Extended Formulations for Combinatorial Optimization

Samuel Fiorini

Université libre de Bruxelles (ULB, Brussels)

ISCO 2016 Summer School
Day 2 (afternoon)

Warm-up

Extended Formulation of the Stable Set Polytope of a Comparability Graph

$P = (V, \preccurlyeq)$ partially ordered set (poset)

$G = G(P)$ comparability graph of P (always perfect)

Warm-up

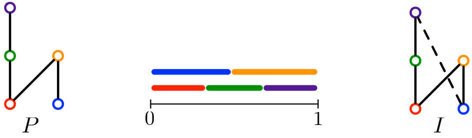
Extended Formulation of the Stable Set Polytope of a Comparability Graph

$P = (V, \preccurlyeq)$ partially ordered set (poset)

$G = G(P)$ comparability graph of P (always perfect)

$\{(y_{v^-}, y_{v^+})\}_{v \in V}$ is *consistent* with P if

- $\forall v \in V$: (y_{v^-}, y_{v^+}) open interval $\subseteq (0,1)$
- $v \preccurlyeq w \implies y_{v^+} \leqslant y_{w^-}$



Warm-up

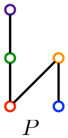
Extended Formulation of the Stable Set Polytope of a Comparability Graph

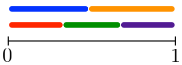
$P = (V, \preccurlyeq)$ partially ordered set (poset)

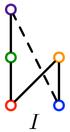
$G = G(P)$ comparability graph of P (always perfect)

$\{(y_{v^-}, y_{v^+})\}_{v \in V}$ is *consistent* with P if

- $\forall v \in V$: (y_{v^-}, y_{v^+}) open interval $\subseteq (0, 1)$
- $v \preccurlyeq w \implies y_{v^+} \leqslant y_{w^-}$







$$\text{STAB}(G(P)) = \Big\{ x \in \mathbb{R}^V \;|\exists \{(y_{v^-}, y_{v^+})\}_{v \in V} \text{ consistent with } P \\ \text{s.t. } x_v = y_{v^+} - y_{v^-} \quad \forall v \in V \Big\}$$

$f : A \times B \to \{0, 1\}$ Boolean function ($\equiv$ binary matrix)

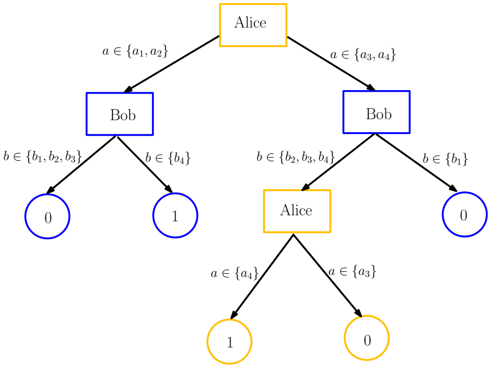
Two players:

- Alice knows $a \in A$
- Bob knows $b \in B$

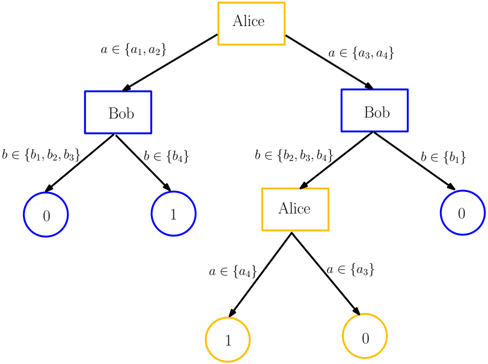
want to **compute** $f(a, b)$ **by exchanging bits**

Goal: Minimize complexity := #bits exchanged

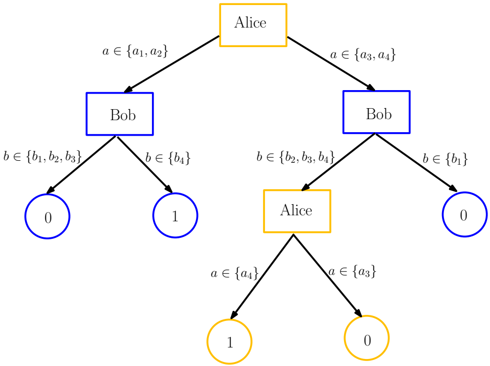
	b_1	b_2	b_3	b_4
a_1	0	0	0	1
a_2	0	0	0	1
a_3	0	0	0	0
a_4	0	1	1	1



	b_1	b_2	b_3	b_4
a_1	0	0	0	1
a_2	0	0	0	1
a_3	0	0	0	0
a_4	0	1	1	1



	b_1	b_2	b_3	b_4
a_1	0	0	0	1
a_2	0	0	0	1
a_3	0	0	0	0
a_4	0	1	1	1



Observation

$\exists$ complexity c protocol for computing $f \implies \mathrm{rk}_+(f) \leqslant 2^c$

