Solving SAT by Reducing to Max2XOR

Jordi Levy
IIIA, CSIC, Barcelona, Spain

Simons Institute SAT Reunion, Berkeley
June 15, 2022

Why Reducing to Max2XOR?

- Nice approximation algorithms, like for MaxCUT
- XOR is in P
- Reducing SAT to Max2SAT and using MaxSAT resolution we have polynomial proofs of the PHP
- Sherali-Adams, circular resolution, (negative) weighted MaxSAT resolution,...are equivalent
- 2XOR constraints are the simplest form (apart from MaxCUT) of Boolean constraints

$$
\begin{array}{ll}
x = y & \text{equiv. } x \oplus y = 0 \\
x \neq y & \text{equiv. } x \oplus y = 1 \\
x = 0 & \\
y = 1 &
\end{array}
$$

- Why not?

Ingredients

- A method to translate from SAT to Max2XOR
- A method to find assignments
- A method to find proofs
- ... and a way to integrate the last two

Gadget from SAT to 3SAT

$$x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6 \longrightarrow \left\{ \begin{array}{l} x_1 \vee x_2 \vee b_1 \\ \neg b_1 \vee x_3 \vee b_2 \\ \neg b_2 \vee x_4 \vee b_3 \\ \neg b_3 \vee x_5 \vee x_6 \end{array} \right.$$

Gadget from SAT to 3SAT

$$x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6 \longrightarrow \begin{cases} x_1 \vee x_2 \vee b_1 & \rightarrow 1 \\ \neg b_1 \vee x_3 \vee b_2 & \rightarrow 1 \\ \neg b_2 \vee x_4 \vee b_3 & \rightarrow 1 \\ \neg b_3 \vee x_5 \vee x_6 \end{cases}$$

Gadget from SAT to 3SAT

$$x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6 \longrightarrow \left\{ \begin{array}{l} x_1 \vee x_2 \vee b_1 \\ \neg b_1 \vee x_3 \vee b_2 \\ \neg b_2 \vee x_4 \vee b_3 \\ \neg b_3 \vee x_5 \vee x_6 \end{array} \right.$$

1

1

1

1

0

Gadget from SAT to 3SAT

$$x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6 \longrightarrow \begin{cases} x_1 \vee x_2 \vee b_1 \\ \neg b_1 \vee x_3 \vee b_2 \\ \neg b_2 \vee x_4 \vee b_3 \\ \neg b_3 \vee x_5 \vee x_6 \end{cases}$$

3

$\alpha - 1$ = sum of weights of satisfied constraints when original constraint is falsified

4

α = sum of weights of satisfied constraints when original constraint is satisfied

4

β = sum of weights of constraints

Gadget from SAT to 3SAT

$$x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6 \longrightarrow \begin{cases} x_1 \vee x_2 \vee b_1 \\ \neg b_1 \vee x_3 \vee b_2 \\ \neg b_2 \vee x_4 \vee b_3 \\ \neg b_3 \vee x_5 \vee x_6 \end{cases}$$

$\alpha - 1 \to 3$ = sum of weights of satisfied constraints when original constraint is falsified

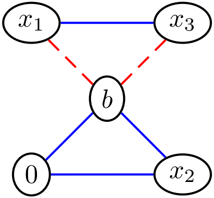
$\alpha \to 4$ = sum of weights of satisfied constraints when original constraint is satisfied

$\beta \to 4$ = sum of weights of constraints

For approximation we want smallest α

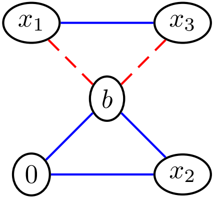
Now we want smallest $\beta - \alpha \to 0$

$$x_1 \vee x_2 \vee x_3 \longrightarrow \begin{cases} x_1 \neq x_3 \\ x_1 = b \\ x_3 = b \\ x_2 \neq b \\ b \neq 0 \\ x_2 \neq 0 \end{cases}$$



