Lower Bounds for Cutting Planes Using Games

Yuval Filmus   Toniann Pitassi
University of Toronto

International Workshop on
Logic and Computational Complexity 2011

Executive summary

New perspective on two old results:

- BPR: *Lower bounds for cutting planes proofs with small coefficients* (Bonet, Pitassi, Raz, 1997).
- K: *Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic* (Krajíček, 1997).

Hope is to extend results to arbitrary coefficients.

Plan of talk

- Semantic Cutting Planes.

Plan of talk

- Semantic Cutting Planes.
- Communication protocols.

Plan of talk

- Semantic Cutting Planes.
- Communication protocols.
- The difficult proposition (BPR version).

Plan of talk

- Semantic Cutting Planes.
- Communication protocols.
- The difficult proposition (BPR version).
- Proof of the lower bound.

Plan of talk

- Semantic Cutting Planes.
- Communication protocols.
- The difficult proposition (BPR version).
- Proof of the lower bound.
- Extensions of the framework.

Semantic Cutting Planes

Refutation system with lines of the form

$$\sum_i a_i x_i \geq b$$

Variables x_i are implicitly assumed to be Boolean.
Derivation rule: $\ell_1, \ell_2 \vdash \ell$ if every $0/1$ assignment satisfying ℓ_1, ℓ_2 also satisfies ℓ.

Communication protocols

Two players cooperating to calculate $f(x, y)$.
Player 1 knows x.
Player 2 knows y.

Example: $f(x, y)$ is $\langle a, x\rangle + \langle b, y\rangle \geq c$.
Protocol $P_{\geq}$:

- Player 1 sends $s_1 \triangleq \langle a, x\rangle$.
- Player 2 sends $s_2 \triangleq \langle b, y\rangle$.
- Now both can compute $\langle a, x\rangle + \langle b, y\rangle$.

Transcript (communicated bits): $s_1 s_2$.

Communication protocols

Protocol dag is defined by:

- Set of states S (partial transcripts).
- Starting state $s_0 \in S$.
- Set of final states $F \subset S$.
- At non-final state s, player $P(s)$ sends a bit b.
- Protocol transitions to state $t(s, b)$.
- At final state s, protocol output is $\varphi(s)$.

Communication protocols

Protocol also includes:

- Strategy $\sigma_1(s, x)$ for Player 1.
- Strategy $\sigma_2(s, y)$ for Player 2.

Correctness:

If Player 1 uses σ_1 with her input x
and Player 2 uses σ_2 with his input y
then $\varphi(s_{\text{final}}) = f(x, y)$.

Communication protocols

Protocol also includes:

- Strategy $\sigma_1(s, x)$ for Player 1.
- Strategy $\sigma_2(s, y)$ for Player 2.

Correctness:

If Player 1 uses σ_1 with her input x
and Player 2 uses σ_2 with his input y
then $\varphi(s_{\text{final}}) = f(x, y)$.

Players don't have to use σ_1, σ_2!
When they do: *honest run* for x, y.

The difficult contradiction

Informally:

A graph on n vertices both has an m-clique and is $(m-1)$-colorable.

We take $m = \sqrt[3]{n}$.

The difficult contradiction

Formally:

- x_{vi}: vertex v is ith vertex of clique
- y_{vc}: vertex v gets color c
- $v \in [n]$, $i \in [m]$, $c \in [m-1]$

The difficult contradiction

Formally:

- x_{vi}: vertex v is ith vertex of clique
- y_{vc}: vertex v gets color c
- $v \in [n]$, $i \in [m]$, $c \in [m-1]$
- $\forall i\colon\ \sum_v x_{vi} \geq 1$
- $\forall v, i_1 \neq i_2\colon x_{vi_1} + x_{vi_2} \leq 1$
- $\forall v_1 \neq v_2, i\colon x_{v_1 i} + x_{v_2 i} \leq 1$

The difficult contradiction

Formally:

- x_{vi}: vertex v is ith vertex of clique
- y_{vc}: vertex v gets color c
- $v \in [n]$, $i \in [m]$, $c \in [m-1]$
- $\forall i: \sum_v x_{vi} \geq 1$
- $\forall v, i_1 \neq i_2: x_{vi_1} + x_{vi_2} \leq 1$
- $\forall v_1 \neq v_2, i: x_{v_1 i} + x_{v_2 i} \leq 1$
- $\forall v: \sum_c y_{vc} \geq 1$
- $\forall v, c_1 \neq c_2: y_{vc_1} + y_{vc_2} \leq 1$

The difficult contradiction

Formally:

- x_{vi}: vertex v is ith vertex of clique
- y_{vc}: vertex v gets color c
- $v \in [n]$, $i \in [m]$, $c \in [m-1]$
- $\forall i: \sum_v x_{vi} \geq 1$
- $\forall v, i_1 \neq i_2: x_{vi_1} + x_{vi_2} \leq 1$
- $\forall v_1 \neq v_2, i: x_{v_1 i} + x_{v_2 i} \leq 1$
- $\forall v: \sum_c y_{vc} \geq 1$
- $\forall v, c_1 \neq c_2: y_{vc_1} + y_{vc_2} \leq 1$
- $\forall v_1 \neq v_2, i_1 \neq i_2, c: x_{v_1 i_1} + x_{v_2 i_2} + y_{v_1 c} + y_{v_2 c} \leq 3$

Plan of proof

Basic idea:

Transform a refutation to a monotone circuit
of comparable size.
Use a monotone circuit lower bound.