G graph with n vertices

$A = \{a \in \{0,1\}^n \mid a \text{ encodes a clique in } G\}$

$B = \{b \in \{0,1\}^n \mid b \text{ encodes a stable set in } G\}$

$$f(a,b) = \begin{cases} 1 & \text{if } a, b \text{ are disjoint} \\ 0 & \text{if } a, b \text{ intersect} \end{cases}$$

G graph with n vertices

$A = \{a \in \{0,1\}^n \mid a \text{ encodes a clique in } G\}$

$B = \{b \in \{0,1\}^n \mid b \text{ encodes a stable set in } G\}$

$$f(a,b) = \begin{cases} 1 & \text{if } a, b \text{ are disjoint} \\ 0 & \text{if } a, b \text{ intersect} \end{cases}$$

Theorem (Yannakakis '91)

$\exists\ O(\log^2 n)$-*complexity protocol for* $f = f(G)$

G graph with n vertices

$A = \{a \in \{0,1\}^n \mid a \text{ encodes a clique in } G\}$

$B = \{b \in \{0,1\}^n \mid b \text{ encodes a stable set in } G\}$

$$f(a,b) = \begin{cases} 1 & \text{if } a, b \text{ are disjoint} \\ 0 & \text{if } a, b \text{ intersect} \end{cases}$$

Theorem (Yannakakis '91)

$\exists$ $O(\log^2 n)$-*complexity protocol for* $f = f(G)$

Corollary (Yannakakis '91)

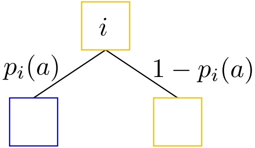
$\forall$ *perfect graphs* G: $\quad xc(STAB(G)) = 2^{O(\log^2 n)} = n^{O(\log n)}$

Extended Formulations as Communication Protocols

Randomized Protocols Computing a Function in Expectation

The main differences:

- Alice and Bob can use (private) random bits to make choices



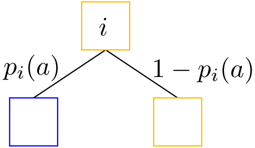
- $f : A \times B \to \mathbb{R}_+$, Alice and Bob can output any value $\in \mathbb{R}_+$

Extended Formulations as Communication Protocols

Randomized Protocols Computing a Function in Expectation

The main differences:

- Alice and Bob can use (private) random bits to make choices



- $f : A \times B \to \mathbb{R}_+$, Alice and Bob can output any value $\in \mathbb{R}_+$

Theorem (Faenza, F, Grappe & Tiwary '11)

If $c = c(f)$ is the minimum complexity of a randomized communication protocol with nonnegative outputs computing f in expectation, then

$$rk_+(f) = \Theta(2^c)$$

Write $M = TU$, where

$$\left|\begin{array}{l} T \in \mathbb{R}_+^{m\times r} \text{ row-stochastic (w.l.o.g.)} \\ U \in \mathbb{R}_+^{r\times n} \\ r \leqslant \mathsf{rk}_+(M) + 1 \end{array}\right.$$

Write $M = TU$, where $\left| \begin{array}{l} T \in \mathbb{R}_+^{m \times r} \text{ row-stochastic (w.l.o.g.)} \\ U \in \mathbb{R}_+^{r \times n} \\ r \leqslant \mathsf{rk}_+(M) + 1 \end{array} \right.$

Protocol:

- Alice gets row index i, Bob gets column index j
- Alice picks random column index $k \in [r]$ w.p. T_{ik}, sends it to Bob
- Bob outputs value U_{kj}

Write $M = TU$, where $\left| \begin{array}{l} T \in \mathbb{R}_+^{m \times r} \text{ row-stochastic (w.l.o.g.)} \\ U \in \mathbb{R}_+^{r \times n} \\ r \leqslant \mathsf{rk}_+(M) + 1 \end{array} \right.$

Protocol:

- Alice gets row index i, Bob gets column index j
- Alice picks random column index $k \in [r]$ w.p. T_{ik}, sends it to Bob
- Bob outputs value U_{kj}

Expected value on input (i, j): $\displaystyle\sum_{k=1}^{r} T_{ik} U_{kj} = M_{ij}$

Complexity: $\log \mathsf{rk}_+(M) + O(1)$

Three equivalent ways to look at EFs:

1. A linear system $Ex + Fy = g$, $y \geqslant 0$ with $y \in \mathbb{R}^r$

Three equivalent ways to look at EFs:

1. A linear system $Ex + Fy = g$, $y \geqslant 0$ with $y \in \mathbb{R}^r$

2. A rank-r nonnegative factorization $S = TU$ of slack matrix S

Three equivalent ways to look at EFs:

1. A linear system $Ex + Fy = g$, $y \geqslant 0$ with $y \in \mathbb{R}^r$
2. A rank-r nonnegative factorization $S = TU$ of slack matrix S
3. A $\log r$-complexity randomized protocol computing S in expectation

Monotone Boolean Functions, Monotone Circuits

Definition (monotone Boolean function)