Gadget from 3SAT to Max2XOR

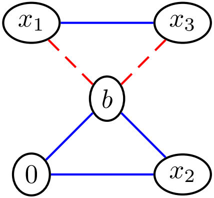
$$x_1 \vee x_2 \vee x_3 \longrightarrow \begin{cases} x_1 \neq x_3 \\ x_1 = b \rightarrow 1 \\ x_3 = b \rightarrow 1 \\ x_2 \neq b \\ b \neq 0 \\ x_2 \neq 0 \end{cases}$$



Setting $b = 0$

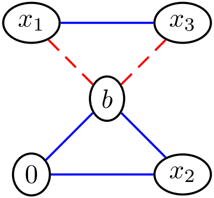
Gadget from 3SAT to Max2XOR

$$x_1 \vee x_2 \vee x_3 \;(x_3 \to 1) \longrightarrow \begin{cases} x_1 \neq x_3 & \to 1 \\ x_1 = b & \\ x_3 = b & \to 1 \\ x_2 \neq b & \to 1 \\ b \neq 0 & \to 1 \\ x_2 \neq 0 & \end{cases}$$



Setting $b = 1$

Gadget from 3SAT to Max2XOR



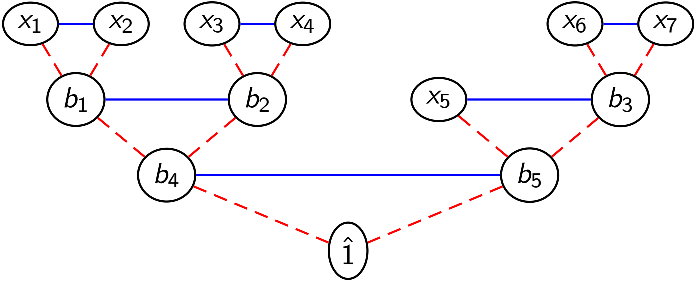
$$x_1 \vee x_2 \vee x_3 \longrightarrow \begin{cases} x_1 \neq x_3 \\ x_1 = b \\ x_3 = b \\ x_2 \neq b \\ b \neq 0 \\ x_2 \neq 0 \end{cases}$$

Too big

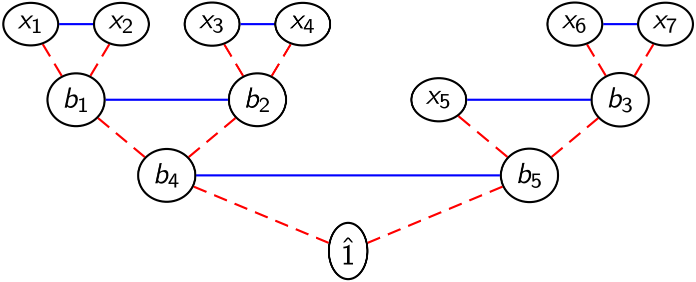
Composition introduces $k - 3 + k - 2 = 2k - 5$ variables

Direct Gadget from SAT to Max2XOR

$x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6 \vee x_7$



$x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6 \vee x_7$



Every dendogram of variables induces a reduction

$$\boxed{\boxed{x_1\ \ x_2}\ \boxed{x_3\ \ x_4}}\ \boxed{x_5\ \boxed{x_6\ \ x_7}}$$

Could something similar be done to reduce SAT to 3SAT?

Mixing Method for MaxCUT

MaxCUT: Given $G = (V, E)$,

$$\max \sum_{(i,j) \in E} 1/2 - 1/2\, x_i\, x_j$$

such that $x_i \in \{1, -1\} \quad \forall i \in V$

Mixing Method for MaxCUT

MaxCUT: Given $G = (V, E)$,

$$\max \sum_{(i,j)\in E} 1/2 - 1/2\, x_i\, x_j \longrightarrow y_{ij}$$

such that $x_i \in \{1, -1\} \quad \forall i \in V$

SDP relaxation [Goemans-Williamson]:

$$\max \sum_{(i,j)\in E} 1/2 - 1/2\, y_{ij}$$

such that $y_{ii} = 1 \; \forall i \in V$

$\{y_{ij}\}$is SDP

Mixing Method for MaxCUT

SDP relaxation [Goemans-Williamson]:

$$\max \sum_{(i,j)\in E} 1/2 - 1/2\, y_{ij}$$

such that $y_{ii} = 1 \ \forall i \in V$

$\{y_{ij}\}$ is SDP

$\exists \vec{u}_i \in \mathbb{R}^n$ such that $y_{ij} = \vec{u}_i \cdot \vec{u}_j$

