i=1}^{h} l_i \leq b$, la complexité en termes de nombre de clauses (et de variables) générées est de $O(h \log^2 b)$ pour CardNetwork. Nous avons choisi l'encodage BinMerge [18] pour encoder les contraintes Pseudo-Booléennes, qui présente une complexité de $O(h \log^2 h \log W)$, où h désigne le nombre de littéraux dans la somme et W le poids maximal. Enfin, pour chaque valeur de n et chaque modèle, nous avons généré 20 instances en utilisant des graines aléatoires distinctes. Notre code source pour la génération des instances est disponible en ligne [5].

4.2 Résultats

Les résultats concernant le nombre d'instances résolues et les temps de résolution sont présentés dans le Tableau 2. Cette analyse met en évidence une distinction nette entre efficacité immédiate et passage à l'échelle. Sur les instances de petite taille, le solveur CP-SAT, et en particulier le modèle CP4, domine en termes de rapidité. Cependant, cette efficacité initiale ne se maintient pas lorsque la taille des instances augmente : les modèles CP voient leurs performances se dégrader fortement à partir de $n \geq 70$.

À l'inverse, les solveurs CaDiCaL et CryptoMiniSat se montrent plus robustes sur les instances de grande taille. CryptoMiniSat, associé au modèle CNF1, obtient la meilleure performance globale avec un total de 153 instances résolues. Bien que moins rapides sur les cas les plus simples, ces solveurs restent capables de traiter des instances de taille $n \geq 70$, là où les approches CP échouent quasi systématiquement à produire une solution dans le temps imparti.

L'analyse du compromis entre temps de résolution et taux de succès met en avant le modèles CNF1 associé à CryptoMiniSAT. Contrairement à CP-SAT, dont les performances se dégradent fortement sur les grandes instances, jusqu'à ne résoudre presque aucune instance pour $n \geq 80$, les approches SAT maintiennent une progression bien plus prévisible et stable. Le modèle CNF1 s'impose ici comme le plus robuste, étant le seul modèle capable de résoudre des instances jusqu'à $n = 100$.

La Figure 1 présente les temps de résolution triés par ordre croissant pour chaque configuration. Les courbes permettent de visualiser clairement le passage à l'échelle de chaque approche. On observe que les configurations CP-SAT décollent très tôt : leurs courbes commencent à diverger fortement à partir des instances $n = 120$, avec des temps qui atteignent rapidement le plafond de 3600 secondes. À l'inverse, les courbes SAT restent nettement plus basses sur l'ensemble de la distribution, ne dépassant le seuil de 1000 secondes que pour les instances au-delà de $n = 135$. Ce comportement illustre bien l'écart de passage à l'échelle : là où CP-SAT sature sur les instances difficiles, les solveurs SAT continuent à résoudre des instances que CP-SAT ne parvient pas à traiter.

Compte tenu de ces résultats, bien que CP-SAT s'avère efficace sur les instances de petite taille, il ne parvient pas à maintenir des performances compétitives à mesure que la difficulté du problème augmente, résolvant nettement moins d'instances que les configurations SAT. En raison de cet écart marqué en termes de passage à l'échelle et de taux de réussite global, la suite de cette étude se concentrera exclusivement sur l'analyse des solveurs SAT, car ils démontrent une capacité supérieure à traiter les instances les plus complexes du problème LW3SD.

2. https://developers.google.com/optimization/cp/cp_solver
3. https://www.matrics.u-picardie.fr/
4. https://pysathq.github.io/
5. https://github.com/carlberton/sat-LWZSD.git

