

Satisfiability for Large Weight Syndrome Decoding

Conference on Principles and Practice of Constraint Programming (CP 2026) - Lisbon

Carl Berton, Sami Cherif and Claire Delaplace

MIS Laboratory - University of Picardie Jules Verne - France, Amiens



Outline

- 1 Syndrome decoding
- 2 Satisfiability
- 3 Encodings
- 4 Experimental Evaluation
- 5 Conclusion and Future Work

Error-correcting codes



Error-correcting codes



001101011100100011



Error-correcting codes



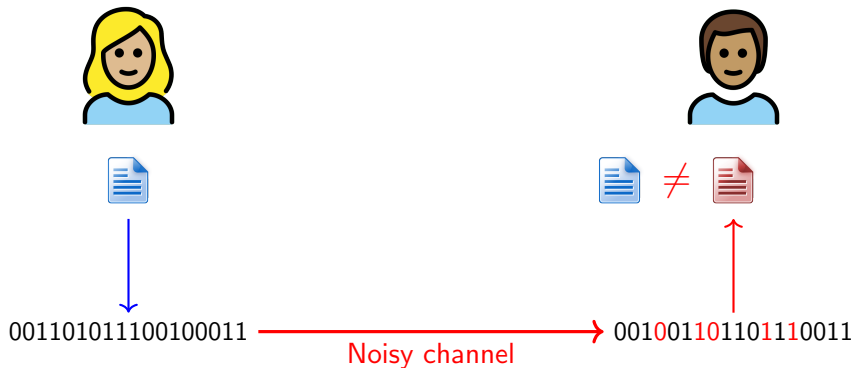
001101011100100011

—————→ 001001101101110011

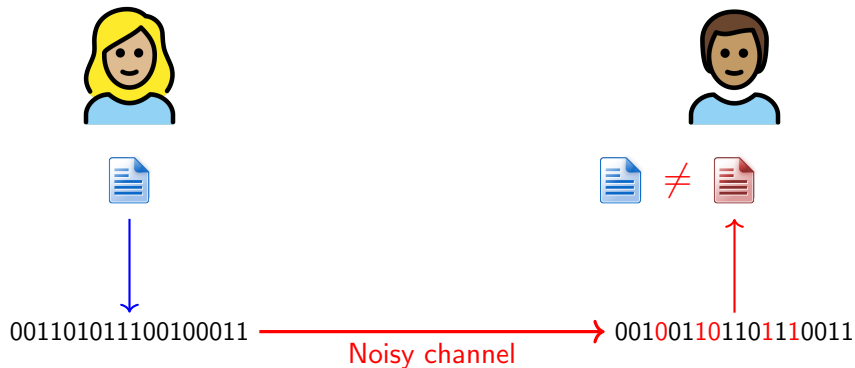
Noisy channel



Error-correcting codes

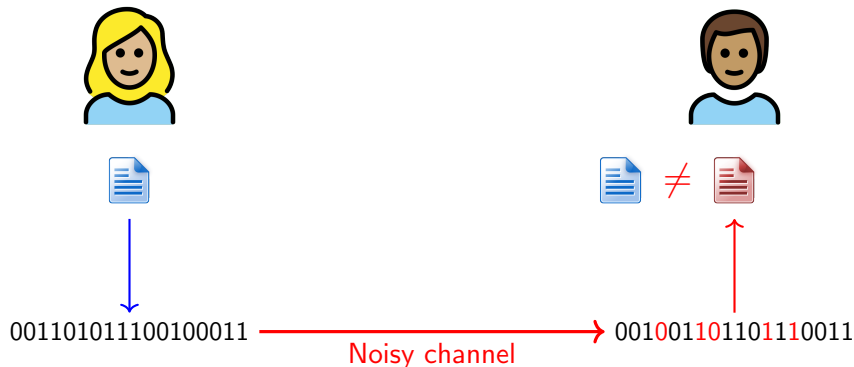


Error-correcting codes



- **Solution:** Add redundancy to detect and correct errors

Error-correcting codes



- **Solution:** Add redundancy to detect and correct errors
- This is the role of **error-correcting codes**

