CSC 2429 Winter 2018
Proof Complexity, Mathematical Programming, and Algorithms

Toni Pitassi
www.cs.toronto.edu/~toni

Course webpage: click on teaching
This course is about large classes of (sound) algorithms for solving NP-hard optimization problems (+ decision problems).

Marriage between complexity theory (proof complexity) and algorithms which has emerged over the last 10-15 years.

Very roughly: Proof system gives rise to a family of sound and feasible algorithms.

Can be used to rule out major approaches.

Can also be used to give new algorithms!

Extended Formulations

Automat. of sos proof upper bounds
Example 1  MaxSAT / MaxCSP

Given \( f = C_1 \land C_2 \land \ldots \land C_m \) over \( x_1 \ldots x_n \)
find an assignment to \( x_1 \ldots x_n \) that maximizes
the number of satisfied clauses.

Easy to formulate as an integer program:

Say \( f = (x_1 \lor x_2 \lor \overline{x}_3)(x_1 \lor \overline{x}_3)(x_1 \lor \overline{x}_2)(\overline{x}_1) \)

\[
\text{Max} \quad C_1 + C_2 + C_3 + C_4 \\
\text{st.} \quad x_1 + x_2 + (1-x_3) \geq \varsigma_1 \\
\quad x_1 + x_3 \geq \varsigma_2 \\
\quad x_1 + (1-x_2) \geq \varsigma_3 \\
\quad (1-x_1) \geq C_4 \\
\quad x_i, \varsigma_i \in \{0,1\},
\]
What happens if we relax constraints to
\[ 0 \leq x_i \leq 1 \quad 0 \leq \xi \leq 1 \]?

Then we have an LP so can solve in polytime (ellipsoid alg Khachian '79)

How to solutions compare?

For unsat 3CNF \( f \), integer OPT is \( \frac{7}{8}m \)

But LP has a fractional OPT of \( m \)

Our example: set \( x_1 = x_2 = x_3 = \frac{1}{2} \), \( \xi = \xi_2 = \xi_3 = \xi_4 = 1 \)

satisfies all constraints & has fractional OPT = 4
Note we could alternatively formulate MAXSAT as a decision problem.

Constraints are as above plus

\[ \xi_1 + \xi_2 + \xi_3 + \xi_4 \geq 4 \]

Over integers this system of inequalities is infeasible, but there are feasible solutions whenever \( 0 \leq \xi_i \leq 1 \).
Example 2: \textbf{MAX-CUT}

Let $G = (V, E)$, $V = \{v_1, \ldots, v_n\}$ with non-negative edge weights $w_{ij}$.

Find $S \subseteq V$ that maximizes $\text{cut}(S)$.

Easy to formulate as a \underline{quadratic program}:

\[
\text{Max } \frac{1}{2} \sum_{i < j} w_{ij} (1 - y_i y_j)
\]

\[y_i \in \{-1, 1\} \]
Relaxation of Quadratic Program to SDP:

\[
\text{Max } \frac{1}{2} \sum_{i<j} W_{ij} (1 - u_i^T u_j)
\]
\[
u \in \mathbb{R}^n, \quad \|u_i\| = 1
\]

Let \( X = \text{gram matrix of } u_1 \ldots u_n \)

So \( X = U^T U \) where \( U = \begin{bmatrix} u_1 & u_2 & \ldots & u_n \end{bmatrix} \)

Fact: \( X = U^T U \) is equivalent to \( X \) being positive-semidefinite \( \text{PSD} \) \( (z^T X z \geq 0 \; \forall z) \)

Thus relaxation becomes

\[
\text{Max } \frac{1}{2} \sum_{i<j} W_{ij} (1 - X_{ij})
\]
\[
X \geq 0 \text{ PSD}
\]
The relaxation is a SDP and can be solved in polynomial time (Ellipsoid Alg, Khachiyan '79)

Goesman-Williamson gave an amazing approx alg for MaxCUT:

1. Solve above SDP
2. apply randomized rounding to convert solution to 0/1 solution

They prove that the resulting (rounded) solution is always $\geq 0.872 \cdot \text{OPT}$
This course:

we study systematic techniques to improve the relaxations in order to get better algorithms.

