Relation $S \subseteq \mathcal{Z} \times Q$ can be viewed as SEARCH PROBLEM with

- input $z \in \mathcal{Z}$
- Output $q \in Q$
- Task: Given z, find q such that $(z, q) \in S$

Assume S is TOTAL, so that

$\forall z \, \exists q \; (z, q) \in S$

We will always have $\mathcal{Z} = \{0,1\}^m$, so fix this from now on

For $\ell \in \mathbb{N}^+$, the LIFT OF LENGTH ℓ of S is the search problem

$$Lift_\ell(S) \subseteq [\ell]^m \times \{0,1\}^{m \cdot \ell} \times Q$$

with input domain $[\ell]^m \times \{0,1\}^{m \cdot \ell}$

output range Q

such that for $\vec{x} \in [\ell]^m$

$\vec{y} = \{y_{i,j}\}_{i \in [m],\, j \in [\ell]}$

it holds that

$$(\vec{x}, \vec{y}, q) \in Lift_\ell(S)$$

iff

$$((y_{1,x_1}, y_{2,x_2}, \ldots, y_{m,x_m}), q) \in S$$

$\vec{x}$: SELECTOR VARIABLES

$\vec{y}$: MAIN VARIABLES

$\mathrm{Ind}(x_i, y_i) = y_{i, x_i}$

$\mathrm{Ind}(\vec{x}, \vec{y}) = \vec{y}_{\vec{x}} = (y_{1,x_1}, \ldots, y_{m,x_m})$

LIFTED CNF FORMULA

CNF formula F over $u_1, \ldots, u_n$, $\ell \in \mathbb{N}^+$

The LIFT OF LENGTH ℓ of F, denoted $\mathrm{Lift}_\ell(F)$ is CNF formula with

SELECTOR VARIABLES $\{x_{i,j}\}_{i \in [n], j \in [\ell]}$

MAIN VARIABLES $\{y_{i,j}\}_{i \in [n], j \in [\ell]}$

containing following clauses

(1) For every $i \in [n]$ an AUXILIARY CLAUSE

$$x_{i,1} \vee x_{i,2} \vee \cdots \vee x_{i,\ell}$$

(2) For every $C \in F$, $C = u_{i_1} \vee \cdots \vee u_{i_s} \vee \bar{u}_{i_{s+1}} \vee \cdots \vee \bar{u}_{i_t}$ and every tuple $(j_1, \ldots, j_t) \in [\ell]^t$ a MAIN CLAUSE

$$\bar{x}_{i_1,j_1} \vee y_{i_1,j_1} \vee \cdots \vee \bar{x}_{i_s,j_s} \vee y_{i_s,j_s} \vee \bar{x}_{i_{s+1},j_{s+1}} \vee \bar{y}_{i_{s+1},j_{s+1}} \vee \cdots \vee \bar{x}_{i_t,j_t} \vee \bar{y}_{i_t,j_t}$$

F is unsatisfiable iff $H = \mathrm{Lift}_\ell(F)$ is unsatisfiable

If F is a k-CNF formula with m clauses then H is $\max(2k, \ell)$-CNF formula with at most $m \cdot \ell^k + n$ clauses

We will have $\ell \gg k$, but can convert wide clauses to constant width — ignore this detail

Our proof cplx trade-offs will be for CNF formula $Lift_\ell(F)$ and will follow from lower bounds on search problem $Search(Lift_\ell(F))$

But our communication complexity lower bounds will be for lifted search problems

$Lift_\ell(Search(F))$

Slight type mismatch... But this is not a problem — $Lift_\ell(Search(F))$ is just the "restricted version" where selector variables are always assigned as expected, so that auxiliary clauses not falsified

OBSERVATION 1

If F is an unsatisfiable CNF formula, then any two-player protocol for $Search(Lift_\ell(F))$ where all selector variables $x_{i,j}$ in the same block are given to the same player can be adapted to a protocol for $Lift_\ell(Search(F))$ with same parameters

Proof sketch For every $x_i = a \in [\ell]$, Alice just translates to $x_{i,a} = 1$, $x_{i,j} = 0$ for $j \neq a$. Then run protocol for CNF formula.

