Efficient Assistance to LDPC Code-based Erasure Recovery in NVM Storage

Anxiao (Andrew) Jiang

In collaboration with: Pulakesh Upadhyaya (TAMU), Ying Wang (TAMU, Qualcomm), Krishna Narayanna (TAMU), Hongchao Zhou (Shandong University), Jin Sima (Caltech), Jehoshua Bruck (Caltech)
Stopping Set
Stopping Set Elimination Problem

Given a stopping set, how to remove the fewest erasures from it, so that the remaining erasures become decodable by BP decoding?

Minimum Elimination Set
Remaining erasures become decodable
Stopping Set Elimination Problem

Applications

- Distributed Storage
- Deep Learning Based storage
Distributed storage

```
011010010010011110101010010
111100111001110111101110000
110000111000010111110011011
```

```
111100111001110111101110000
110000111000010111110011011
```

...
Distributed storage

Each column is an LDPC codeword
Distributed storage

When stopping sets appear in decoding, we can contact remote mirror data center for data. But we need to minimize remote-transmission cost.
Deep Learning based Storage in NVM

Example: LDPC codeword for text:
011010010010011110101010010 …..

When stopping sets appear in a text file, we can use NLP to help correct erasures. But we need to minimize NLP-decoding cost.


Y. Wang, K. R. Narayanan and A. Jiang, Exploiting Source Redundancy to Improve the Rate of Polar Codes, in ISIT 2017.

Y. Wang, K. R. Narayanan and A. Jiang, Exploiting Source Redundancy to Improve the Rate of Polar Codes, in ISIT 2017.
Stopping Set Elimination Problem

Applications

NP Hard
Stopping Set Elimination Problem

Applications

A reduction using:
Set Cover Problem
Pseudo Set Cover Problem

NP Hard
Stopping Set Elimination Problem is NP Hard
Stopping Set Elimination Problem is NP Hard

Pseudo Set Cover Problem

SSE Problem
Stopping Set Elimination Problem is NP Hard

Pseudo Set Cover Problem

SSE Problem
Stopping Set Elimination Problem is NP Hard

Pseudo Set Cover Problem

SSE Problem
Stopping Set Elimination Problem is NP Hard

Pseudo Set Cover Problem

SSE Problem
Stopping Set Elimination Problem is NP Hard

Pseudo Set Cover Problem

SSE Problem
Stopping Set Elimination Problem is NP Hard

Pseudo Set Cover Problem

SSE Problem
Stopping Set Elimination Problem is NP Hard

Pseudo Set Cover Problem

SSE Problem
Stopping Set Elimination Problem is NP Hard

Pseudo
Set Cover
Problem

SSE
Problem
Stopping Set Elimination Problem

Applications

NP Hard

Generalization: SSE for decoding in k BP iterations
Stopping Set Elimination Problem

Applications

NP Hard

Generalization:
SSE for decoding in k BP iterations

Even if k=1

NP Hard
SSE Problem is NP Hard even if k=1

Not-all-equal SAT Problem

Let $x_1, x_2, \ldots, x_n$ be $n$ Boolean variables. A literal is either $x_i$ or $\bar{x}_i$ (namely, the NOT of $x_i$) for some $i \in \{1, 2, \ldots, n\}$. Let a clause be a set of three literals. Let

$$S = \{C_1, C_2, \ldots, C_k\}$$

be a set of $k$ clauses. The question is: Is there a truth assignment to the $n$ Boolean variables such that for every clause in $S$, the three literals in the clause are neither all true nor all false (namely, every clause has at least one true literal and also at least one false literal)? (If the answer is “yes”, the problem is called “satisfiable”.)
SSE Problem is NP Hard even if k=1

Not-all-equal SAT Problem

Example Consider the following instance of the Not-all-equal SAT Problem. Let \( n = 4 \) and \( k = 5 \). Let the Boolean variables be \( x_1, x_2, x_3, x_4 \), and let the set of clauses be \( C_1 = (x_1, \bar{x}_2, x_3) \), \( C_2 = (\bar{x}_1, \bar{x}_2, x_4) \), \( C_3 = (x_2, x_3, x_4) \), \( C_4 = (x_1, \bar{x}_3, \bar{x}_4) \), \( C_5 = (\bar{x}_1, x_2, x_3) \).

The above instance is satisfiable because we can let the truth assignment be

\[
x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 1, x_5 = 1.
\]

Correspondingly, the clauses become \( C_1 = (1, 0, 0) \), \( C_2 = (0, 0, 1) \), \( C_3 = (1, 0, 1) \), \( C_4 = (1, 1, 0) \), \( C_5 = (0, 1, 0) \). None of the clauses is \((1, 1, 1)\) (namely, all true) or \((0, 0, 0)\) (namely, all false). \( \square \)
SSE Problem is NP Hard even if $k=1$

Not-all-equal SAT Problem

$C_1 = (x_1, x_3, \bar{x}_4), \quad C_2 = (x_1, \bar{x}_2, \bar{x}_3)$

SSE Problem with $k=1$
SSE Problem is NP Hard even if $k=1$

The Not-all-equal SAT Problem is satisfiable if and only if the corresponding SSE Problem (with $k=1$) has a solution (an elimination set) of size:

$$A + 3B$$

$A$: number of variables in the Not-all-equal SAT Problem

$B$: number of clauses in the Not-all-equal SAT Problem
SSE Problem is NP Hard even if k=1

Not-all-equal SAT Problem

\[ C_1 = (x_1, x_3, \overline{x}_4), \quad C_2 = (x_1, \overline{x}_2, \overline{x}_3) \]

X1=1, x2=1, x3=0, x4=1

SSE Problem with k=1
SSE Problem is NP Hard even if $k=1$

Not-all-equal SAT Problem

$$C_1 = (x_1, x_3, \bar{x}_4), \ C_2 = (x_1, \bar{x}_2, \bar{x}_3)$$

$x_1=1, \ x_2=1, \ x_3=0, \ x_4=1$

SSE Problem with $k=1$
Stopping Set Elimination Problem

Applications

NP Hard

Generalization:
SSE for decoding in k BP iterations

k=1

Approximation algorithm

NP Hard
Stopping Set Elimination Problem

Applications

NP Hard

Generalization: SSE for decoding in k BP iterations

NP Hard

Approximation algorithm

Optimal algorithms for trees

Linear time
Optimal Algorithms for Stopping Sets that form Trees
Stopping Set Elimination Problem

Applications

NP Hard

Optimal algorithms for trees

Generalization: SSE for decoding in $k$ BP iterations

Approximation algorithm

NP Hard

Approximation and optimal algorithms for cyclic graphs