Mixing Method [Wang et al.]:

$$\max \sum_{(i,j)\in E} 1/2 - 1/2\, \vec{u}_i \cdot \vec{u}_j$$

such that $|\vec{u}_i| = 1 \ \forall v \in V$

$\vec{u}_i \in \mathbb{R}^m$

with $m < n$

Use gradient-descent instead of a SDP solver

Use branch&bound on top to get an exact MaxCUT solver

Mixing Method for MaxCUT

Mixing Method [Wang et al.]:

$$\max \sum_{(i,j)\in E} 1/2 - 1/2\, \vec{u}_i \cdot \vec{u}_j$$

such that $|\vec{u}_i| = 1 \ \forall v \in V$ with $m < n$

$\vec{u}_i \in \mathbb{R}^m$

Use gradient-descent instead of a SDP solver

Use branch&bound on top to get an exact MaxCUT solver

Easily extensible to Max2XOR:

$$\max \sum_{x_i \oplus x_j = 1} 1/2 - 1/2\, \vec{u}_i \cdot \vec{u}_j + \sum_{x_i \oplus x_j = 0} 1/2 + 1/2\, \vec{u}_i \cdot \vec{u}_j$$

Consider a constant $\hat{0}$ with fixed vector $u_{\hat{0}}$ and $x = 0$ (resp. $x = 1$) equivalent to $x \oplus \hat{0} = 0$ (resp. $x \oplus \hat{0} = 1$)

Use decimation to fix variable values and **satisfied constraints**

Promissing results :-)

Proof System for Max2SAT

$$x_1 \vee \ldots \vee x_k \longrightarrow \begin{cases} x_i & i = 1, \ldots, k \\ \neg x_i \vee \neg b_i & i = 1, \ldots, k-1 \\ \neg x_{i+1} \vee b_i & i = 1, \ldots, k-2 \\ b_i \vee \neg b_{i+1} & i = 1, \ldots, k-2 \\ \text{replacing } b_{k-1} \text{ by } x_k & \end{cases}$$

Max2SAT resolution:

$$\frac{\begin{array}{c} x \vee y \\ \neg x \vee z \end{array}}{\begin{array}{c} y \vee z \\ x \vee y \vee \neg z \\ \neg x \vee \neg y \vee z \end{array}}$$

$$x_1 \vee \ldots \vee x_k \longrightarrow \begin{cases} x_i & i = 1,\ldots,k \\ \neg x_i \vee \neg b_i & i = 1,\ldots,k-1 \\ \neg x_{i+1} \vee b_i & i = 1,\ldots,k-2 \\ b_i \vee \neg b_{i+1} & i = 1,\ldots,k-2 \\ \text{replacing } b_{k-1} \text{ by } x_k & \end{cases}$$

Max2SAT resolution:

$$\frac{\begin{array}{c} x \vee y \\ \neg x \vee z \end{array}}{\begin{array}{c} y \vee z \\ x \vee y \vee \neg z \\ \neg x \vee \neg y \vee z \end{array}}$$

x

y

$\neg z$

$\neg x \vee \neg b$

$\neg y \vee \neg z$

$\neg y \vee b$

$b \vee z$

$(-1)\ \square$

Complete?

Short proof for PHP

Proof System for Max(2)XOR

Does it exist?

Does it exist for MaxCUT?

SAT solvers combine assignment and proof search

Could we do the same for Max2XOR?

IDEAS:

Time to consider stronger proof systems e.g. Sherali-Adams,...