LEMMA 2

If a CNF formula H can be refuted in cutting planes in length L and line space s, then for any partition of the variables between Alice and Bob there is a real communication protocol solving $Search(H)$ in $\lceil \log L \rceil$ rounds with total communication cost at most $s \cdot \lceil \log L \rceil$

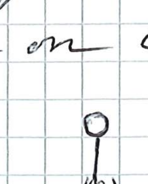
Proof sketch Binary search over configurations in refutation

THEOREM 3 (SIMULATION THEOREM)

Let S be a relation with input domain $\{0,1\}^m$ and let $\ell = m^{3+\varepsilon}$ for $\varepsilon > 0$. If there is an r-round real communication protocol in cost c that solves $Lift_\ell(S)$, then there is a parallel decision tree in depth r solving S using a total of $O(c / \log \ell)$ queries.

LEMMA 4

Parallel decision tree for $Search(Peb_G)$ in depth r with c queries $\Rightarrow$ Pebbler has winning strategy in r-round Dymond – Tompa game on G in cost at most $c+1$

Proof sketch Add extra sink to get $\hat{G}$ = 

For $\hat{G}$ the correspondence is exact.
Clearly changes cost by at most 1

LEMMA 5

For any $n, r \in \mathbb{N}^+$ exists explicitly constructible DAG $G(n,r)$ with $O(rn \log n)$ vertices such that the r-round Dymond–Tompa game on $G(n,r)$ is at least $\Omega(n)$.

Proof sketch Stack butterfly graphs on top of one another.

Combining all of these results allows to establish cutting planes ~~size~~-space trade-offs as follows

THEOREM 6 (MAIN THEOREM)

Let G DAG over m vertices such that winning strategy for Pebbler in the r-round Dymond-Tompa game on G has cost $\Omega(c)$, and let $\ell = m^{3+\varepsilon}$ for constant $\varepsilon > 0$. Then $Lift_\ell(Peb_G)$ is a 6-CNF formula with $N = \Theta(m^{10+3\varepsilon})$ clauses over $\Theta(m^{4+\varepsilon})$ variables such that any cutting planes refutation of $Lift_\ell(Peb_G)$ in line space $< \frac{c}{r} \log N$, even with coefficients of unbounded size, requires length $\exp(\Omega(r))$

Proof

Suppose towards contradiction that
$\exists\, \pi : \mathrm{Lift}_\ell(\mathrm{Peb}_G) \vdash \bot$ in length $L(\pi) =$
$= \exp(o(r))$ and line space $< \frac{c}{r} \log N$

Lemma 2 $\Rightarrow$ $\exists$ real communication protocol for $\mathrm{Lift}_\ell(\mathrm{Search}(\mathrm{Peb}_G))$ in $o(r)$ rounds and total cost $o(c \log N)$

Theorem 3 $\Rightarrow$ $\exists$ parallel decision tree solving $\mathrm{Search}(\mathrm{Peb}_G)$ using $o(c)$ queries and depth $o(r)$

Lemma 4 $\Rightarrow$ Pebbler wins $o(r)$-round DT game on G in cost $o(c)$. Contradiction $\lightning$ $\square$

We also need to find short and small-space refutations of $\mathrm{Lift}_\ell(\mathrm{Peb}_G)$

<u>Short refutation</u> Use black pebbling upper bounds as in [BN11, HN12] but be more careful in pebbling step since lift length is large.

<u>Space efficient refutation</u> Use that a shallow (standard) decision tree yields space-efficient tree like resolution refutations.

We will probably not have time to go into details about the upper bounds, but they are not hard.