A **binary linear code** C is a subspace of $\mathbb{F}_2^n$ defined by the linear map

$$\begin{aligned} g : \mathbb{F}_2^k &\mapsto \mathbb{F}_2^n \\ x &\mapsto x \cdot G \end{aligned}$$

where $G \in \mathbb{F}_2^{k \times n}$ is the **generator matrix** of C , with $n > k$. Thus:

$$C = \{x \cdot G \mid x \in \mathbb{F}_2^k\}$$

A **binary linear code** C is a subspace of $\mathbb{F}_2^n$ defined by the linear map

$$\begin{aligned} g : \mathbb{F}_2^k &\mapsto \mathbb{F}_2^n \\ x &\mapsto x \cdot G \end{aligned}$$

where $G \in \mathbb{F}_2^{k \times n}$ is the **generator matrix** of C , with $n > k$. Thus:

$$C = \{x \cdot G \mid x \in \mathbb{F}_2^k\}$$

Let $H \in \mathbb{F}_2^{(n-k) \times n}$ be the **parity-check matrix** of the code C , such that:

$$C = \{c \in \mathbb{F}_2^n \mid H \cdot c^T = 0\}.$$

Encoding over a noisy channel



$$c = x \cdot G$$



$$y = x \cdot G \oplus e \in \mathbb{F}_2^n$$

Encoding over a noisy channel



$$c \oplus e$$



$$c = x \cdot G$$

$$y = x \cdot G \oplus e \in \mathbb{F}_2^n$$

Decoding problem: Given a received word y , how can we recover x ?

Encoding over a noisy channel



$$c \oplus e$$



$$c = x \cdot G$$

$$y = x \cdot G \oplus e \in \mathbb{F}_2^n$$

Decoding problem: Given a received word y , how can we recover x ?

Key idea: Recover $e \Rightarrow$ recover $c \Rightarrow$ recover x

Syndrome decoding

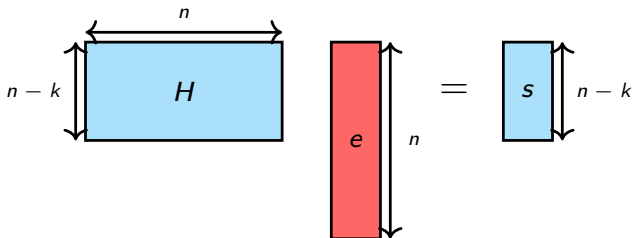
- t : maximum number of errors, i.e., $\text{wt}(e) \leq t$ (assumed to be small)

Syndrome decoding

- t : maximum number of errors, i.e., $\text{wt}(e) \leq t$ (assumed to be small)

Syndrome :

$$s = H \cdot y^T = H \cdot (\textcolor{red}{x} \cdot G \oplus \textcolor{red}{e})^T = H \cdot (\textcolor{red}{x} \cdot G)^T \oplus H \cdot \textcolor{red}{e}^T = H \cdot \textcolor{red}{e}^T$$

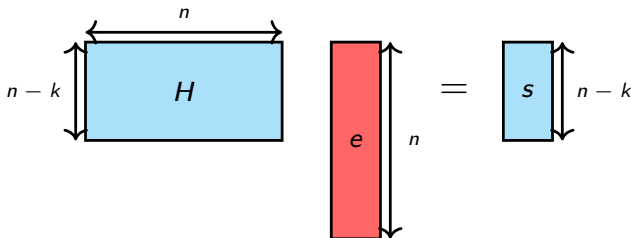


Syndrome decoding

- t : maximum number of errors, i.e., $\text{wt}(e) \leq t$ (assumed to be small)

Syndrome :

$$s = H \cdot y^T = H \cdot (\mathbf{x} \cdot G \oplus \mathbf{e})^T = H \cdot (\mathbf{x} \cdot G)^T \oplus H \cdot \mathbf{e}^T = H \cdot \mathbf{e}^T$$



Goal: Find an error vector $e \in \mathbb{F}_2^n$ such that $He^T = s$ and $\text{wt}(e) \leq t$

Syndrome decoding: a simple example

Let $H \in \mathbb{F}_2^{2 \times 4}$, $s \in \mathbb{F}_2^2$ and $t = 1$:

$$H = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \quad s = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

The vector $e = (0, 0, 1, 0)$ is a valid solution since $\text{wt}(e) = 1 \leq t$ and $He^T = s$.