A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is said to be *monotone* if

$$\forall x, y \in \{0,1\}^n \quad : \quad x \leqslant y \implies f(x) \leqslant f(y)$$

Monotone Boolean Functions, Monotone Circuits

Definition (monotone Boolean function)

A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is said to be *monotone* if

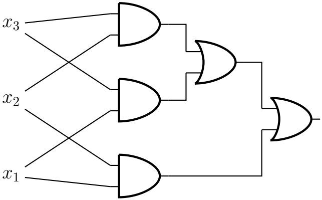
$$\forall x, y \in \{0,1\}^n \quad : \quad x \leqslant y \implies f(x) \leqslant f(y)$$

Examples:

- *Weighted threshold function*. Given $s_1, \ldots, s_n, D \in \mathbb{R}_+$:

  $$f(x) = 1 \iff \sum_{i=1}^{n} s_i x_i \geqslant D$$

- Any function f computed by a **monotone circuit** with n inputs



Definition (monotone Karchmer-Wigderson game)

- Alice is given $a \in \{0,1\}^n$ such that $f(a) = 0$
- Bob is given $b \in \{0,1\}^n$ such that $f(b) = 1$

Goal: compute together an index i^* such that $a_{i^*} = 0$ and $b_{i^*} = 1$

Definition (monotone Karchmer-Wigderson game)

- Alice is given $a \in \{0,1\}^n$ such that $f(a) = 0$
- Bob is given $b \in \{0,1\}^n$ such that $f(b) = 1$

Goal: compute together an index i^* such that $a_{i^*} = 0$ and $b_{i^*} = 1$

Theorem (Karchmer and Wigderson '90)

$\min$ *(# bits for solving the monotone KW game)* $= \min$ *(depth of a monotone circuit computing f)*

Definition (monotone Karchmer-Wigderson game)

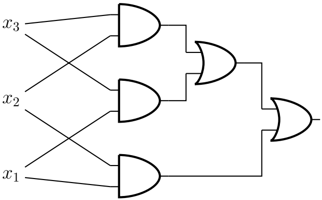
- Alice is given $a \in \{0,1\}^n$ such that $f(a) = 0$
- Bob is given $b \in \{0,1\}^n$ such that $f(b) = 1$

Goal: compute together an index i^* such that $a_{i^*} = 0$ and $b_{i^*} = 1$

Theorem (Karchmer and Wigderson '90)

$\min$ *(# bits for solving the monotone KW game)* $= \min$ *(depth of a monotone circuit computing f)*

Proof. $(\leqslant)$



□

- $P = P(f) := \mathsf{conv}(f^{-1}(1))$
- $Q = Q(f) := \{x \mid \forall a \in f^{-1}(0) : \sum_{i:a_i=0} x_i \geqslant 1\}$

- $P = P(f) := \mathrm{conv}(f^{-1}(1))$
- $Q = Q(f) := \{x \mid \forall a \in f^{-1}(0) : \sum_{i:a_i=0} x_i \geqslant 1\}$

Observation (Hrubeš, rediscovered by Göös '16)

$\exists$ depth-D monotone circuit computing $f \implies \exists$ complexity-D protocol for the corresponding monotone KW game $\implies \exists$ size-$n2^D$ EF of the pair (P, Q)

- $P = P(f) := \mathsf{conv}(f^{-1}(1))$
- $Q = Q(f) := \{x \mid \forall a \in f^{-1}(0) : \sum_{i:a_i=0} x_i \geqslant 1\}$

Observation (Hrubeš, rediscovered by Göös '16)

$\exists$ depth-D monotone circuit computing f $\implies$ $\exists$ complexity-D protocol for the corresponding monotone KW game $\implies$ $\exists$ size-$n2^D$ EF of the pair (P, Q)

Proof.

$$S_{ab} = \left(\sum_{i:a_i=0} b_i\right) - 1 = \sum_{i:a_i=0,\ i \neq i^*} b_i = (a_1, \ldots, a_{i^*-1}, a_{i^*+1}, \ldots, a_n) \cdot \begin{pmatrix} b_1 \\ \vdots \\ b_{i^*-1} \\ b_{i^*+1} \\ \vdots \\ b_n \end{pmatrix}$$

$\square$

Extended Formulations as Communication Protocols

Consequences of the KW/EF connection

Theorem (Raz-Wigderson '90)

Every monotone circuit that decides if a given n-vertex graph has a perfect matching, has depth $\Omega(n)$.

follows in a *black-box* way from

Theorem (Rothvoss'14)

The matching polytope $P_{\mathrm{match}}(K_n)$ has extension complexity $2^{\Omega(n)}$.

Theorem (matroids from the bases)

A collection $\mathcal{B}$ of subsets of a finite set E form the bases of a matroid if and only if

1. *$\mathcal{B}$ is nonempty*
2. *$\forall A, B \in \mathcal{B}$ and $a \in A \smallsetminus B$, $\exists b \in B \smallsetminus A$ s.t. $A \smallsetminus \{a\} \cup \{b\} \in \mathcal{B}$*

Definition (sparse paving matroid)

Let $\mathcal{N}$ be a collection of subsets of E s.t.