SIMULATING COMMUNICATION PROTOCOLS BY DECISION TREES

Clearly, if Alice and Bob know a good decision tree T for search problem S, they can build protocol Π for $Lift_G(S)$ by simulating T

When T queries z_i, Alice sends $\vec{x}_i$ to Bob and Bob answers with $y_{i,\vec{x}_i} = Ind(\vec{x}_i, \vec{y}_i) = z_i$
Then they move to next node in T according to z_i

What the simulation theorem says is this is best possible. From any protocol Π, can extract decision tree $T(\Pi)$ "that is being simulated" with same parameters

Two views of decision trees:

(1) A directed tree
- non-leaf nodes labelled by queried variables
- edges labelled by (Boolean) values
- leaf nodes labelled by answers

(2) Any algorithm that
- collects potentially useful info during offline phase
- during online phase
  - (a) issues queries
  - (b) makes arbitrary computations, but only based on offline info & query results
- announces answers

Query complexity and depth are same notion for both perspectives

Given protocol Π, we design algorithm $eval_{\Pi}(\vec{z})$ as in (2) that will walk through protocol tree Π

For node $v \in \Pi$ let

$\mathcal{X}^v$ = Alice's inputs consistent with v

$\mathcal{Y}^v$ = Bob's — " —

Maintain nicely structured but large subsets

$A \subseteq \mathcal{X}^v$

$B \subseteq \mathcal{Y}^v$

Worry about leakage of correlated info from protocol

Bob cannot really leak useful info in $\vec{y}_i$, since # bits ℓ for single variable z_i >> total # variables

But Alice can identify coordinate x_i with few bits. Want to make sure not to leak such info

For not yet queried coordinates I

Make sure for $i \in I$ that even if Alice fixes x_j for all $j \in I \setminus \{i\}$ to values, there are still many choices for x_i

Simulation in $eval_\Pi(\vec{z})$ at node v in Π proceeds as follows

COMMUNICATION EVENTS

Send "safe messages" $\vec{a_v}$ & $\vec{b_v}$ that shrink A & B as little as possible

If moving to node $w \in \Pi$, set

$A := A \cap X^w$

$B := B \cap Y^w$

then clean up A & B to restore nice structure

QUERY EVENTS

If some coordinate i is too constrained (for Alice, we don't worry about Bob) then

- issue query to z_i
- fix x_i & $\vec{y_i}$ so that $Ind(x_i, \vec{y_i}) = y_{i,x_i} = z_i$ in A & B
- remove i from I

Once we reach a leaf v in Π, all $x \in X^v$ and $y \in Y^v$ are consistent with answer given, and we can query all remaining z_i and fix x_i & y_i so that $Ind(x, y) = z$, meaning that decision tree also gives correct answer

To control # decision tree queries maintain density function for Alice's coordinates in I. Density:

- increases at communication events
- decreases at query events
- but is always non-negative

so query cplx $\lessapprox$ communication

So far, this is generic description of simulation theorems in communication cplx.

Extra complication for us: Normally queries are issued and $\mathrm{Ind}(x_i, y_i) = z_i$ fixed as soon as coordinate becomes constrained. We have to wait till end of round!

- Fix Alices coordinate x_i right away
- But wait with querying z_i till end of round

Still have to make sure that everything is consistent.

To make things manageable, we will only prove the simulation theorem for <u>deterministic communication</u>, which gives our main theorem for cutting planes with coefficients of at most polynomial magnitude (= logarithmic # bits) often referred to as CP^*.

THEOREM 3'

If $\exists$ deterministic protocol solving $Lift_\ell(S)$ for $\ell = m^{3+\varepsilon}$ using communication c and r rounds, then $\exists$ PDT solving S using $O(c/\log \ell)$ queries and depth r.

