Sparse Boolean Equations and Circuit Lattices

Igor Semaev
University of Bergen
Norway

SHARCS, 9 September 2009

Problem

- X Boolean variable set, size n
- f_i Boolean polynomials in $X_i \subseteq X$
- Find all $0, 1$-solutions to

$$f_1(X_1) = 0, \dots, f_m(X_m) = 0$$

- were $|X_i| \leq l$ for small $l = 3, 4, \dots$
- No other restrictions
- E.g. TRIVIUM: 951 Boolean variables and equations
- each depends on 6 variables

Zakrevskij-Raddum Representation of Equations

- In [Zakrevskij,1999]
- Independently [Raddum,2004]
- $f_i(X_i) = 0 \Leftrightarrow$ solutions V_i in variables $X_i \Leftrightarrow E_i = (X_i, V_i)$
- E.g.

$$x_1x_2 + x_3 \equiv 0 \mod 2 \Leftrightarrow \begin{array}{ccccccc} x_1 & & 0 & 0 & 1 & 1 \\ x_2 & = & 0 & 1 & 0 & 1 \\ x_3 & & 0 & 0 & 0 & 1 \end{array}$$

- Solve $E_1, \ldots, E_m$ by guessing some variable values and check by Pairwise Agreeing (a kind of simplification, called local reduction by Zakrevskij and graph algorithm by Raddum)
- Combine equations by Gluing [Semaev,2005]

Agreeing-Gluing family

- Expected complexity much lower than worst case bounds, [Semaev,2007-08]
- Practically, Linear Algebra variant(MRHS) is far better than F4 in Magma
- E.g., more than 20000 times faster than F4 on AES-type random eqations with 48 Boolean variables, [Raddum-Semaev,2007]. MiniSat should be slow here, as the problem is not sparse in common sense
- Overcomes MiniSat in small sparsity, as 3,4,5, for randomly generated common sparse equations, [Schilling, in progress]
- Reasons for further development

Contribution Outline

- Graph equation representation and its simplification
- Pairwise equation modification with Agreeing2 method [Raddum-Semaev,2007]
- Agreeing2 with Circuit Lattices
- Reduced Circuit Lattices (require much low number of transistors)
- DES and TripleDES equation systems

Equation System Graph and Pairwise Agreeing

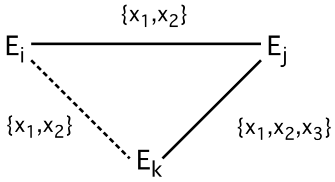
- **System**: $f_1(X_1) = 0, \ldots, f_m(X_m) = 0$
- Connect $E_i = (X_i, V_i)$ and $E_j = (X_j, V_j)$ by
- Edge labeled $X_i \cap X_j \neq \emptyset$

$$\mathrm{E_i} \;\overset{\mathrm{X_i} \,\cap\, \mathrm{X_j}}{\rule[0.5ex]{8em}{0.4pt}}\; \mathrm{E_j}$$

- **Pairwise Agreeing.** Let $Y \subseteq X_i \cap X_j$
- Learn ban $Y \neq a$ from E_i
- Expand to E_j through the edge
- Modify E_j accordingly

Obsolescent Edges

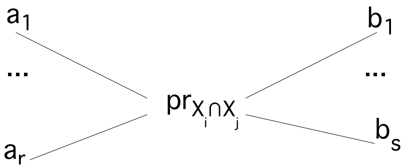
- Remove some edges and keep Algorithm's output
- E.g. $X_i = \{x_1, x_2, x_4\}, X_j = \{x_1, x_2, x_3, x_5\}, X_k = \{x_1, x_2, x_3\}$



- Left edges called maximal
- 16831 edges in Triple DES equation system initially
- 3929 maximal (left after removals)

Fast Pairwise Agreeing (Agreeing2 method)

- For maximal edges (E_i, E_j)
- $a_1, \ldots, a_r$ and $b_1, \ldots, b_s$ solutions to E_i and E_j with the same projection to $X_i \cap X_j$



