Erasure Decoding in 2D (1,3)-RLL Constraint

Putranto H. Utomo[1], Tri A. Kusmayadi[2], Diari Indriati [3], and Titin S. Martini[4]

[1]Depeartment of Mathematics, Faculty of Mathematics and Natural Sciences, Sebelas Maret University, putranto@staff.uns.ac.id
[2]tri.atmojo.kusmayadi@staff.uns.ac.id
[3]diari.indri@staff.uns.ac.id
[4] titinsmartini@staff.uns.ac.id

Abstract

In information theory, the run length limited (RLL) constraint is one of the topics that many researchers have worked on. It has many benefits in designing a code to reduce and correct errors. The 1D (d,k)-RLL constraint is a binary sequence satisfying the number of a consecutive 0's is at most k and the number of consecutive 0's is at least d. Looking at the trend of the digital era, every year, there is a significant increase in data produced. Scientists, especially in the field of physics and computer science, are trying to find the answer to this question: How do we store data in a more efficient way? Lately, there has been a breakthrough in storing digital data, that is, by means of holographic recording, which uses the 2D format to store data inside a crystal. Theoretically, one could store more data per unit square of area. By storing data as a picture (2D format), there is a possibility that data could be distorted horizontally and vertically. Hence, we need to expand the theory in 1D constrained sequence to 2D constrained array. In this paper, we analyze the case of 2D (1,3)-RLL constraint from erasure decoding point of view

Erasure decoding, RLL, error correcting codes

1 Introduction

As one see in [1], the 1D (d,k)-RLL constraint is a binary sequence satisfying

1. There should be at least d 0's between 1's.
2. There should be at most k 0's between 1's.

From the practical point of view, the 1D (d,k)-RLL has benefit to reduce the errors in timing for storage system, such as magnetic tape, compact disk (CD), digital versatile disk (DVD), and also hard drive. The theory of 1D (d,k)-RLL constraint has flourist and eveloped very well in the last decades because there exist mathematical background theory to support the development. Unfortunately, that is not the case for 2D (d,k)-RLL constraint. It is very difficult to analyze the two dimension case. What one could do is by analyzing case-by-case.

In this paper we consider one case of 2D (d,k)-RLL cosntraint, that is 2D (1,3)-RLL constraint, We assume that the 2D array has same number of columns and rows. Hence, we are analyzing the collection of $n \times n$ arrays satisfying the following two condition, namely

- The number of 0's between 1's is at least 1 in every columns and in every rows.
- The number of 0's between 1's is at most 3 in every columns and in every rows

Now, suppose that an erasure has happened, that is some of the element are erased. The process of reconstructing eresured array is called erasure decoding. We analyze 2D (1,3)-RLL constraint from erasure decoding point of view.

In particular, we are interested in decoding an erasured array satisfying the 2D (1,3)- RLL constraint. Ideally we want to find an efficient decoder in solving the problem. Unfortunately, there are yet exist a polynomial complexity for the decoder. Therefore we use the following three approach for the decoder, that is brute force approach, SAT solver, and SMT solver.

In the brute force approach, we try to find an array satisfying all the condition/requirement. The idea is by checking one-by-one arrays in its domain. Clearly that this approach is not efficient and has exponential growth. The idea of SAT solver in by transforming the problem into satisfiability problem, that is the problem of finding a solution of a boolean formula written in CNF formula. The last approach is by transforming the problem in a bit-vector format, using satisfiability modulo theory (SMT) [3]. Then we use a SMT solver to solve the problem.

2 2D (1,3)-RLL Constraint from erasure coding point of view

Suppose we have an $n \times n$ array satisfying 2D (1,3)-RLL constraint. Next, we send the array over a noisy channel where it is possible to have some of its elements get erased. At the receiving end, we want to be able recover the erased elements. The process of recovering the array is called erasure decoding.

Let us consider the brute force approach. Suppose that we have an $n \times n$ array where some of its elements are erased. The idea of this approach is by enumerating all $n \times n$ array in $F_2^{n\times n}$ and check whether it satisfy the received erased array.