1. all the sets in $\mathcal{N}$ have the same size r
2. no two sets in $\mathcal{N}$ have Hamming distance $\leqslant 2$

Then $\mathcal{B} := \binom{E}{r} \setminus \mathcal{N}$ form the bases of a matroid, called a *sparse paving matroid*.

Exercise

Prove that there are at least $2^{2^{\Omega(n)}}$ sparse paving matroids.

Observation (Göös)

Facet-defining rank inequalities for sparse paving matroid of rank r are all of the form

$$\sum_{e \in F} x_e \leqslant r - 1 = |F| - 1 \iff \sum_{e \in F} (1 - x_e) \geqslant 1$$

where $F \in \mathcal{B}$ or

$$\sum_{e \in E} x_e \leqslant r$$

Get from this:

- every depth-D monotone circuit for the function $f : \{0,1\}^E \to \{0,1\}$ such that $f(x) = 1$ iff $\overline{x}$ is independent implies a size-$O(n2^D)$ extended formulation for the matroid polytope of a sparse paving matroid.
- f is a *slice* function, so the same applies to non-monotone circuits $\implies$ huge obstacle for lower bounds

Given *sizes* $s_1, \ldots, s_n \in \mathbb{R}_+$, *demand* $D \in \mathbb{R}_+$:

- Weighted threshold function: $f(x) = 1 \iff \sum_{i=1}^n s_i x_i \geqslant D$
- *Min-knapsack polytope*: $P := \mathrm{conv}\,\{x \in \{0,1\}^n \mid \sum_{i=1}^n s_i x_i \geqslant D\}$

Question

Is there a size-$n^{O(1)}$ $O(1)$-apx EF for min-knapsack?

Given *sizes* $s_1, \ldots, s_n \in \mathbb{R}_+$, *demand* $D \in \mathbb{R}_+$:

- Weighted threshold function: $f(x) = 1 \iff \sum_{i=1}^n s_i x_i \geqslant D$
- *Min-knapsack polytope*: $P := \operatorname{conv}\{x \in \{0,1\}^n \mid \sum_{i=1}^n s_i x_i \geqslant D\}$

Question

Is there a size-$n^{O(1)}$ $O(1)$-apx EF for min-knapsack?

Theorem (Bazzi, F, Huang, Svensson '16)

$\forall \varepsilon > 0$, $\exists$ *size-*$(1/\varepsilon)^{O(1)} n^{O(\log n)}$ $(2+\varepsilon)$*-apx EF for min-knapsack*

Extended Formulations as Communication Protocols

Min-knapsack

Given *sizes* $s_1, \ldots, s_n \in \mathbb{R}_+$, *demand* $D \in \mathbb{R}_+$:

- Weighted threshold function: $f(x) = 1 \iff \sum_{i=1}^n s_i x_i \geqslant D$
- *Min-knapsack polytope*: $P := \text{conv}\,\{x \in \{0,1\}^n \mid \sum_{i=1}^n s_i x_i \geqslant D\}$

Question

Is there a size-$n^{O(1)}$ $O(1)$-apx EF for min-knapsack?

Theorem (Bazzi, F, Huang, Svensson '16)

$\forall \varepsilon > 0$, $\exists$ *size-*$(1/\varepsilon)^{O(1)} n^{O(\log n)}$ $(2+\varepsilon)$*-apx EF for min-knapsack*

Starting point:

- Hrubeš-Göös connection
- $O(\log^2(n))$-depth monotone circuits for weighted threshold functions (Beimel and Weinreb '06)
- Knapsack cover inequalities (Carr, Fleischer, Leung and Phillips '06)

Starting point:

- Hrubeš-Göös connection
- $O(\log^2(n))$-depth monotone circuits for weighted threshold functions (Beimel and Weinreb '06)
- Knapsack cover inequalities (Carr, Fleischer, Leung and Phillips '06)

For $a \in f^{-1}(0)$, if let $D(a) := D - \sum_{i=1}^{n} s_i a_i > 0$ then

$$\sum_{i:a_i=0} \min(\{s_i, D(a)\}) \cdot x_i \geqslant D(a)$$

is a valid inequality, called the (capacitated) *knapsack-cover inequality*

Starting point:

- Hrubeš-Göös connection
- $O(\log^2(n))$-depth monotone circuits for weighted threshold functions (Beimel and Weinreb '06)
- Knapsack cover inequalities (Carr, Fleischer, Leung and Phillips '06)

For $a \in f^{-1}(0)$, if let $D(a) := D - \sum_{i=1}^{n} s_i a_i > 0$ then

$$\sum_{i:a_i=0} \min(\{s_i, D(a)\}) \cdot x_i \geqslant D(a)$$

is a valid inequality, called the (capacitated) *knapsack-cover inequality*

Theorem (Carr et al. '06)

The (exponentially many) knapsack-cover inequalities provide a 2-apx of min-knapsack

Our approach:

- relax right-hand side to $\alpha D(a)$ where $\alpha = 2/(2+\varepsilon) \approx 1$
- coarse $+$ fine approximation
- try to find a "big" item i^* such that $a_{i^*} = 0$, $b_{i^*} = 1$
- use several threshold functions
- make things constructive (given an objective function!)