Syndrome decoding: a simple example

Let $H \in \mathbb{F}_2^{2 \times 4}$, $s \in \mathbb{F}_2^2$ and $t = 1$:

$$H = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \quad s = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

The vector $e = (0, 0, 1, 0)$ is a valid solution since $\text{wt}(e) = 1 \leq t$ and $He^T = s$.

- Syndrome decoding is an NP-complete problem
- It is at the core of several code-based cryptographic schemes, such as McEliece
- It is of cryptographic interest for post-quantum cryptography
- SAT encodings of this problem have already been proposed

Large Weight Syndrome Decoding

- Variant of the classical syndrome decoding problem

Input: $H \in \mathbb{F}_Z^{(n-k) \times n}$, $s \in \mathbb{F}_Z^{n-k}$, where Z is prime, and $t \in \mathbb{N}$.

Differences from the classical problem:

- Finite field $\mathbb{F}_Z$ with $Z > 2$
- Search for a large-weight vector

Goal: Find $e \in \mathbb{F}_Z^n$ such that $He^T = s$ and $\text{wt}(e) \geq t$

Outline

- 1 Syndrome decoding
- 2 Satisfiability
- 3 Encodings
- 4 Experimental Evaluation
- 5 Conclusion and Future Work

The Boolean Satisfiability Problem (SAT)

Conjunctive Normal Form (CNF):

- **Literal:** A Boolean variable $l = x$ or its negation $l = \neg x$
- **Clause:** A disjunction of literals, $c = l_1 \vee \dots \vee l_n$, satisfied when at least one of its literals is satisfied
- **CNF formula:** A conjunction of clauses, $\varphi = c_1 \wedge \dots \wedge c_m$

Example:

$$\varphi = (x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_1 \vee x_4) \wedge (x_3 \vee \neg x_4)$$

The **Boolean satisfiability problem (SAT)** consists in determining whether there exists an assignment of the variables that satisfies a given formula.

Pseudo-Boolean Constraints (PB)

General form:

$$\left(\sum_{i=1}^h a_i \times l_i \right) \circ k,$$

where $a_i, k \in \mathbb{N}$, l_i are Boolean literals, and $\circ \in \{\leq, =, \geq\}$

Example:

$$x_1 + x_2 + x_3 + x_4 \geq 2$$

This constraint requires at least two of the variables x_1, x_2, x_3, x_4 to be true.

Outline

- 1 Syndrome decoding
- 2 Satisfiability
- 3 Encodings**
- 4 Experimental Evaluation
- 5 Conclusion and Future Work

Variables and notation

- $e_j \in \{0, \dots, Z - 1\}$: component of the error vector
- $e_{j,k}$: Boolean variable encoding the value of e_j ,

$$e_{j,k} = 1 \iff e_j \geq k, \quad 1 \leq k \leq Z - 1.$$

Variables and notation

- $e_j \in \{0, \dots, Z - 1\}$: component of the error vector
- $e_{j,k}$: Boolean variable encoding the value of e_j ,

$$e_{j,k} = 1 \iff e_j \geq k, \quad 1 \leq k \leq Z - 1.$$

- s_i : i -th component of the syndrome
- E_i : i -th equation defined by the i -th row of H ,

$$E_i : \sum_{j=0}^{n-1} H_{i,j} e_j \equiv s_i \pmod{Z}.$$

Variables and notation

- $e_j \in \{0, \dots, Z - 1\}$: component of the error vector
- $e_{j,k}$: Boolean variable encoding the value of e_j ,

$$e_{j,k} = 1 \iff e_j \geq k, \quad 1 \leq k \leq Z - 1.$$

- s_i : i -th component of the syndrome
- E_i : i -th equation defined by the i -th row of H ,

$$E_i : \sum_{j=0}^{n-1} H_{i,j} e_j \equiv s_i \pmod{Z}.$$

- V_{E_i} : indices of the variables appearing in E_i ,

$$V_{E_i} = \{j \mid H_{i,j} \neq 0\}.$$

Variables and notation

- $e_j \in \{0, \dots, Z - 1\}$: component of the error vector
- $e_{j,k}$: Boolean variable encoding the value of e_j ,