Intuition: Relaxations (linearizations) where products of literals are "replaced" by new variables

Umpractical even for small (constant) size products: consider certain pairs of literals

Resolution Proof System for Max(2)XOR

$$\frac{\begin{array}{c} x \oplus A = k_1 \\ x \oplus B = k_2 \end{array}}{A \oplus B = k_1 \oplus k_2}$$

Resolution Proof System for Max(2)XOR

$$\frac{\begin{array}{cc} x \oplus A = k_1 & = 1 \\ x \oplus B = k_2 & = 1 \end{array}}{\begin{array}{cc} A \oplus B = k_1 \oplus k_2 & = 1 \end{array}}$$

Resolution Proof System for Max(2)XOR

$$
\begin{array}{c}
x \oplus A = k_1 \\
x \oplus B = k_2 \\
\hline
A \oplus B = k_1 \oplus k_2
\end{array}
\qquad
\begin{array}{cccc}
= 1 & 0 & 0 \\
= 1 & 1 & 0 \\
\hline
= 1 & 0 & 1
\end{array}
$$

#unsat not preserved

Resolution Proof System for Max(2)XOR

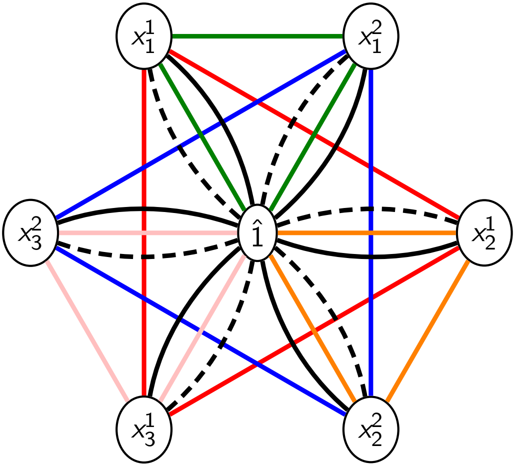
$$\begin{array}{rl}
x \oplus A = k_1 & = 1 \quad 0 \quad 0 \\
x \oplus B = k_2 & = 1 \quad 1 \quad 0 \\
\hline
A \oplus B = k_1 \oplus k_2 & = 1 \quad 0 \quad 1
\end{array}$$

#unsat not preserved

- We should add some constraints in the conclusion,... but we cannot express them as XOR
- For 2^k assignments there are 3^k SAT clauses, but only 2^k XOR clauses
- Given $f : \{0,1\}^n \to \mathbb{Q}^+$ like $f(a,b,c) = 2 + a\,(1-c) + c\,b$
  Expressable as (weighted) MaxSAT:
  $P_1 = \{\top,\ a \vee c,\ b \vee \neg c\}$
  $P_2 = \{a \vee b,\ c \vee a \vee \neg b,\ \neg c \vee b \vee \neg a\}$
  As MaxXOR:
  $P_3 = \{\ {}_{(3/2)}\,\top,\ {}_{(1/2)}\, a = 1,\ {}_{(1/2)}\, b = 1,$
  ${}_{(1/2)}\, x \oplus a = 1,\ {}_{(1/2)}\, x \oplus b = 0\}$

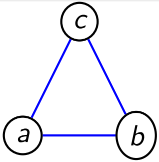
3 Pigeons and 2 Holes

$x_1^1 \vee x_1^2$
$x_2^1 \vee x_2^2$
$x_3^1 \vee x_3^2$
$\neg x_1^1 \vee \neg x_2^1$
$\neg x_2^1 \vee \neg x_3^1$
$\neg x_1^1 \vee \neg x_3^1$
$\neg x_1^2 \vee \neg x_2^2$
$\neg x_2^2 \vee \neg x_3^2$
$\neg x_1^2 \vee \neg x_3^2$
11 unsatisfiable constraints



Proof System a là Sherali-Adams

$$a \neq b$$
$$b \neq c$$
$$c \neq a$$



Number of unsatisfied:

$$a\,b + (1-a)(1-b) + b\,c + (1-b)(1-c) + a\,c + (1-a)(1-c) =$$
$$3 - 2(a+b+c-ab-bc-ac) \overset{?}{\geq} 1$$

Add $a+b+c-ab-bc-ac \leq 1$ as an axiom

$$a \neq b$$
$$a \neq c$$
$$a \neq d$$
$$b \neq c$$
$$b \neq d$$
$$c \neq d$$