Solving systems of XOR equations

Jules Massart

October 2023

1 The System

Let $(n, m) \in \mathbb{N}$.

Let $(y_1, y_2, \ldots, y_n) \in \mathbb{N}^n$ and (S) be the following system in $\mathbb{N}^m$:

$$\begin{aligned} a_{1,1}x_1 \oplus a_{1,2}x_2 \oplus \cdots \oplus a_{1,m}x_m &= y_1 \\ a_{2,1}x_1 \oplus a_{2,2}x_2 \oplus \cdots \oplus a_{2,m}x_m &= y_2 \\ &\vdots \\ a_{n,1}x_1 \oplus a_{n,2}x_2 \oplus \cdots \oplus a_{n,m}x_m &= y_n \end{aligned}$$

with $a_{i,j} \in \{0, 1\}$.

$s = \min_{i \in \mathbb{N}^*} \left\{ i \,\middle|\, 2^i \geq y_k,\ k \in [1, n] \right\}$.

Let $Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$ can be represented by the matrix:

$$M = \begin{bmatrix} b_{y_1,1} & b_{y_1,2} & \cdots \\ b_{y_2,1} & b_{y_2,2} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

where $b_{y_i,j}$ represents the j-th bit of y_i (indexed from 1).

Let's work in the field $\mathbb{F}_2$ from now on.

Since XOR is a bit-by-bit operation, the system (S) can be decomposed into m systems on each column of M, where $A = (a_{i,j})_{1 \leq i \leq n, 1 \leq j \leq m}$. Furthermore, let's assume that $\text{rank}(A) = m$.

Let $(L_1, L_2, \ldots, L_s)$ denote the columns of M.

Let's solve $A \cdot X_1 = L_1$. This can be done using Gaussian elimination. The same procedure applies for $X_2, X_3, \ldots, X_s$ to produce the matrix $R = [X_1, X_2, \ldots, X_s]$.

Going back in R, for all $i \in [1, m]$, $x_i = \sum_{k=1}^{s} 2^k \cdot R_{i,k}$.

A Python implementation can be found on github.com/Herivelismus/xor_solver.