$$e_{j,k} = 1 \iff e_j \geq k, \quad 1 \leq k \leq Z - 1.$$

- s_i : i -th component of the syndrome
- E_i : i -th equation defined by the i -th row of H ,

$$E_i : \sum_{j=0}^{n-1} H_{i,j} e_j \equiv s_i \pmod{Z}.$$

- V_{E_i} : indices of the variables appearing in E_i ,

$$V_{E_i} = \{j \mid H_{i,j} \neq 0\}.$$

- K_{E_i} : possible values of the weighted sum compatible with s_i ,

$$K_{E_i} = \left\{ v \in \{v_{\min}^{(i)}, \dots, v_{\max}^{(i)}\} \mid v \equiv s_i \pmod{Z} \right\}.$$

One-Hot Encoding

- $x_{i,v} = 1 \iff \sum_{j \in V_{E_i}} H_{i,j} e_j = v, \quad v \in K_{E_i}$

One-Hot Encoding

- $x_{i,v} = 1 \iff \sum_{j \in V_{E_i}} H_{i,j} e_j = v, \quad v \in K_{E_i}$

Force the sum of equation E_i to take a valid value:

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

Select exactly one possible value:

$$\sum_{v \in K_{E_i}} x_{i,v} = 1, \quad \forall i \quad (1b)$$

Ensure consistency of the encoding of e_j :

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

Large-weight constraint:

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

Unary Encoding

- $x_{i,v} = 1 \iff \sum_{j \in V_{E_i}} H_{i,j} e_j \geq v, \quad v \in K_{E_i}$

Unary Encoding

- $x_{i,v} = 1 \iff \sum_{j \in V_{E_i}} H_{i,j} e_j \geq v, \quad v \in K_{E_i}$

Force the sum of equation E_i to take a valid value:

$$\sum_{j \in V_{E_i}} \sum_{c=1}^{Z-1} H_{i,j} e_{j,c} + \sum_{v \in K_{E_i} \setminus \{v_{\min}^{(i)}\}} Z \bar{x}_{i,v} = v_{\max}^{(i)}, \quad \forall i \quad (2a)$$

Ensure consistency of the unary encoding of $x_{i,v}$:

$$\bar{x}_{i,v} \vee x_{i,v-Z}, \quad \forall i, v \in K_{E_i} \setminus \{v_{\min}^{(i)}\} \quad (2b)$$

Ensure consistency of the encoding of e_j :

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

Large-weight constraint:

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

Binary Encoding: General Idea

For each equation E_i , the modulo Z constraint can be rewritten using an integer $q \in \mathbb{N}$ such that:

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

Binary Encoding: General Idea

For each equation E_i , the modulo Z constraint can be rewritten using an integer $q \in \mathbb{N}$ such that:

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

The quotient q is represented in binary using auxiliary variables $x_{i,\ell}$:

$x_{i,\ell} = 1 \iff$ the ℓ -th bit of the binary representation of q is equal to 1.

Binary Encoding: General Idea

For each equation E_i , the modulo Z constraint can be rewritten using an integer $q \in \mathbb{N}$ such that:

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

The quotient q is represented in binary using auxiliary variables $x_{i,\ell}$:

$x_{i,\ell} = 1 \iff$ the ℓ -th bit of the binary representation of q is equal to 1.

Previous example: $Z = 3$, $s_i = 1$, $K_{E_i} = \{4, 7\}$

$$4 = 1 + 3 \times 1, \quad 7 = 1 + 3 \times 2.$$

Thus, the quotient q can take the values 1 or 2.