Plan of proof

Basic idea:

Transform a refutation to a monotone circuit
of comparable size.
Use a monotone circuit lower bound.

Monotone circuit takes an input graph G, given as edge variables $G(v_1, v_2)$.

- Returns 1 if G has an m-clique.
- Returns 0 if G is $(m-1)$-colorable.

Plan of proof

Basic idea:

Transform a refutation to a monotone circuit of comparable size.
Use a monotone circuit lower bound.

Monotone circuit takes an input graph G, given as edge variables $G(v_1, v_2)$.

- Returns 1 if G has an m-clique.
- Returns 0 if G is $(m-1)$-colorable.

Lower bound (Alon/Boppana): $2^{\Omega(\sqrt[3]{n})}$.

Plan of reduction

- Two players (clique player and coclique player) play a game on the proof dag.
- Game starts at the final line, proceeds toward the axioms.
- Game ends at an axiom

$$x_{v_1 i_1} + x_{v_2 i_2} + y_{v_1 c} + y_{v_2 c} \leq 3.$$

- If $G(v_1, v_2) = 1$, clique player wins.
- If $G(v_1, v_2) = 0$, coclique player wins.

Rules of the game

Suppose game is at a line ℓ deduced from ℓ_1, ℓ_2.

- Players use protocol $P_{\geq}$ to determine which of ℓ_1, ℓ_2 are falsified.
  - Clique player is Player 1.
  - Coclique player is Player 2.
- Record transcripts $\tau(\ell_1), \tau(\ell_2)$.
- Local consistency: $\tau(\ell), \tau(\ell_1), \tau(\ell_2)$ must correspond to some legal honest run *jointly*.
  - Enforced by limiting what bits players can send.
- If ℓ_1 is falsified, proceed to ℓ_1, otherwise proceed to ℓ_2.

Winning strategy for the clique player

If G has an m-clique:

- Fix an encoding $\tilde{x}$ of an m-clique.
- Clique player plays honestly using $\tilde{x}$: at state s, she outputs $\sigma_1(s, \tilde{x})$.
- Local consistency implies: each visited line is falsfied by $\tilde{x}$ and *some* y.
- Game ends at an axiom

$$x_{v_1 i_1} + x_{v_2 i_2} + y_{v_1 c} + y_{v_2 c} \leq 3$$

- Must have $\tilde{x}_{v_1 i_1} = \tilde{x}_{v_2 i_2} = 1$.
- Since $\tilde{x}$ encodes a clique, $G(v_1, v_2) = 1$.

From game to circuit

Convert the game to a monotone circuit:

- Construct the state dag of the game.
- Each time it is the clique player's turn to speak, put an $\vee$ gate.
- Each time it is the coclique player's turn to speak, put an $\wedge$ gate.
- Replace a (v_1, v_2) leaf with $G(v_1, v_2)$.

From game to circuit

Convert the game to a monotone circuit:

- Construct the state dag of the game.
- Each time it is the clique player's turn to speak, put an $\vee$ gate.
- Each time it is the coclique player's turn to speak, put an $\wedge$ gate.
- Replace a (v_1, v_2) leaf with $G(v_1, v_2)$.
- Clique player has a winning strategy: circuit outputs 1.
- Coclique player has a winning strategy: circuit outputs 0.

Size of circuit

Game states: $\langle \ell, \tau(\ell), \tau(\ell_1), \tau(\ell_2) \rangle$

- Current node ℓ
- Transcript $\tau(\ell)$ from previous step
- Partial transcripts $\tau(\ell_1), \tau(\ell_2)$

Size of circuit

Game states: $\langle \ell, \tau(\ell), \tau(\ell_1), \tau(\ell_2) \rangle$

- Current node ℓ
- Transcript $\tau(\ell)$ from previous step
- Partial transcripts $\tau(\ell_1), \tau(\ell_2)$

Size of circuit: $L2^{3C}$

- L: number of lines in proof
- C: communication complexity of $P_{\geq}$ (number of communicated bits)

Wrapping up

Protocol $P_{\geq}$ involves sending $\langle a, x \rangle, \langle b, y \rangle$.

If coefficients a_i, b_i are of size 2^C,
communication complexity is roughly $O(C)$.

So $L = \Omega\left(2^{\sqrt[3]{n} - O(C)}\right)$.

Only interesting if $C = o(\sqrt[3]{n})$.

Extensions

Can add random public coin tosses to the game:

- Convert game to a monotone *real* circuit.
- Replace $\vee$ gates by max gates.
- Replace $\wedge$ gates by min gates.
- Coin tosses correspond to *average* gates.
- Output is probability that clique player wins.

Pudlák extended the lower bound to this case.

Open questions

Pudlák (1997) proved lower bound for *syntactic* Cutting Planes with arbitrary coefficients, using monotone real circuits.

Can BPR/K be extended to arbitrary coefficients?

- Use a randomized “greater than” protocol.
- Allow circuit to err on some inputs.

Open questions

Pudlák (1997) proved lower bound for *syntactic* Cutting Planes with arbitrary coefficients, using monotone real circuits.

Can BPR/K be extended to arbitrary coefficients?

- Use a randomized "greater than" protocol.
- Allow circuit to err on some inputs.

Is semantic Cutting Planes stronger than syntactic Cutting Planes?