- Pre-compute all $\{a_1, \ldots, a_r; b_1, \ldots, b_s\}$

Fast Pairwise Agreeing (Agreeing2 method)

- **Notation:** $a_i \neq$ part of a global solution $\Rightarrow$ mark $\bar{a}_i$
- Each tuple $\{a_1, \ldots, a_r; b_1, \ldots, b_s\}$ is equivalent to
- $\bar{a}_1, \ldots, \bar{a}_r \Rightarrow \bar{b}_1, \ldots, \bar{b}_s$ and $\bar{b}_1, \ldots, \bar{b}_s \Rightarrow \bar{a}_1, \ldots, \bar{a}_r$
- Solving the system:
- Introducing a guess $\equiv$ marking some of a_i
- Expand marking through the tuples

Example

- Equations

$$\begin{array}{c|ccc} & a_1 & a_2 & a_3 \\ \hline x_1 & 0 & 0 & 1 \\ x_2 & 0 & 1 & 1 \\ x_3 & 1 & 1 & 0 \end{array}, \quad \begin{array}{c|cc} & b_1 & b_2 \\ \hline x_1 & 0 & 1 \\ x_4 & 1 & 0 \end{array}, \quad \begin{array}{c|ccc} & c_1 & c_2 & c_3 \\ \hline x_2 & 0 & 1 & 1 \\ x_3 & 1 & 0 & 1 \\ x_4 & 1 & 1 & 0 \end{array}$$

- Tuples

$$\begin{array}{lll} \{a_1, a_2; b_1\}, & \{a_3; b_2\}, & \{b_1; c_2\}, \\ \{b_2; c_1, c_3\}, & \{a_1; c_1\}, & \{a_2; c_3\}, \\ \{a_3; c_2\} & & \end{array}$$

Example

- Assume $x_4 = 0 \Rightarrow b_1$ should be marked(not a solution part)
- Implies marking expansion

$$
\begin{tikzcd}
\{b_1; a_1,a_2\} \arrow[r] \arrow[d] & \{a_1; c_1\} \arrow[d] & & & \\
\{a_2 ; c_3\} \arrow[r] & \{c_1,c_3 ; b_2\} \arrow[r] & \{b_2 ; a_3\} \arrow[r] & \{a_3 ; c_2\} \arrow[r] & \{c_2 ; b_1\}
\end{tikzcd}
$$

- All instances(b_2 at early stage) got marked
- The system is inconsistent for $x_4 = 0$

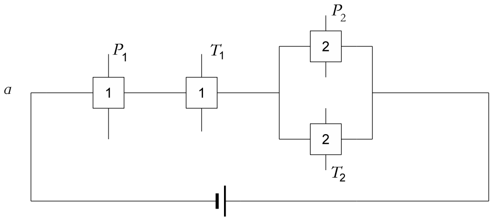
Circuit Lattice (Basic Construction)

- Circuit Lattice is a combination of switches and wires
- Two types of switches:

- 1-Switch controls vertical circuit by the horizontal
- 2-Switch controls horizontal circuit by the vertical

Horizontal circuits

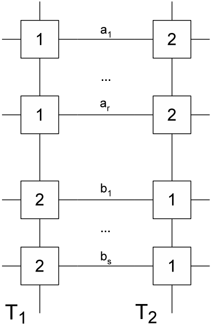
- Each local solution $a \in E_i$ determines one horizontal circuit
- 1-Switches are connected in series(or in parallel)
- 2-Switches are connected in parallel



- Endings are connected with a battery

Vertical circuits

- Tuple $T = \{a_1, \ldots, a_r; b_1, \ldots, b_s\}$ defines two vertical circuits



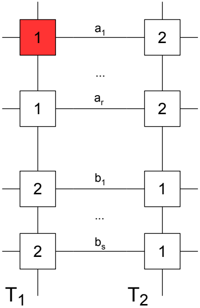
- Implement implications

$$\bar{a}_1, \ldots, \bar{a}_r \Rightarrow \bar{b}_1, \ldots, \bar{b}_s, \quad \bar{b}_1, \ldots, \bar{b}_s \Rightarrow \bar{a}_1, \ldots, \bar{a}_r$$