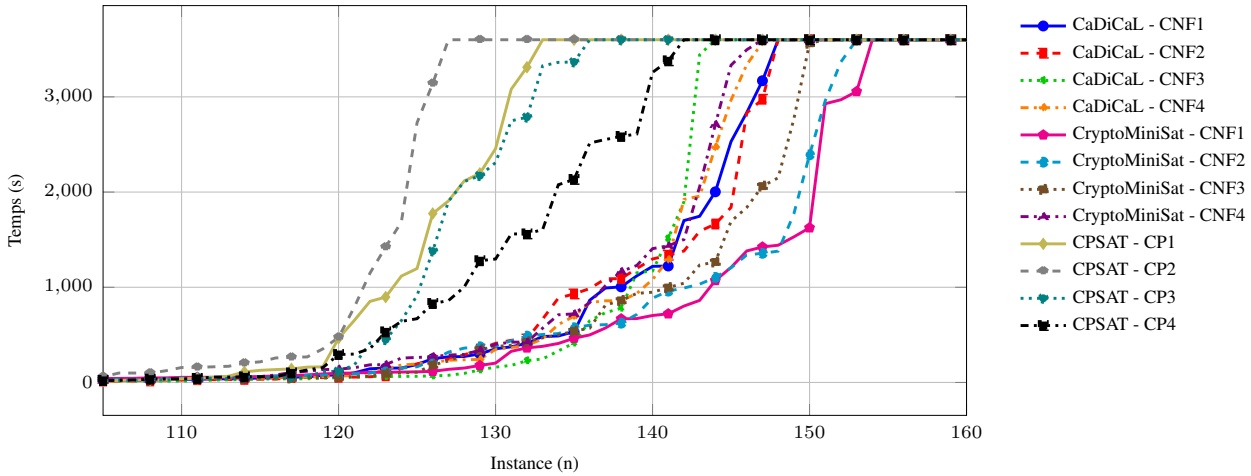

FIGURE 1 – Évolution des temps de résolution triés par ordre croissant pour les différents solveurs et modèles du problème LW3SD.

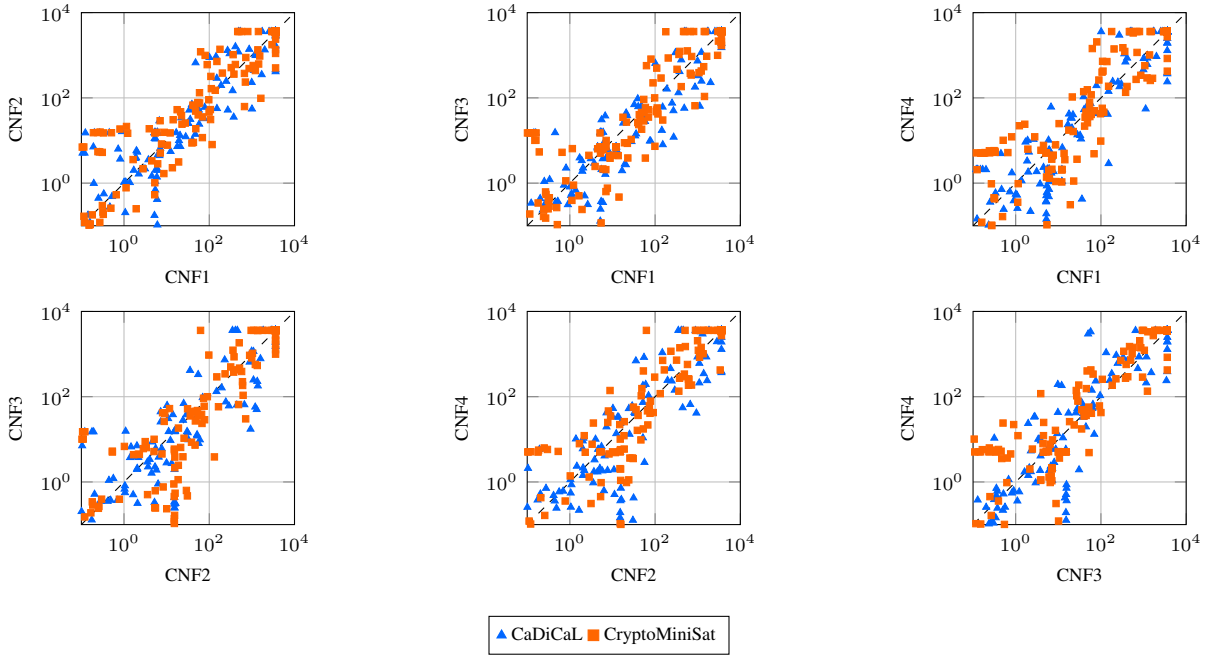

FIGURE 2 – Comparaison des temps de résolution en secondes par instance du problème LW3SD.

La Figure 2 présente une comparaison par instance sous forme de nuage de points comparatif de paires de modèles. L'encodage CNF1 s'impose comme la solution la plus robuste : il surpasse CNF4 dans 48% des cas avec CaDiCaL contre seulement 29% pour CNF4. Cette domination se confirme face à CNF2, où CNF1 l'emporte dans 41.5% des instances avec CryptoMiniSat. À l'inverse, l'encodage CNF3 s'avère être le moins efficace : il est systématiquement surpassé par CNF4, qui s'impose dans 43% des instances contre seulement 34% pour CNF3 avec CryptoMiniSat, validant ainsi l'intérêt de l'approche par filtrage compact par rapport au filtrage exhaustif. Enfin, la comparaison entre CNF2 et CNF4 montre des performances équilibrées, bien qu'un léger avantage se dessine pour CNF2 avec CryptoMiniSat (38.5% de victoires contre 34%). Enfin, il est important de noter qu'environ 20% à 26% des instances se distinguent par une performance similaires, principalement due aux limites de temps atteintes par les deux configurations ou à des résolutions instantanées.