One could think an improvement for the brute force approach, that is designing back-track search. This approach is an improvement of brute force approach, that is instead of enumerating all possible arry in $F_2^{n\times n}$, we consider guessing each place of the erased element. If there is a contradiction, we go back to previous step and change the initial guess. This approach has the same complexity with brute force approach, but it is very memory efficient, since it does not have to enumerate all the $n \times n$ arrays.

Another approach we like to consider is by using SAT solver. This approach require the problem to be transformed into a logical expression. Hence, each element in the array correspond to a binary variable. Furthermore, we set the corresponding variable to "true" if the element is equal to one, and "false" if the element is equal to zero. Recall that there are two constraints regarding the problem, that is

1. There should be at least one 0's between 1's
2. There should be at most three 0's between 0's

Let us consider the first constraint. A row or a column will not satisfy the first constraint if there are 2 consecutive ones. Hence, if x_1 and x_2 represent two consecutive cell,

$$\neg(x_1 \land x_2)$$

should be true. Now let us consider a vector $(x_1, x_2, \dots, x_n)$, then

$$\bigwedge_{i=1}^{n-1}[\neg(x_i \wedge x_{i+1})\,]$$

ensure that there are no two consecutive 1's in that particular vector.

For the second constraint, there should be at most three 0's between 1's. Let us consider 4 consecutive cell represented by variables (x_1, x_2, x_3, x_4) in the array. In order to satisfy the second constraint, one of the variables should be "1". Hence the following expression has to be true:

$$x_1 \vee x_2 \vee x_3 \vee x_4.$$

Therefore, for a vector of length n, the corresponding logical expression is

$$\bigwedge_{i=1}^{n-3}(x_i \wedge x_{i+1} \wedge x_{i+2} \wedge x_{i+3})$$

Hence the corresponding logical expression for our problem are

$$\bigwedge_{j=1}^{n}\left[\bigwedge_{i=1}^{n-1}[\neg(x_{ij} \wedge x_{(i+1)j})]\right],$$

$$\bigwedge_{i=1}^{n}\left[\bigwedge_{j=1}^{n-1}[\neg(x_{ij} \wedge x_{i(j+1)})]\right],$$

$$\bigwedge_{j=1}^{n}\left[\bigwedge_{i=1}^{n-3}[(x_{ij} \wedge x_{(i+1)j} \wedge x_{(i+2)j} \wedge x_{(i+3)j})]\right],$$

$$\bigwedge_{i=1}^{n}\left[\bigwedge_{j=1}^{n-3}[(x_{ij} \wedge x_{i(j+1)} \wedge x_{i(j+2)} \wedge x_{i(j+3)})]\right].$$

The last approach is by SMT Solver. Technically, we could use SMT Solver to solve the boolean expression constructed in previous section, but previous research stated that it will be much faster if we use bit-vector representation compared to the boolean expression [2]. In our previous research, we solve binary puzzle using SMT Solver [4].

3 Conclusion

We conclude that the problem of erasure decoding can be approached my means of satisfiability (SAT) problem. Using the modern SAT solver and SMT solver, we could solve the NP Complete problem in a reasonable amount of time.

References

[1] Marcus, Brian H. and Roth, Ron M. and Siegel, Paul H. An Introduction to Coding for Constrained Systems https://www.math.ubc.ca/~marcus/Handbook/ index.html

[2] Barrett, Clark and Sebastiani, Roberto and Seshia, Sanjit A. and Tinelli, Cesare. Satisfiability Modulo Theories, Handbook of Satisfiability, chapter 12, pages 737-797, 2008. https://people.eecs.berkeley.edu/~sseshia/pubdir/SMT-BookChapter.pdf
[3] Barrett, Clark and Fontaine, Pascal and Tinelli, Cesare. The Satisfiability Modulo Theories Library (SMT-LIB) www.SMT-LIB.org
[4] Utomo, Putranto. Satisfiability modulo theory and binary puzzle. Journal of Physics: Conference Series 855(1):012057. 2017.