We need some magic constants

$$\gamma, \delta, \lambda, \mu \in (0,1)$$

$$\gamma = \frac{1}{3+\varepsilon} \qquad (a)$$

$$\lambda - \gamma > \mu \qquad (b)$$

$$\mu + \delta - 1 > \gamma \qquad (c)$$

$$\gamma + \delta < 1 \qquad (d)$$

Can set $\xi = (\gamma - 1/3)/2$ and

$$\lambda = 1 - \xi$$

$$\mu = \delta = 2/3$$

λ used to measure average constrainedness

μ measures minimum required non-constrainedness

Maintain $A \subseteq X^v \subseteq [\ell]^m$

$B \subseteq Y^v \subseteq \{0,1\}^{\ell m}$

Coordinates not yet queried by decision tree are $I \subseteq [m]$

For sets & vectors use subscripts to indicate what index sets they range over, so

$$A_I \subseteq [\ell]^I = \{(x_i) \mid i \in I, x_i \in [\ell]\}$$

For $I \cap J = \emptyset$ let

$x_I \cdot x_J$ denote concatenation (respecting order of indices in $I \cup J$)

Projection of x_J to $I \subseteq J$ is denoted

$\pi_I(x_J)$ = vector x_I s.t. $\exists\, x_{J \setminus I}$ for which $x_J = x_I \cdot x_{J\setminus I}$

Generalize to set S_y

$$\Pi_I(S_y) = \{\pi_I(x_j) : x_j \in S_y\}$$

For $A_I \subseteq [\ell]^I$ and $i \in I$, define bipartite graph $\mathrm{Graph}_i(A_I)$ to measure how constrained coordinate i is for Alice

- Left vertices $x_{\{i\}} \in [\ell]$
- Right vertices $x_{I \setminus \{i\}} \in [\ell]^{I \setminus \{i\}}$
- Edge $(x_{\{i\}}, x_{I\setminus\{i\}})$ if

$$x_{I\setminus\{i\}} \circ x_{\{i\}} \in A_I$$

Look at right degrees of this graph, ignoring isolated vertices (which have already been ruled out for Alice)

$\mathrm{MinDeg}_i(A_I)$ = minimum (non-zero) right degree

$\mathrm{AvgDeg}_i(A_I)$ = average (non-zero) right degree

A fixed coordinate i is considered unconstrained if $\mathrm{AvgDeg}_i(A_I) \geq \ell^{\lambda}$ (note $\lambda \approx 1$)

A_I is THICK if for all $i \in I$

$\mathrm{MinDeg}_i(A_I) \geq \ell^{\mu}$ (note $\mu = 2/3$)

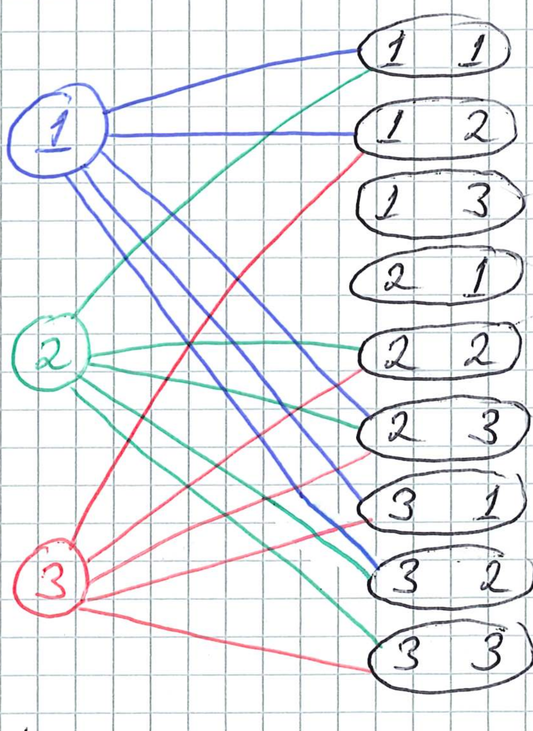
We want to maintain property that A_I is always thick.