5 Conclusion

Cet article présente une étude de différentes modélisations CNF pour le problème du décodage de syndrome à poids élevé (LWZSD) sur le corps fini $\mathbb{F}_Z$ (avec Z premier). Nous avons mené une évaluation expérimentale sur des instances ternaires ($Z = 3$) en comparant les performances des modèles introduits à l'aide de solveurs SAT de l'état de l'art, plus précisément CryptoMiniSat et CaDiCaL, ainsi que du solveur CP-SAT. Nos expériences démontrent que les sol-

veurs SAT sont beaucoup plus robustes et parviennent à résoudre les instances les plus complexes.
Une perspective intéressante de nos travaux consiste à adapter les mécanismes internes des solveurs SAT afin de mieux exploiter la structure intrinsèque du problème LWZSD, notamment par la conception d'heuristiques de branchement dédiées. Par exemple, on pourrait favoriser les variables d'erreur en fonction de la distribution des coefficients dans les équations de parité ou du poids d'erreur cible. Par ailleurs, il serait intéressant de concevoir des modèles plus compacts capturant mieux les relations complexes entre les différentes équations du syndrome. De tels modèles pourraient exploiter les similitudes structurelles apparaissant entre les équations et éviter ainsi d'encoder des motifs redondants. Cela permettrait potentiellement de réduire la taille de l'encodage et d'améliorer l'efficacité des solveurs. Enfin, pour traiter des instances encore plus complexes, il serait pertinent d'explorer des méthodes incomplètes dédiées ou de développer des approches hybrides combinant techniques complètes et incomplètes, afin de concilier le passage à l'échelle avec les garanties de solution.

Remerciements

Ce travail est soutenu par les projets ANR-24-CE23-6126 (BforSAT) et 22-PECY-0010 (CRYPTANALYSE), financés par l'Agence Nationale de la Recherche (ANR). Ce travail a également bénéficié d'un accès aux ressources de calcul de la "Plateforme MatriCS" de l'Université de Picardie Jules Verne, cofinancée par l'Union européenne via le Fonds Européen de Développement Régional (FEDER) et le Conseil Régional des Hauts-de-France.

Références

[1] Roberto Asín, Robert Nieuwenhuis, Albert Oliveras, and Enric Rodríguez-Carbonell. Cardinality networks and their applications. In Oliver Kullmann, editor, *SAT 2009*, volume 5584 of *LNCS*, pages 167–180. Springer, 2009.

[2] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in 2 n/20 : How 1 + 1 = 0 improves information set decoding. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*, pages 520–536. Springer, 2012.

[3] Elwyn R. Berlekamp. *Algebraic coding theory*. McGraw-Hill series in systems science. McGraw-Hill, 1968.

[4] Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. A. van Tilborg. On the inherent intractability of certain coding problems (corresp.). *IEEE Trans. Inf. Theory*, 24(3) :384–386, 1978.

[5] Carl Berton, Sami Cherif, and Claire Delaplace. Satisfiabilité pour le décodage par syndrome. In *Journées Francophones de Programmation par Contraintes (JFPC 2025)*, Dijon, France, June 2025.

[6] Carl Berton, Sami Cherif, and Claire Delaplace. Sat-based syndrome decoding and low-weight codewords. In *27th International Symposium on Formal Methods (FM 2026)*, Tokyo, Japan, 2026. To appear.

[7] Armin Biere, Tobias Faller, Katalin Fazekas, Mathias Fleury, Nils Froleyks, and Florian Pollitt. Cadical 2.0. In Arie Gurfinkel and Vijay Ganesh, editors, *Computer Aided Verification - 36th International Conference, CAV 2024, Montreal, QC, Canada, July 24-27, 2024, Proceedings, Part I*, volume 14681 of *Lecture Notes in Computer Science*, pages 133–152. Springer, 2024.

[8] Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability - Second Edition*, volume 336 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2021.

[9] Rémi Bricout, André Chailloux, Thomas Debris-Alazard, and Matthieu Lequesne. Ternary syndrome decoding with large weight. In Kenneth G. Paterson and Douglas Stebila, editors, *Selected Areas in Cryptography - SAC 2019 - 26th International Conference, Waterloo, ON, Canada, August 12-16, 2019, Revised Selected Papers*, volume 11959 of *Lecture Notes in Computer Science*, pages 437–466. Springer, 2019.

[10] Lily Chen, Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray A Perlner, and Daniel Smith-Tone. *Report on post-quantum cryptography*, volume 12. US Department of Commerce, National Institute of Standards and Technology . . ., 2016.

[11] Stephen A. Cook. The complexity of theorem-proving procedures. In Michael A. Harrison, Ranan B. Banerji, and Jeffrey D. Ullman, editors, *STOC 1971*, pages 151–158. ACM, 1971.

[12] Nicolas T. Courtois and Gregory V. Bard. Algebraic cryptanalysis of the data encryption standard. In Steven D. Galbraith, editor, *Cryptography and Coding, 11th IMA International Conference, Cirencester, UK, December 18-20, 2007, Proceedings*, volume 4887 of *Lecture Notes in Computer Science*, pages 152–169. Springer, 2007.