- All 1-Switches closed $\Rightarrow$ Voltage in the vertical circuit

Vertical circuits

- Tuple $T = \{a_1, \dots, a_r; b_1, \dots, b_s\}$ defines two vertical circuits



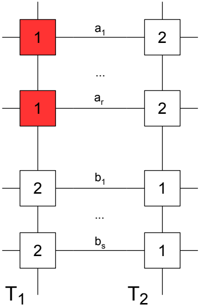
- Implement implications

$$\bar{a}_1, \dots, \bar{a}_r \Rightarrow \bar{b}_1, \dots, \bar{b}_s, \quad \bar{b}_1, \dots, \bar{b}_s \Rightarrow \bar{a}_1, \dots, \bar{a}_r$$

- All 1-Switches closed $\Rightarrow$ Voltage in the vertical circuit

Vertical circuits

- Tuple $T = \{a_1, \ldots, a_r; b_1, \ldots, b_s\}$ defines two vertical circuits



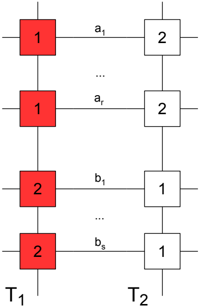
- Implement implications

$$\bar{a}_1, \ldots, \bar{a}_r \Rightarrow \bar{b}_1, \ldots, \bar{b}_s, \quad \bar{b}_1, \ldots, \bar{b}_s \Rightarrow \bar{a}_1, \ldots, \bar{a}_r$$

- All 1-Switches closed $\Rightarrow$ Voltage in the vertical circuit

Vertical circuits

- Tuple $T = \{a_1, \dots, a_r; b_1, \dots, b_s\}$ defines two vertical circuits



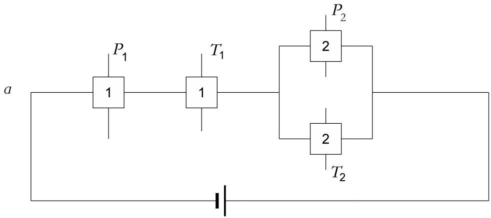
- Implement implications

$$\bar{a}_1, \dots, \bar{a}_r \Rightarrow \bar{b}_1, \dots, \bar{b}_s, \quad \bar{b}_1, \dots, \bar{b}_s \Rightarrow \bar{a}_1, \dots, \bar{a}_r$$

- All 1-Switches closed $\Rightarrow$ Voltage in the vertical circuit

How horizontal circuit works

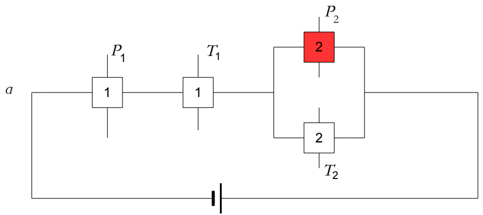
- $T = \{a, a_1, \dots, a_2; b_1, \dots, b_2\}$, $P = \{a, a_3, \dots, a_4; c_1, \dots, c_2\}$
- Define



- Voltage in vertical $P_2 \Rightarrow$ 2-Switch closed
- Voltage in circuit a (marking a)
- $\Rightarrow$ 1-Switches closed
- Voltage may appear in the vertical circuit T_1

How horizontal circuit works

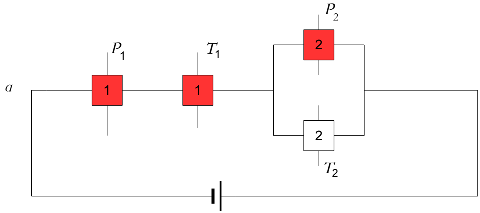
- $T = \{a, a_1, \ldots, a_2; b_1, \ldots, b_2\}$, $P = \{a, a_3, \ldots, a_4; c_1, \ldots, c_2\}$
- Define