- The variables $x_{i,\ell}$ represent the bits of the quotient $q = \sum_{\ell=0}^{L_i-1} 2^\ell x_{i,\ell}$

Binary Encoding

- The variables $x_{i,\ell}$ represent the bits of the quotient $q = \sum_{\ell=0}^{L_i-1} 2^\ell x_{i,\ell}$

Force the congruence modulo Z :

$$\sum_{j \in V_{E_i}} H_{i,j} e_j + Z \sum_{\ell=0}^{L_i-1} 2^\ell \bar{x}_{i,\ell} = s_i + Z \sum_{\ell=0}^{L_i-1} 2^\ell, \quad \forall i \quad (3a)$$

Ensure consistency of the encoding of e_j :

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

Large-weight constraint:

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

Binary Encoding: Exhaustive Filtering

- The admissible values of the quotient q are:

$$q_{\min}^{(i)} \leq q \leq q_{\max}^{(i)}.$$

- All other values are forbidden:

$$I^{(i)} = \{0, \dots, q_{\min}^{(i)} - 1\} \cup \{q_{\max}^{(i)} + 1, \dots, 2^{L_i} - 1\}.$$

Binary Encoding: Exhaustive Filtering

- The admissible values of the quotient q are:

$$q_{\min}^{(i)} \leq q \leq q_{\max}^{(i)}.$$

- All other values are forbidden:

$$I^{(i)} = \{0, \dots, q_{\min}^{(i)} - 1\} \cup \{q_{\max}^{(i)} + 1, \dots, 2^{L_i} - 1\}.$$

Forbid each value $v \in I^{(i)}$:

$$\bigvee_{\substack{0 \leq \ell \leq L_i - 1 \\ b_{i,\ell}(v) = 0}} x_{i,\ell} \vee \bigvee_{\substack{0 \leq \ell \leq L_i - 1 \\ b_{i,\ell}(v) = 1}} \bar{x}_{i,\ell} \quad \forall i, v \in I^{(i)} \quad (4)$$

Example: $v = 12$, $b_i(v) = (1, 1, 0, 0)$

$$\bar{x}_{i,3} \vee \bar{x}_{i,2} \vee x_{i,1} \vee x_{i,0}$$

- One clause is added for each forbidden value of q

Goal: Avoid enumerating all forbidden values of q one by one

Principle:

- We directly analyze the binary representation of $q_{\max}^{(i)}$ and $q_{\min}^{(i)}$.
- We introduce auxiliary prefix variables $p_{i,j}^{\max}$ and $p_{i,j}^{\min}$.
- These variables make it possible to forbid values $q > q_{\max}^{(i)}$ and $q < q_{\min}^{(i)}$ without enumerating them one by one.

Outline

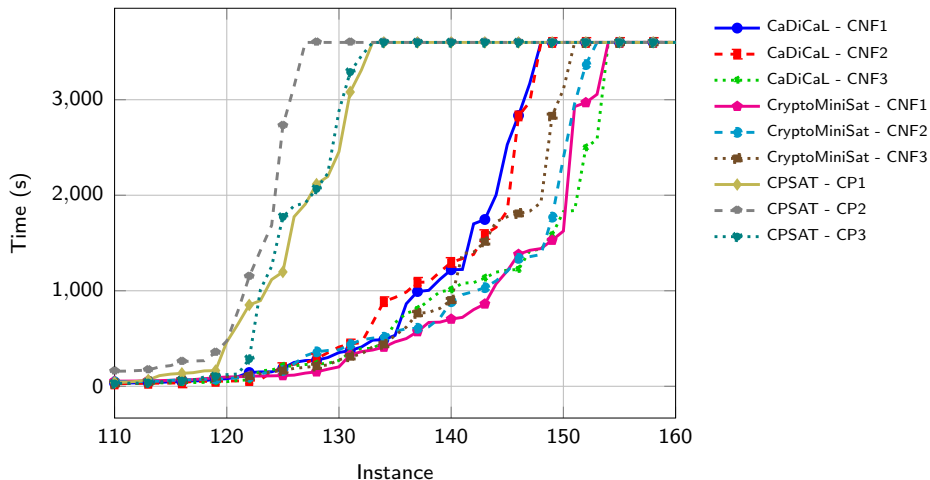
- 1 Syndrome decoding
- 2 Satisfiability
- 3 Encodings
- 4 Experimental Evaluation**
- 5 Conclusion and Future Work