Extended Formulations as Communication Protocols

Making things constructive

Consider for p, q large:

- p inequalities: $A_i x \geqslant b_i$ $(i \in [p])$
- q solutions: s_j $(j \in [q])$ of this system

Data defines slack matrix $M \in \mathbb{R}_+^{p \times q}$ by $M_{ij} := A_i s_j - b_i$

Extended Formulations as Communication Protocols

Making things constructive

Consider for p, q large:

- p inequalities: $A_i x \geqslant b_i$ $(i \in [p])$
- q solutions: s_j $(j \in [q])$ of this system

Data defines slack matrix $M \in \mathbb{R}_+^{p \times q}$ by $M_{ij} := A_i s_j - b_i$

Assume: $M = FV$, where $F \in \mathbb{R}_+^{p \times r}$ and $V \in \mathbb{R}_+^{r \times q}$

Extended Formulations as Communication Protocols

Making things constructive

Consider for p, q large:

- p inequalities: $A_i x \geqslant b_i$ $(i \in [p])$
- q solutions: s_j $(j \in [q])$ of this system

Data defines slack matrix $M \in \mathbb{R}_+^{p \times q}$ by $M_{ij} := A_i s_j - b_i$

Assume: $M = FV$, where $F \in \mathbb{R}_+^{p \times r}$ and $V \in \mathbb{R}_+^{r \times q}$

This gives extended formulation

$$Ax - b = Fy, \quad y \geqslant 0$$

Extended Formulations as Communication Protocols

Making things constructive

Consider for p, q large:

- p inequalities: $A_i x \geqslant b_i$ $(i \in [p])$
- q solutions: s_j $(j \in [q])$ of this system

Data defines slack matrix $M \in \mathbb{R}_+^{p \times q}$ by $M_{ij} := A_i s_j - b_i$

Assume: $M = FV$, where $F \in \mathbb{R}_+^{p \times r}$ and $V \in \mathbb{R}_+^{r \times q}$

This gives extended formulation

$$Ax - b = Fy, \quad y \geqslant 0$$

This has p equations, only $\leqslant n + r$ are non-redundant!

Extended Formulations as Communication Protocols

Making things constructive

Consider for p, q large:

- p inequalities: $A_i x \geqslant b_i$ ($i \in [p]$)
- q solutions: s_j ($j \in [q]$) of this system

Data defines slack matrix $M \in \mathbb{R}_+^{p \times q}$ by $M_{ij} := A_i s_j - b_i$

Assume: $M = FV$, where $F \in \mathbb{R}_+^{p \times r}$ and $V \in \mathbb{R}_+^{r \times q}$

This gives extended formulation

$$Ax - b = Fy, \quad y \geqslant 0$$

This has p equations, only $\leqslant n + r$ are non-redundant!

Question

How can we pick the equations? How can we use the extended formulation in an algorithm?

In case the non-negative factorization comes from a communication protocol, can write:

$$
\begin{aligned}
A_i x - b_i &= \sum_{\ell \text{ leaf}} q_i(\ell) \cdot y_\ell \quad \forall i \in [p] \\
y_\ell &\geqslant 0 \quad \forall \ell \text{ leaf}
\end{aligned}
$$

where $q_i(u) :=$ probability of reaching node u of the protocol tree on any input pair of the form $(i, *)$

In case the non-negative factorization comes from a communication protocol, can write:

$$A_i x - b_i = \sum_{\ell \text{ leaf}} q_i(\ell) \cdot y_\ell \quad \forall i \in [p]$$
$$y_\ell \geqslant 0 \quad \forall \ell \text{ leaf}$$

where $q_i(u) :=$ probability of reaching node u of the protocol tree on any input pair of the form $(i, *)$

Lemma

Let $\Delta := \max\{-\log(q_i(\ell)) \mid i \in [p],\ \ell \text{ leaf},\ q_i(\ell) > 0\}$ and let h denote the height of the protocol tree. For any fixed $i \in [p]$, one can write down the corresponding equation in $O(2^h \Delta \log \Delta \log \log \Delta)$ time and $O(2^h \Delta)$ space.

This leads to a **cutting plane algorithm**. Given costs $c \in \mathbb{R}^n$

1. Initialize $I \subseteq [p]$
2. Solve
$$\begin{aligned} \min c^\intercal x \\ \text{s.t. } A_i x - b_i &= \textstyle\sum_{\ell \text{ leaf}} p_i(\ell) \cdot y_\ell \quad && \forall i \in I \\ y_\ell &\geqslant 0 && \forall \ell \text{ leaf} \end{aligned}$$
3. Get optimum solution x^*
4. Check if x^* violates any constraint $A_i x \geqslant b_i$ $(i \in [p])$
5. If yes, add i to I and repeat. If no, **STOP**.