[13] Joan Daemen and Vincent Rijmen. *The Design of Rijndael : AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.

[14] Thomas Debris-Alazard, Nicolas Sendrier, and Jean-Pierre Tillich. Wave : A new code-based signature scheme. *IACR Cryptol. ePrint Arch.*, page 996, 2018.

[15] Andre Esser and Emanuele Bellini. Syndrome decoding estimator. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *IACR 2022*, volume 13177 of *LNCS*, pages 112–141. Springer, 2022.

[16] Aarti Gupta, Malay K. Ganai, and Chao Wang. Sat-based verification methods and applications in hardware verification. In Marco Bernardo and Alessandro Cimatti, editors, *SFM 2006, Advanced Lectures*, volume 3965 of *LNCS*, pages 108–143. Springer, 2006.

[17] Frédéric Lafitte, Jorge Nakahara Jr., and Dirk Van Heule. Applications of SAT solvers in cryptanalysis : Finding weak keys and preimages. *J. Satisf. Boolean Model. Comput.*, 9(1) :1–25, 2014.

[18] Norbert Manthey, Tobias Philipp, and Peter Steinke. A more compact translation of pseudo-boolean constraints into CNF such that generalized arc consistency is maintained. In Carsten Lutz and Michael Thielscher, editors, *KI 2014*, volume 8736 of *LNCS*, pages 123–134. Springer, 2014.

[19] R. J. McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory. *Deep Space Network Progress Report*, 44 :114–116, January 1978.

[20] Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loıc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Edoardo Persichetti, Gilles Zémor, and IC Bourges. Hamming quasi-cyclic (hqc). *NIST PQC Round*, 2(4) :13, 2018.

[21] Ilya Mironov and Lintao Zhang. Applications of SAT solvers to cryptanalysis of hash functions. In Armin Biere and Carla P. Gomes, editors, *Theory and Applications of Satisfiability Testing - SAT 2006, 9th International Conference, Seattle, WA, USA, August 12-15, 2006, Proceedings*, volume 4121 of *Lecture Notes in Computer Science*, pages 102–115. Springer, 2006.

[22] Shintaro Narisada, Shusaku Uemura, Hiroki Okada, Hiroki Furue, Yusuke Aikawa, and Kazuhide Fukushima. Solving mceliece-1409 in one day - cryptanalysis with the improved BJMM algorithm. In Nicky Mouha and Nick Nikiforakis, editors, *ISC 2024*, volume 15258 of *LNCS*, pages 3–23. Springer, 2024.

[23] Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Trans. Inf. Theory*, 8(5) :5–9, 1962.

[24] Jussi Rintanen. Planning and SAT. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability - Second Edition*, volume 336 of *Frontiers in Artificial Intelligence and Applications*, pages 765–789. IOS Press, 2021.

[25] Olivier Roussel and Vasco Manquinho. Pseudo-boolean and cardinality constraints. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability - Second Edition*, Frontiers in Artificial Intelligence and Applications, pages 1087–1129. IOS Press, 2021.

[26] Claude E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3) :379–423, 1948.

[27] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.*, 41(2) :303–332, 1999.

[28] João P. Marques Silva and Karem A. Sakallah. GRASP - a new search algorithm for satisfiability. In Rob A. Rutenbar and Ralph H. J. M. Otten, editors, *ICCAD 1996*, pages 220–227. IEEE Computer Society / ACM, 1996.

[29] Mate Soos, Karsten Nohl, and Claude Castelluccia. Extending SAT solvers to cryptographic problems. In Oliver Kullmann, editor, *SAT 2009*, volume 5584 of *LNCS*, pages 244–257. Springer, 2009.

[30] Mate Soos, Karsten Nohl, and Claude Castelluccia. Extending SAT solvers to cryptographic problems. In Oliver Kullmann, editor, *SAT 2009*, volume 5584 of *LNCS*, pages 244–257. Springer, 2009.

[31] Jacques Stern. A method for finding codewords of small weight. In Gérard D. Cohen and Jacques Wolfmann, editors, *Coding Theory and Applications, 3rd International Colloquium, Toulon, France, November 2-4, 1988, Proceedings*, volume 388 of *Lecture Notes in Computer Science*, pages 106–113. Springer, 1988.

[32] Monika Trimoska, Gilles Dequen, and Sorina Ionica. Logical cryptanalysis with wdsat. In Chu-Min Li and Felip Manyà, editors, *Theory and Applications of Satisfiability Testing - SAT 2021 - 24th International Conference, Barcelona, Spain, July 5-9, 2021, Proceedings*, volume 12831 of *Lecture Notes in Computer Science*, pages 545–561. Springer, 2021.