- Voltage in vertical $P_2 \Rightarrow$ 2-Switch closed
- Voltage in circuit a (marking a)
- $\Rightarrow$ 1-Switches closed
- Voltage may appear in the vertical circuit T_1

How horizontal circuit works

- $T = \{a, a_1, \ldots, a_2; b_1, \ldots, b_2\}$, $P = \{a, a_3, \ldots, a_4; c_1, \ldots, c_2\}$
- Define

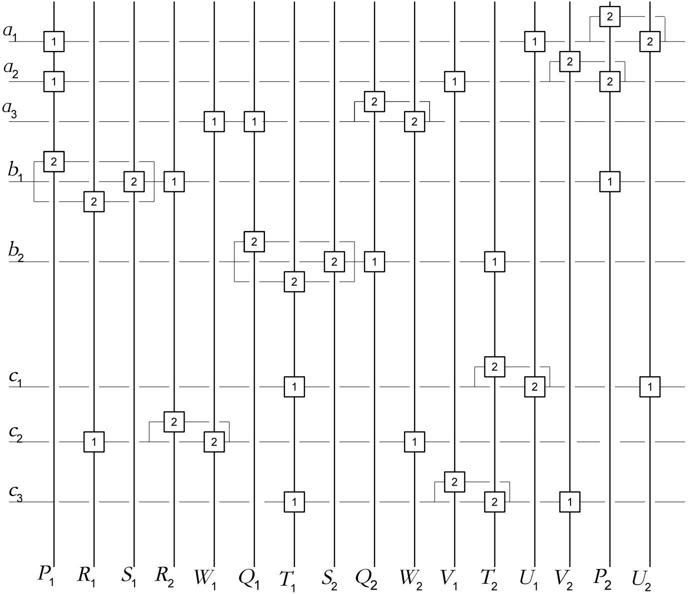


- Voltage in vertical $P_2 \Rightarrow$ 2-Switch closed
- Voltage in circuit a (marking a)
- $\Rightarrow$ 1-Switches closed
- Voltage may appear in the vertical circuit T_1

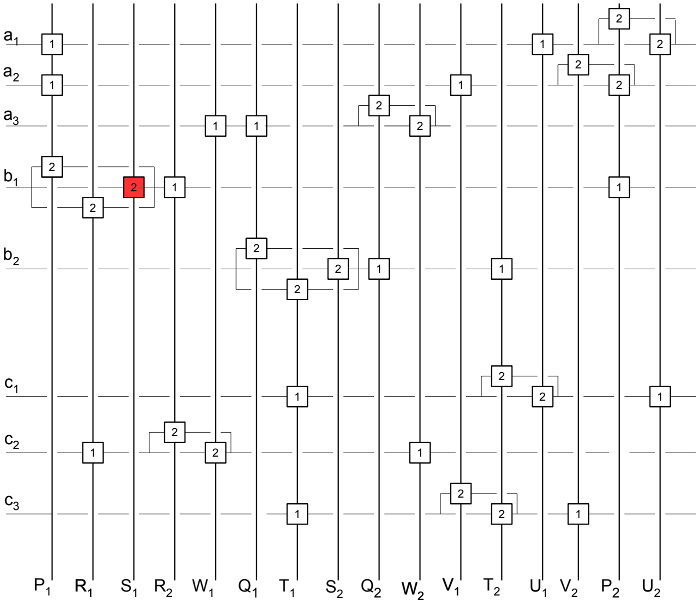
Introduce the guess

- Generally, no voltage in initial circuit lattice
- Assume E_i depends on x_j
- $a_1, \ldots, a_2$ solutions to E_i, where $x_j = 0$
- Add 2-Switch to each $a_1, \ldots, a_2$, connect them
- Guessing $x_j = 0$ is inducing voltage in new circuit
- Similarly, guessing $x_j = 1$
- s-variable guess - $2s$ new vertical circuits