Experimental Setup

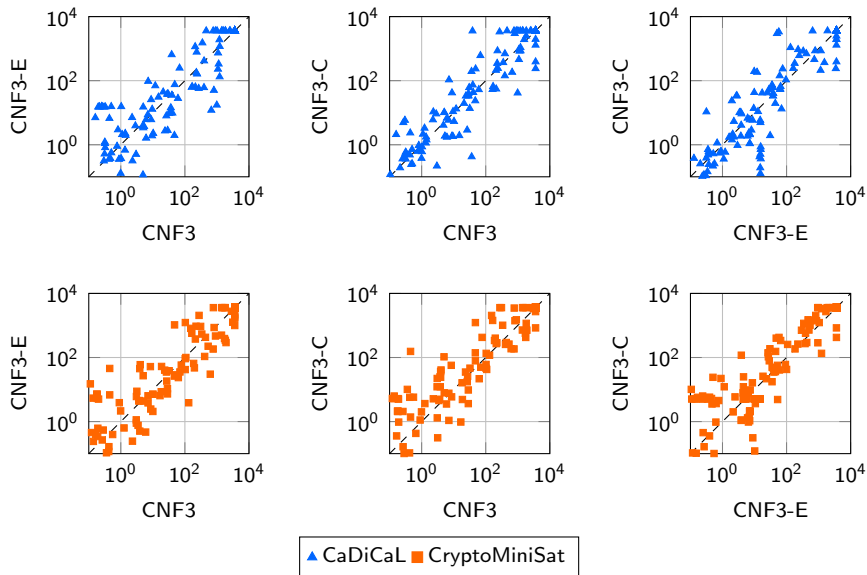
- **Instances:** Taken from the Large Weight Ternary Syndrome Decoding Challenge ¹ with $n \in \{10, \dots, 100\}$
- **Evaluated models:** CNF1, CNF2, CNF3, CNF3-E, CNF3-C, as well as their CP-SAT versions
- **Solvers:** CryptoMiniSat5 and CaDiCaL 2.1.3 for the SAT models, and OR-Tools CP-SAT for the CP models
- **Encodings:** CardNetwork for cardinality constraints, BinMerge for PB constraints, using PySAT
- **Environment:** MatriCS platform, bigmem partition, dual-socket Intel Xeon E5-2680 v4 servers with 512 GB of RAM
- **Time limit:** 3600 seconds per instance

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

Results



Results



Outline

- 1 Syndrome decoding
- 2 Satisfiability
- 3 Encodings
- 4 Experimental Evaluation
- 5 Conclusion and Future Work

Conclusion:

- CNF encodings of LWZSD over $\mathbb{F}_Z$
- Experiments on ternary instances ($Z = 3$)
- SAT solvers are the most robust on difficult instances
- The tested filtering methods do not provide significant improvements

Conclusion:

- CNF encodings of LWZSD over $\mathbb{F}_Z$
- Experiments on ternary instances ($Z = 3$)
- SAT solvers are the most robust on difficult instances
- The tested filtering methods do not provide significant improvements

Future work:

- Dedicated SAT heuristics for LWZSD
- More compact encodings exploiting the structure of the equations
- Incomplete or hybrid methods to improve scalability

Thank you for your attention!

carl.berton@u-picardie.fr



n	CNF1		CNF2		CNF3		CNF3-E		CNF3-C	
	Variables	Clauses	Variables	Clauses	Variables	Clauses	Variables	Clauses	Variables	Clauses
10	531	1124	369	812	437	952	437	962	445	975
20	2451	5506	1797	4116	1898	4327	1898	4379	1922	4398
30	6366	14249	4803	10899	4564	10260	4564	10384	4601	10387
40	12337	27306	9460	21032	8428	19031	8428	19258	8491	19249
50	22503	48039	17049	368901	14291	31739	14291	32044	14366	32011
60	35921	75354	27300	58625	22505	48765	22505	49283	22590	49102
70	50530	106383	38682	83096	31624	67846	31624	68621	31743	68323
80	71133	150217	54866	116820	43581	93393	43581	94388	43740	94042
90	96538	202353	74821	156641	57257	123030	57257	124141	57439	123763
100	124336	257438	96230	198815	71746	154296	71746	155511	71937	155078

Table: Average number of variables and clauses for each CNF encoding over 20 seeds.