This leads to a **cutting plane algorithm**. Given costs $c \in \mathbb{R}^n$

1. Initialize $I \subseteq [p]$
2. Solve

$$\begin{aligned} \min\, & c^\intercal x \\ \text{s.t. } A_i x - b_i &= \textstyle\sum_{\ell \text{ leaf}} p_i(\ell) \cdot y_\ell && \forall i \in I \\ y_\ell &\geqslant 0 && \forall \ell \text{ leaf} \end{aligned}$$

3. Get optimum solution x^*
4. Check if x^* violates any constraint $A_i x \geqslant b_i$ ($i \in [p]$)
5. If yes, add i to I and repeat. If no, **STOP**.

Observation

At most $n + r$ elements are added to I

This leads to a **cutting plane algorithm**. Given costs $c \in \mathbb{R}^n$

1. Initialize $I \subseteq [p]$
2. Solve

$$\begin{aligned} \min\ & c^\intercal x \\ \text{s.t. } A_i x - b_i &= \textstyle\sum_{\ell \text{ leaf}} p_i(\ell) \cdot y_\ell \quad && \forall i \in I \\ y_\ell &\geqslant 0 && \forall \ell \text{ leaf} \end{aligned}$$

3. Get optimum solution x^*
4. Check if x^* violates any constraint $A_i x \geqslant b_i$ $(i \in [p])$
5. If yes, add i to I and repeat. If no, **STOP**.

Observation

At most $n + r$ elements are added to I

Assuming that the separation can be done in time $T(n)$, can solve the LP in time

$$O\left((n+r)(T(n) + r\Delta \log \Delta \log\log \Delta + n^{O(1)}(n+r)\Delta)\right)$$

Announcement

It is time for a break!

More Lower-Bounding techniques

Overview

Techniques to prove **lower bounds** on non-negative rank:

More Lower-Bounding techniques

Overview

Techniques to prove **lower bounds** on non-negative rank:

1. rectangle covering lower bound

More Lower-Bounding techniques

Overview

Techniques to prove **lower bounds** on non-negative rank:

1. rectangle covering lower bound
2. hyperplane separation lower bound (Rothvoss, F.)

More Lower-Bounding techniques

Overview

Techniques to prove **lower bounds** on non-negative rank:

1. rectangle covering lower bound
2. hyperplane separation lower bound (Rothvoss, F.)
3. common information (Braun-Pokutta)

More Lower-Bounding techniques
Overview

Techniques to prove **lower bounds** on non-negative rank:

1. rectangle covering lower bound
2. hyperplane separation lower bound (Rothvoss, F.)
3. common information (Braun-Pokutta)
4. for CSPs: reduction to SA (Chan-Lee-Raghavendra-Steurer)

More Lower-Bounding techniques

Overview

Techniques to prove **lower bounds** on non-negative rank:

1. rectangle covering lower bound
2. hyperplane separation lower bound (Rothvoss, F.)
3. common information (Braun-Pokutta)
4. for CSPs: reduction to SA (Chan-Lee-Raghavendra-Steurer)
5. reductions between problems

Techniques to prove **lower bounds** on non-negative rank:

1. rectangle covering lower bound
2. hyperplane separation lower bound (Rothvoss, F.)
3. common information (Braun-Pokutta)
4. for CSPs: reduction to SA (Chan-Lee-Raghavendra-Steurer)
5. reductions between problems
6. counting (Rothvoss)

More Lower-Bounding techniques

Hyperplane separation lower bound

Consider any matrix $S \in \mathbb{R}_{+}^{k \times \ell}$

Assume that **weights** $W \in \mathbb{R}^{k \times \ell}$ ($-$'s allowed) satisfy:

$$\langle W, X \rangle \leqslant \delta \cdot ||X||_{\infty} \quad \forall X \in \mathbb{R}_{+}^{k \times \ell} \text{ that is rank-1}$$

Then if $S = \sum_{i=1}^{r} X_i$ where $X_i \in \mathbb{R}_{+}^{k \times \ell}$ are rank-1:

More Lower-Bounding techniques

Hyperplane separation lower bound

Consider any matrix $S \in \mathbb{R}_{+}^{k\times\ell}$

Assume that **weights** $W \in \mathbb{R}^{k\times\ell}$ ($-$'s allowed) satisfy:

$$\langle W, X\rangle \leqslant \delta \cdot ||X||_\infty \quad \forall X \in \mathbb{R}_{+}^{k\times\ell} \text{ that is rank-1}$$

Then if $S = \sum_{i=1}^{r} X_i$ where $X_i \in \mathbb{R}_{+}^{k\times\ell}$ are rank-1:

$$\begin{aligned}
\langle W, S\rangle &= \langle W, \sum_{i=1}^{r} X_i\rangle \\
&= \sum_{i=1}^{r} \langle W, X_i\rangle \\
&\leqslant \sum_{i=1}^{r} \delta \cdot ||X_i||_\infty \\
&\leqslant r \cdot \delta \cdot ||S||_\infty
\end{aligned}$$