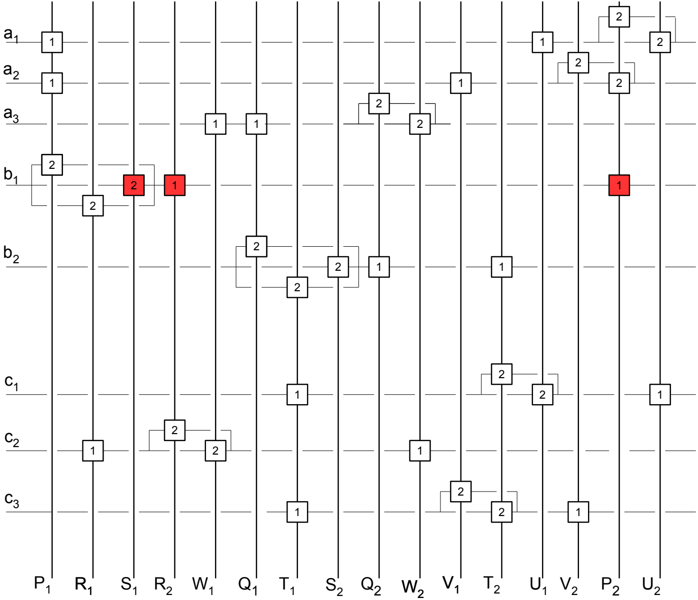
Exemplary circuit



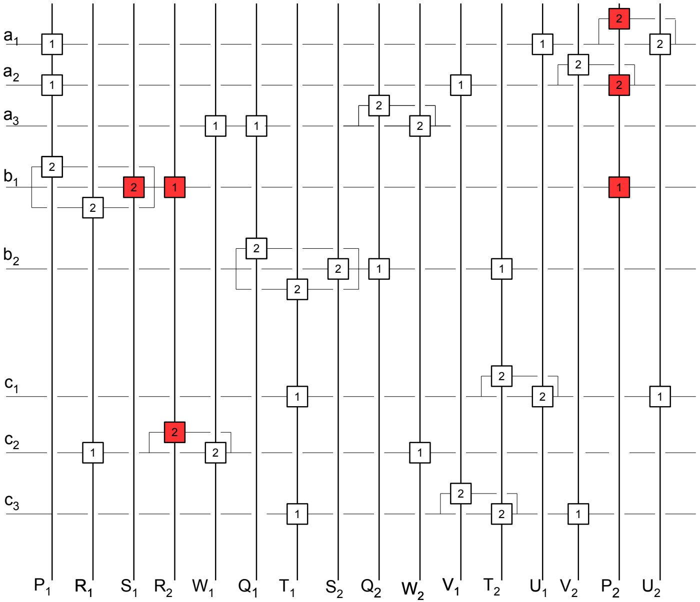
1st turn: introduce guess $x_4 = 0$



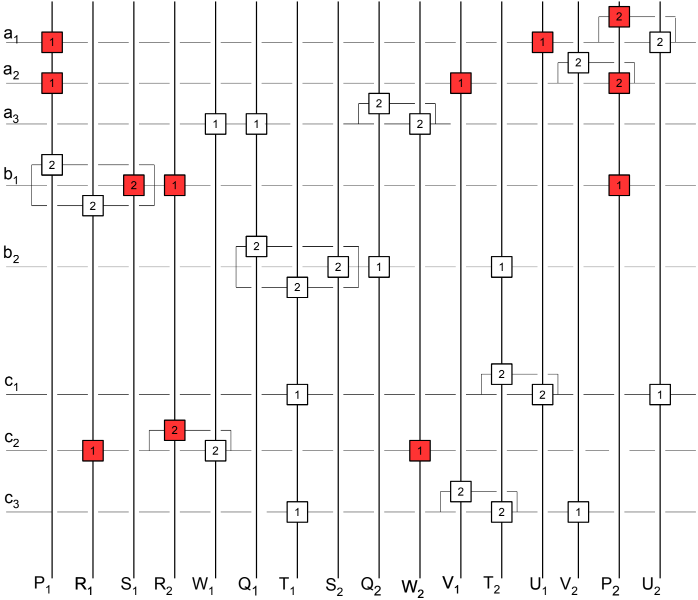
2nd turn



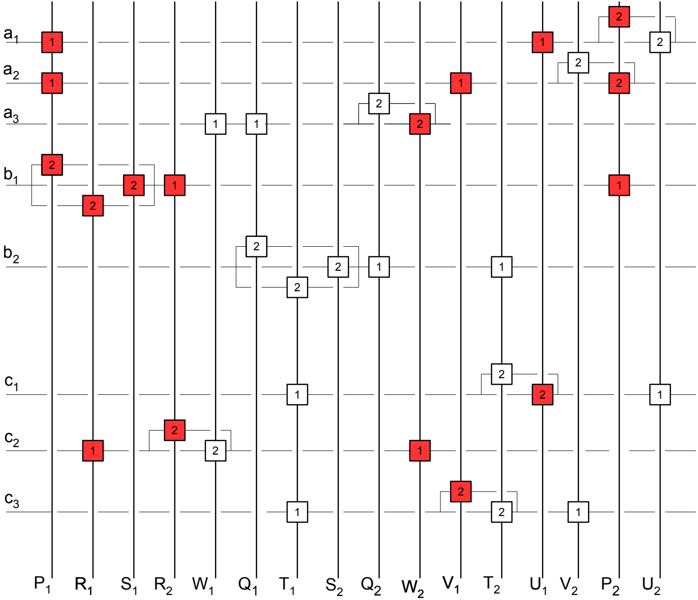
3rd turn



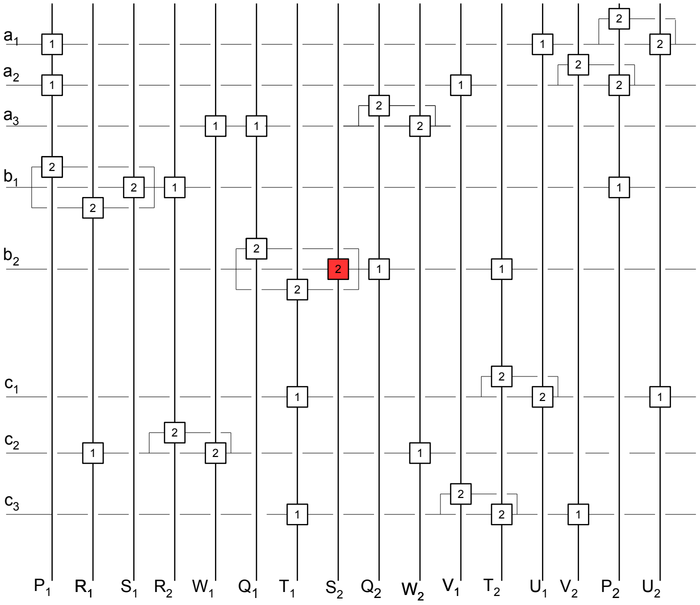
4th turn



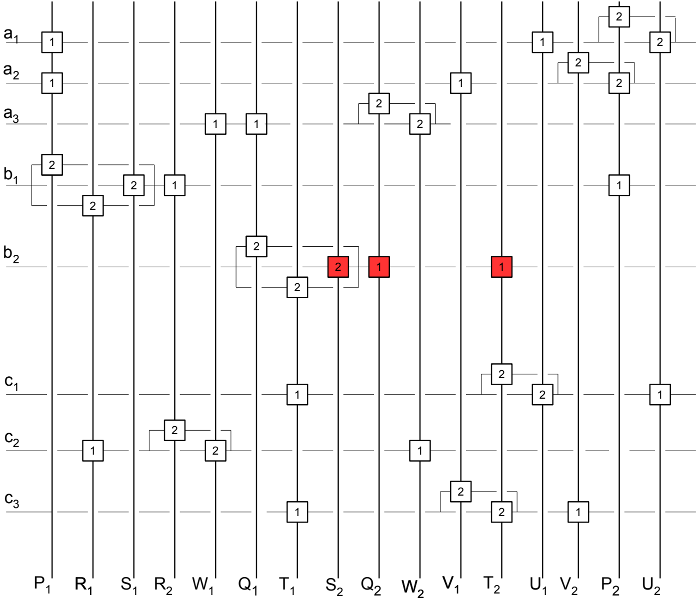
5th turn: Observe Inconsistence



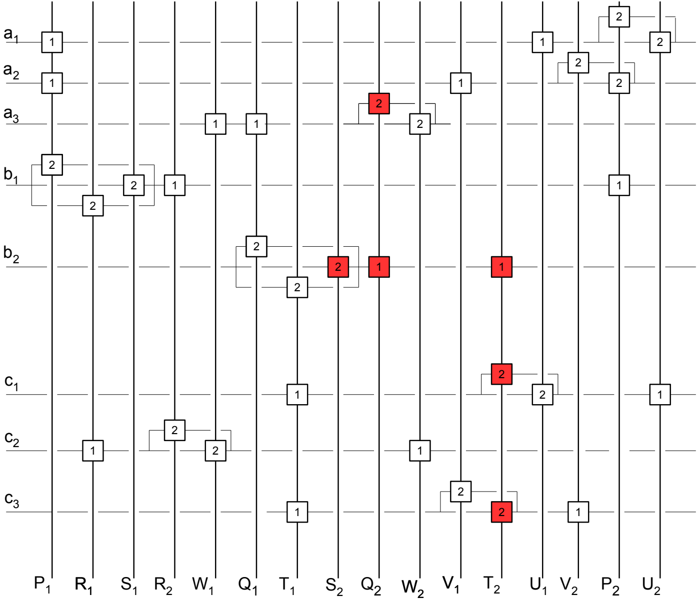
1st turn: introduce guess $x_4 = 1$



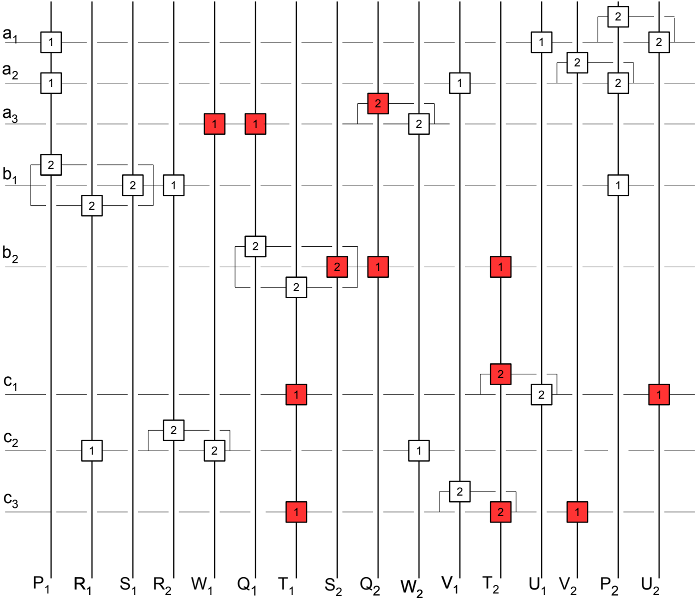
2nd turn


a1
a2
a3
b1
b2
c1
c2
c3
P1 R1 S1 R2 W1 Q1 T1 S2 Q2 W2 V1 T2 U1 V2 P2 U2
1
2

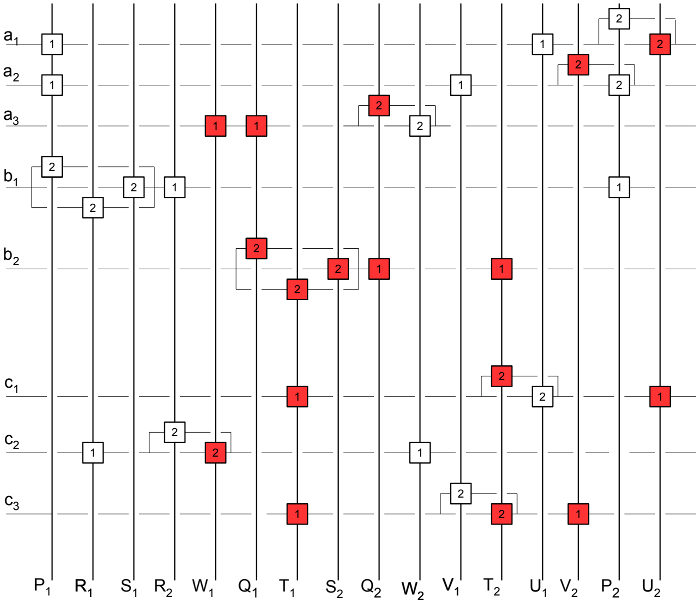
3rd turn



4th turn

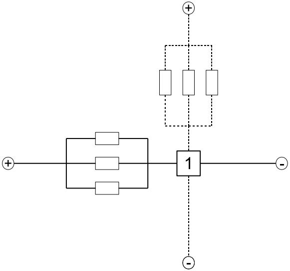