Improvements involve:

adding new variables + new constraints to original set to cut away feasible fractional (non-integer) solutions

Need to keep all integer solutions (this is soundness)
Proof systems are systematic, sound ways to do this.

 Ones we will study:

 Resolution
 Poly Calculus/Nullsatz
 Sherali Adams, Cutting Planes
 SOS

 Viewing them as refutation systems corresponds to analyzing decision problems/feasibility.

 Viewing them as derivation systems corresponds to analyzing optimization problems.
New Algorithms via Proof Complexity UPPER Bands

1. We will study **automatizability** — how hard it is to find proofs in a particular proof system.

   Nearly all proof systems we will study will be (somewhat) automatizable.

   So if a proof of small size/degree exists, there is an efficient algorithm to find one.

2. **Automatizability + small proofs of definability**

   \[ \Rightarrow \text{efficient algorithm} \]
Limitations of Algorithmic Methods via Proof Complexity Lower Bounds

1. DPLL/Res LBs rule out large family of SAT algs

2. SA lower bounds $\Rightarrow$ LBs for a large class of LPs $\Rightarrow$ LBs for large class of extended formulations

3. SOS lower bounds $\Rightarrow$ LBs for large class of SDPs + SDP extended formulations
Proof System Basics

Input: a set of constraints over $x_1 .. x_n$
[usually each $x_i \in \{0,1\}$, but we will also be interested in other cases: Finite field, $\mathbb{R}$]

A proof system $P(y) \rightarrow F$ polytime

$y$: proof in $P$  $F$: what $y$ proves

Soundness: $\text{Range} \subseteq \text{unsat}$
Completeness: $\text{UNSAT} \subseteq \text{Range}$

\[
P(y) \rightarrow (h \rightarrow F)
\]

$h$: hypotheses
$y$: $P$-proof from $h$

$y \rightarrow F$: what is being proven

Soundness: $\text{Range} \subseteq \text{taut}$, Completeness: $\text{tauts} \subseteq \text{Range}$

$P$ as a refutation system

as a derivation system
Proof Length, Automatizability

\[ S_p(F) = \text{size of shortest } P\text{-refutation of } F \]

\( P \) is polynomially automatizable if for all \( n \) sufficiently large and all \( F \) at most \( n \) variables

\[ A(F) \rightarrow y \] \( y \) is a \( P \)-proof of \( F \)

Runtime of \( A \) is \( \text{poly}(S_p(F)) \)

Can define \( g(n) \)-automatizability for other runtime functions \( g(n) \)

\( \Delta \text{Automatable means short proofs } \Rightarrow \text{efficient algorithms} \)
Proof Systems

SOS

SA  CPs  PC

Resolution
Resolution (as refutation system) over $\mathbb{D}$

$$f = (x_1 \lor x_2 \lor x_3)(x_1 \lor \overline{x}_3)(\overline{x}_1 \lor x_3)(\overline{x}_2)$$

Rule: $(x \lor c) (x \lor d) \Rightarrow c \lor d$

$$(x \lor x_2 \lor x_3) (x_1 \lor \overline{x}_3) (\overline{x}_1 \lor x_3) (\overline{x}_2)$$
Let's view Resolution a bit differently over \([0,1]\).

\[
f = (x_1 \lor x_2 \lor x_3) (x_1 \lor \overline{x_3}) (\overline{x_1} \lor x_2) (\overline{x_2})
\]

\[
x_1 + x_2 + x_3 \geq 1, \quad x_1 + (1-x_3) \geq 1, \quad (1-x_1) + x_2 \geq 1, \quad \overline{x_2} \geq 1, \quad 0 \leq x_i \leq 1
\]

A rule preserves all OR feasible solutions.
Let's view Resolution a bit differently

\[ f = (x_1 \lor x_2 \lor x_3)(x_1 \lor \overline{x}_3)(\overline{x}_1 \lor x_2)(\overline{x}_2) \]

\[ x_1 + x_2 + x_3 \geq 1, \quad x_1 + (1 - x_3) \geq 1, \quad (1 - x_1) + x_2 \geq 1, \quad (-x_2) \geq 1, \quad 0 \leq x_i \leq 1 \]

Original constraints: have a fractional solution
Extended constraints: no fractional solutions
• Resolution in the worst-case requires $2^{\Omega(w)}$ length refutations.