More Lower-Bounding techniques

Hyperplane separation lower bound

Consider any matrix $S \in \mathbb{R}_{+}^{k \times \ell}$

Assume that **weights** $W \in \mathbb{R}^{k \times \ell}$ ($-$'s allowed) satisfy:

$$\langle W, X \rangle \leqslant \delta \cdot ||X||_\infty \quad \forall X \in \mathbb{R}_{+}^{k \times \ell} \text{ that is rank-1}$$

Then if $S = \sum_{i=1}^{r} X_i$ where $X_i \in \mathbb{R}_{+}^{k \times \ell}$ are rank-1:

$$\begin{aligned}
\langle W, S \rangle &= \langle W, \sum_{i=1}^{r} X_i \rangle \\
&= \sum_{i=1}^{r} \langle W, X_i \rangle \\
&\leqslant \sum_{i=1}^{r} \delta \cdot ||X_i||_\infty \\
&\leqslant r \cdot \delta \cdot ||S||_\infty
\end{aligned}$$

$$\Longrightarrow \boxed{r \geqslant \frac{\langle W, S \rangle}{\delta \cdot ||S||_\infty}}$$

Observation

Choose optimum δ:

$$\begin{aligned}\delta &:= \max\{\langle W, X\rangle \mid X \text{ rank-1}, X \in [0,1]^{k\times\ell}\\ &= \max\{\sum_{i,j} W_{ij} x_i y_j \mid x \in [0,1]^k, y \in [0,1]^\ell\}\\ &= \max\{\sum_{i,j} W_{ij} x_i y_j \mid x \in \{0,1\}^k, y \in \{0,1\}^\ell\}\\ &= \max\{\langle W, R\rangle \mid R \text{ rectangle}\}\end{aligned}$$

Observation

Choose optimum δ:

$$\begin{aligned}\delta &:= \max\{\langle W, X\rangle \mid X \text{ rank-1}, X \in [0,1]^{k\times\ell}\\ &= \max\{\sum_{i,j} W_{ij}x_i y_j \mid x \in [0,1]^k, y \in [0,1]^\ell\}\\ &= \max\{\sum_{i,j} W_{ij}x_i y_j \mid x \in \{0,1\}^k, y \in \{0,1\}^\ell\}\\ &= \max\{\langle W, R\rangle \mid R \text{ rectangle}\}\end{aligned}$$

The **hyperplane separation bound** was used to prove:

- UDISJ(n) still has super-polynomial non-negative rank even if add same number ρ everywhere, as long as $\rho \leqslant n^{1/2-\varepsilon}$ (Braun, F, Pokutta & Steurer)
- $\mathsf{xc}(P_{\mathrm{match}}(K_n)) = 2^{\Theta(n)}$ (Rothvoss'14)

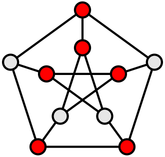
More Lower Bounding Techniques

A Sample Result

$G = (V, E)$ n-vertex graph

Definition

$U \subseteq V$ is a *vertex cover* if every edge has at least one endpoint in U



Basic IP for VC:

$$\begin{array}{lll} \min & \sum_{i \in V} c_i x_i & \\ \text{s.t.} & x_i + x_j \geqslant 1 & \forall ij \in E \\ & x_i \in \{0,1\} & \forall i \in V \end{array}$$

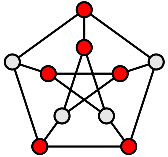
More Lower Bounding Techniques

A Sample Result

$G = (V, E)$ n-vertex graph

Definition

$U \subseteq V$ is a *vertex cover* if every edge has at least one endpoint in U



Basic ~~IP~~ LP for VC:

$$
\begin{array}{lll}
\min & \displaystyle\sum_{i \in V} c_i x_i & \\
\text{s.t.} & x_i + x_j \geqslant 1 & \forall ij \in E \\
 & \cancel{x_i \in \{0,1\}} & \forall i \in V \\
 & 0 \leqslant x_i \leqslant 1 &
\end{array}
$$

Good/Bad news

The basic LP has *integrality gap* $:= \sup_{G,c} \frac{\mathsf{OPT}}{\mathrm{LP}} = 2$

Good/Bad news

The basic LP has *integrality gap* $:= \sup_{G,c} \frac{\mathsf{OPT}}{\mathrm{LP}} = 2$

How can we improve on this?

Good/Bad news

The basic LP has *integrality gap* $:= \sup_{G,c} \frac{\mathsf{OPT}}{\mathrm{LP}} = 2$

How can we improve on this?

1. **Explicit constraints**: e.g., if G contains odd cycle C then

$$\sum_{i \in V(C)} x_i \geqslant \frac{|C|+1}{2}$$

is a valid inequality $\implies$ can add it to the LP ($\exists$ many more!)