5th turn: Observe Inconsistence



Reduced Circuit Lattice

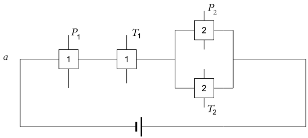
- Use Switches that control several circuits
- They may be controlled by any of several circuits



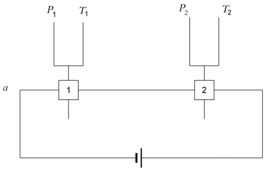
- 1-Switch: Any of horizontal circuits control all vertical circuits
- In 2-Switches, any vertical circuit controls all horizontals

Reduced Horizontal Circuits

- Transform horizontal circuit



- by using new switches to

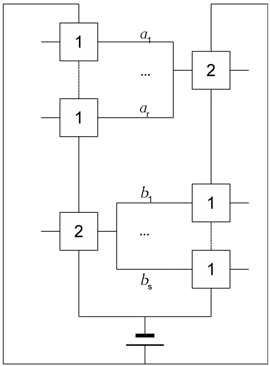


- Number of switches is now

$$2 \times (\text{number of local solutions}) = 2 \sum_i |V_i|$$

Reduced Vertical Circuits

- One 2-Switch can control several horizontal circuits



- Overall number of switches after the reductions

$$\sum_i |V_i| + 2 \sum_{tuples} 1$$

DES and TripleDES equations

- Variables:
- 64-bit plain-text, cipher-text,
- internal state blocks and 56(112)-bit key
- Equations from S-boxes as

$$Y_4 \oplus Z_4 = S(X_6 \oplus K_6)$$

- Each equation: 20 variables and 2^{16} RHS
- Study the system parameters

TripleDES system parameters

- 1712 variables, 384 equations
- 3929 maximal edges
- 71320 tuples
- 2.6×10^7 switches
- $480 = 2 \times 128 + 2 \times 112$ input contacts
- The device doesn't require synchronization

Implement on Modern Semiconductor Crystals for brute force?

- Transistor works as a switch
- 1.7×10^9 transistors on Dual-Core Itanium2 processor (2.6×10^7 required)
- Circuit Lattice speed $\leq 2\times$(number of rounds) transistor turns
- $2 \times 48 + 2$ turns for TripleDES
- One transistor turn, say 100GHz(1000GHz reported)
- 1GHz key-rejecting rate when using for brute force
- Reported(2006) 0.13GHz per chip with implementing encryption

Conclusions

- Using only maximal edges significantly economizes parameters
- Equation solving is shown as voltage expansion through a lattice of switches
- Our approach seems more flexible than implementing encryption as enables handling any equation system representing cipher
- Applications to DES, TripleDES are discussed