• Width-size relationship (for KCONs $F$)
  \[
  \text{width } w \text{ of proof } \Rightarrow \text{size } 2^{O(w)} \text{ Tree-Res} \\
  \text{width } w \text{ of proof } \Rightarrow \text{size } 2^{\sqrt{\text{nlgs}(F)}} \text{ Res}
  \]

• Automatizability:
  \[
  \text{width } w \text{ of proofs } \Rightarrow \text{can be found in } \mathcal{O}(w) \text{ time} \\
  \text{Tree Res is quasi-poly automatizable} \\
  \text{Res is automatizable in time } \exp(\sqrt{\text{nlgs}(F)})
  \]
Example where Resolution tightening helps

Paturi-Pudlák-Saks-Zane

$(1.364)^n$ algorithm for 3SAT ($\text{Hertli}-(\ln 308)^n$)

$(1.308)^n$ alg for unique 3SAT

Idea

Start with $f$.
Run bounded-width Resolution to "tighten" the constraints
Apply simple randomized search repeatedly

- Set unit clauses, or pick random var + assignment

Intuition: unique sat ass $\Rightarrow$ lots of unit clauses after tightening by bdded width Res $\Rightarrow$ more unit clauses
LP (as a proof system)

Sound, complete, poly automatizable proof system for linear inequalities over $\mathbb{R}^{\geq 0}$

LP: \[
\begin{align*}
\text{max} & \quad c^T x \\
\text{s.t.} & \quad A x \leq b \\
& \quad x \geq 0
\end{align*}
\]

Decision version: Is there a value $x$ satisfying (*)?
Farkas' Lemma (Completeness of LP, decision version)

A set \( \{Ax \leq b, x \geq 0\} \) of linear inequalities is unsat (over \( \mathbb{R} \)) iff \( \exists y \geq 0 \) s.t. \( y^T A = 0 \) and \( y^T b = -1 \).

Duality (Implicational Completeness of LP)

Consider any non-neg \( y \) s.t. \( y^T A \geq c^T \)

Then for any feasible solution \( x \) to \( \{Ax \leq b, x \geq 0\} \) we have \( c^T x \leq y^T Ax = y^T b \)

so \( y \geq 0 \) witnesses the upper bound \( b^T y \)

How tight is such an upper bound?
Duality

(P) primal:
\[
\begin{align*}
\text{max} & \quad c^T x \\
st. & \quad Ax \leq b, \quad x \geq 0
\end{align*}
\]

(D) dual:
\[
\begin{align*}
\text{min} & \quad b^T y \\
st. & \quad A^T y \geq c, \quad y \geq 0
\end{align*}
\]

**Duality Theorem** (implied by Farkas' Lemma)

Exactly one of the following holds

(i) Neither (P) nor (D) have a feas. soln
(ii) (P) has solns with arbitrarily large values, *(D) unsat
(iii) (D) has " " " small " " (P) unsat
(iv) Both (P) and (B) have optimal solns, \( x^* \) and \( y^* \)

Then \( c^T x^* = b^T y^* \)

\[ \uparrow \]

so there is a soln to dual that witnesses tight bound
So an LP "reputation" of \( \{Ax \leq b, x \geq 0\} \) is a non-neg linear comb. of these inequalities that equals -1.

An LP "derivation" of \( \{Ax \leq b, x \geq 0\} \rightarrow c^T x = c_0 \)

is nonneg \( \mathbf{y}^* \) s.t. \( (\mathbf{y}^*)^T b = c_0 \)

(since \( c^T x \leq (\mathbf{y}^*)^T Ax = (\mathbf{y}^*)^T b = c_0 \))
Polynomial automatizability of LP

Satisfiability of linear inequalities: in NP
Farkas' lemma / Duality Thm: in coNP

so in NP ∩ coNP

Ellipsoid Alg [Khachiyan ’79]

LP in P
As mentioned earlier, we can view LP as a relaxation of an LP. Then we have Fractional OPT = Integral OPT

Interested in Ratio
Let \( \{Ax \geq 0\} \) be a set of linear inequalities.