SIM $\underline{\overline{VII}}$

$\underline{I} = \{1, 2, 3\}$ Lift length $\ell = 3$

$A_{\underline{I}} = \{(1,1,1), (2,1,1), (3,1,2),$
$(1,1,2), (2,2,2), (3,2,2),$
$(1,2,3), (2,2,3), (3,2,3),$
$(1,3,1), (2,3,2), (3,3,1),$
$(1,3,2), (2,3,3), (3,3,3), \}$

$Graph_1(A_{\underline{I}})$



$MinDeg_1(A_{\underline{I}}) = 2$

Note

- \# edges in $\text{Graph}_i(A_I) = |A_I|$
- \# $\boxed{\text{vertices}}$ $\boxed{\text{right}}$ with positive degree

$$= |\pi_{I\setminus\{i\}}(A_I)|$$

Hence

$$\text{AvgDeg}_i(A_I) = \frac{|A_I|}{|\pi_{I\setminus\{i\}}(A_I)|} \qquad (\dagger)$$

OBSERVATION 7 Minimum degree (and hence thickness) can only increase under projections. I.e., for $j \in I\setminus\{i\}$ it holds that

$$\text{MinDeg}_j(\pi_{I\setminus\{i\}}(A)) \geq \text{MinDeg}_j(\pi_I(A))$$

<u>Proof</u> If $\exists$ edge $(x_{\{j\}}, x_{I\setminus\{j\}})$, so that

$x_{I\setminus\{j\}} \circ x_{\{j\}} \in A_I = \pi_I(A)$, then

$x_{I\setminus\{i,j\}} \circ x_{\{j\}} \in \pi_{I\setminus\{i\}}(A)$, so

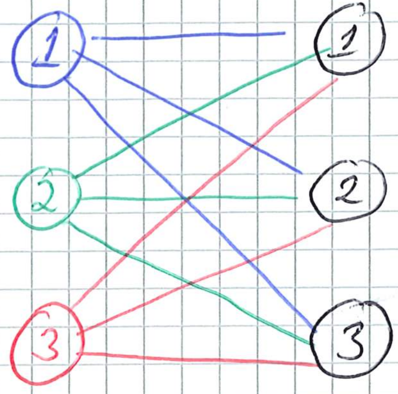
$(x_{\{j\}}, x_{I\setminus\{i,j\}})$ is also an edge

No neighbour on the left are lost. $\boxed{\equiv}$

See example on next page that min degree can go up.

$$A_{I\setminus\{2\}} = \{ (1,1), (2,1), (3,1), (1,2), (2,2), (3,2), (1,3), (2,3), (3,3) \}$$

$Graph_1(A_{I\setminus\{2\}})$



DENSITY LOSS MEASURES for non-queried coordinates I

(Logarithms are base 2)

Alice: $\alpha(A_I) = -\log\left(\frac{|A_I|}{\ell^{|I|}}\right) =$

$= |I| \log \ell - \log |A_I|$

Bob: $\beta(B_I) = -\log\left(\frac{|B_I|}{2^{\ell |I|}}\right)$

$= |I| \cdot \ell - \log |B_I|$

At start of protocol & simulation α & β are both 0.
Need to keep them small throughout simulation

When we project
- total # choices decreases
- but density increases
(and so loss of density decreases)

OBSERVATION 8

$$\alpha\left(\pi_{I\setminus\{i\}}(A)\right) = \alpha(\pi_I(A)) - \log \ell + \log \mathrm{AvgDeg}_i(\pi_I(A)).$$

Proof By definition

$$\alpha(\pi_I(A)) = |I| \log \ell - \log|\pi_I(A)|$$

$$\alpha(\pi_{I\setminus\{i\}}(A)) = (|I|-1)\log \ell - \log|\pi_{I\setminus\{i\}}(A)|$$

$$\alpha\left(\pi_{I\setminus\{i\}}(A)\right) - \alpha(\pi_I(A)) =$$

$$= -\log \ell - \log(|\pi_{I\setminus\{i\}}(A)|) + \log(|\pi_I(A)|)$$

$$= -\log \ell + \log\Bigg(\underbrace{\frac{|\pi_I(A)|}{|\pi_{I\setminus\{i\}}(A)|}}_{=\ \mathrm{AvgDeg}_i(\pi_I(A))}\Bigg)$$

by (†)