A refutation of \( \{Ax \geq 0\} \) (over \( x \in \{0,1\} \)) is a sequence of inequalities s.t. each is either from \( \{Ax \geq 0\} \) or follows from previous lines by a rule, and final line is \(-1 \geq 0\).

**Rules:**

① Can take positive linear comb's of previously derived ineq's

② Division with Rounding:

\[ \sum c_i x_i \geq b, \forall i \]

\[ \Rightarrow \sum \frac{c_i}{k} x_i = \left\lfloor \frac{b}{k} \right\rfloor \]
on Automatizability of CLs

Not known to be automatizable in any sense

(So we won't say much more about it)
Sewali Adams  (LP tightening)

add new variables to represent all low degree \((\leq d)\)

juntas  (Junta: \(x_1 x_2 x_3 \approx x_1 (1-x_2)x_3\))

This "lifts" LP from \(n\) dimensions to \(\leq n^d\) dimensions

projection back to \(x_1...x_n\) preserves all 0/1 solns

and will hopefully remain a lot of fractional solns
Original LP: \( \max c^T x \)
\[
\text{s.t. } A x \leq b \\
0 \leq x \leq 1
\]

Add new variables \( J_{s,t} \) \( \forall s,t \) \( s \cap t = \emptyset \)

\( J_{s,t} \) represents \( \prod_{i \in s} x_i \prod_{j \in t} (1-x_j) \)

New constraints:
\[
J_{s,t} \geq 0, \quad 1 - J_{s,t} \geq 0
\]
\[
J_{s,u,v,t} + J_{s,t} \geq J_{s,t}
\]
\[
J_{s,t} \geq J_{s,u,v,t} + J_{s,t}
\]
\[
J_{s,t} \left( \sum_{i,j \in s} a_{ij} x_j \right) \geq J_{s,t} b_j
\]
Sherali-Adams (static)

Let $\mathcal{G}$ be a set of poly equalities (includes $x_i^2 - x_i = 0$) and a set $\mathcal{H}$ of poly inequalities.

A $\mathcal{SA}$ derivation of $f$ from $(\mathcal{G}, \mathcal{H})$ is $(g_1, \ldots, g_m, P_1, \ldots, P_s)$ s.t.

\[
\sum_{i=1}^m g_i f_i + \sum_{e=1}^s p_e h_e = f
\]

arbitrary poly's

nonnegative linear combination of juntas

Degree $= \max \text{ degree } q \in \langle g_i f_i, p_e h_e \rangle$
SA automatitability

By Farkas' lemma, Duality degree $d$ SA refutations/dervations are automatitatable in time $n^{o(d)}$. 
Sum-of-squares SOS (Static)

Let $\mathcal{G}$ be a set of poly equalities (includes $x_i^2 - x_i = 0$) and a set of poly inequalities.

An SOS derivation of $f$ from $(\mathcal{G}, \mathcal{H})$ is $(g_1, \ldots, g_m, P_1, \ldots, P_s)$ s.t.

$$\sum_{i=1}^{m} g_i f_i + \sum_{a=1}^{s} P_a h_a = f$$

sum of squares

ie. $\sum_i (P_e;i)^2$

Degree = max degree of $g_i f_i, P_e h_e$
Sum-of-Squares SOS (static)
**Nullstellensatz (Static)**

Start with polynomials \( f_1 = 0, \ldots \) \( \) including \( x_i^2 - x_i = 0 \)

A Nullstellensatz derivation of \( f \) is \( (g, \ldots) \) s.t.

\[ \sum g_i f_i = f \]

\( f = -1 \) : refutation of \( \mathcal{F} = \{ f_i = 0, \ldots \} \)

---

**Poly Calculus (Dynamic)**

**Axioms**: \( f = 0 \) \( \forall f \in \mathcal{F} \) (including \( x_i^2 - x_i = 0 \))

**Rules**: \( f = 0 \Rightarrow x_j f = 0 \)

\( f = 0, g = 0 \Rightarrow ag + bf = 0 \)
Automatizability of Nullsatz + PC

Proofs of degree 1 can be found in time $n^{\text{ord}(d)}$.

Nullsatz: solve system of linear eqns

PC: bld-dgree version of gröbner basis alg