Good/Bad news

The basic LP has *integrality gap* $:= \sup_{G,c} \frac{\mathsf{OPT}}{\mathrm{LP}} = 2$

How can we improve on this?

1. **Explicit constraints**: e.g., if G contains odd cycle C then

$$\sum_{i \in V(C)} x_i \geqslant \frac{|C|+1}{2}$$

   is a valid inequality $\implies$ can add it to the LP ($\exists$ many more!)

2. **Implicit constraints** (lift-and-project): e.g., Sherali-Adams hierarchy strengthens current relaxation $Ax \geqslant b$ by
   - Multiplying each inequality of $Ax \geqslant b$ by $\prod_{i \in I} x_i \prod_{i \in J} (1 - x_i)$ for each pair (I, J) of disjoint subsets of $[n]$ with $|I| + |J| = r$
   - Expand using $x_i^2 = x_i$
   - Replace each monomial $\prod_{i \in S} x_i$ by a variable y_S where $S \subseteq [n]$, $|S| \leqslant r + 1$
   - Add the constraint $y_\emptyset = 1$
   - Project by letting $x_i := y_{\{i\}}$

Does this help?

Does this help? “NO”

Does this help? “NO”

Theorem (ABLT’06)

1. *Adding all inequalities with support at most* ϵn *leaves a gap* $\geqslant 2 - \epsilon$
2. *Performing* $O(\log n)$ *rounds of lift-and-project (LS) leaves a gap of* 2

Does this help? “NO”

Theorem (ABLT'06)

1. *Adding all inequalities with support at most* ϵn *leaves a gap* $\geqslant 2 - \epsilon$
2. *Performing* $O(\log n)$ *rounds of lift-and-project (LS) leaves a gap of* 2

$\Longrightarrow$ No obvious way to improve the integrality gap!

Does this help? "NO"

Theorem (ABLT'06)

1. *Adding all inequalities with support at most ϵn leaves a gap $\geqslant 2 - \epsilon$*
2. *Performing $O(\log n)$ rounds of lift-and-project (LS) leaves a gap of 2*

$\Longrightarrow$ No obvious way to improve the integrality gap!

Many follow-up works on **lift-and-project hierarchies**:

- STT'07: $o(n)$ rounds of LS do not help
- GMPT'07: $O(\sqrt{\log n / \log\log n})$ rounds of LS$_+$ do not help
- CMM'09: n^δ rounds of SA do not help (for small $\delta > 0$)

We generalize the results on hierarchies and prove that

every polynomial size LP for VC **has integrality gap** 2

$\Pi = (\mathcal{S}, \mathfrak{I})$ (min) problem

$\Pi = (\mathcal{S}, \mathfrak{I})$ (min) problem

$\mathrm{VC}(G)$ for **fixed** G

$\Pi = (\mathcal{S}, \mathfrak{I})$ (min) problem

$\mathcal{S}$ set of solutions

$\mathrm{VC}(G)$ for **fixed** G

$\Pi = (\mathcal{S}, \mathfrak{I})$ (min) problem

$\mathcal{S}$ set of solutions

$\mathrm{VC}(G)$ for **fixed** G

$U \subseteq V$ vertex cover

$\Pi = (\mathcal{S}, \mathfrak{I})$ (min) problem

$\mathcal{S}$ set of solutions

$\mathfrak{I}$ set of instances

$\mathrm{VC}(G)$ for **fixed** G

$U \subseteq V$ vertex cover

$\Pi = (\mathcal{S}, \mathfrak{I})$ (min) problem

$\mathcal{S}$ set of solutions

$\mathfrak{I}$ set of instances

$\mathrm{VC}(G)$ for **fixed** G

$U \subseteq V$ vertex cover

$\mathcal{I} = \mathcal{I}(c)$ with $c \in \mathbb{R}_+^V$ cost vector

$\Pi = (\mathcal{S}, \mathfrak{I})$ (min) problem

$\mathcal{S}$ set of solutions

$\mathfrak{I}$ set of instances

$\forall \mathcal{I} \in \mathfrak{I}$ have $\text{Cost}_{\mathcal{I}} : \mathcal{S} \to \mathbb{R}_+$ obj fn

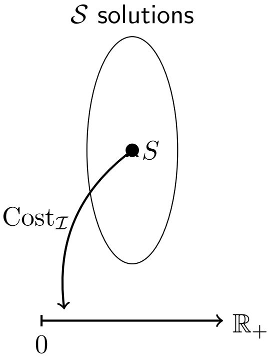
$\mathrm{VC}(G)$ for **fixed** G

$U \subseteq V$ vertex cover

$\mathcal{I} = \mathcal{I}(c)$ with $c \in \mathbb{R}_+^V$ cost vector

$\Pi = (\mathcal{S}, \mathfrak{I})$ (min) problem	$\mathrm{VC}(G)$ for **fixed** G
$\mathcal{S}$ set of solutions	$U \subseteq V$ vertex cover
$\mathfrak{I}$ set of instances	$\mathcal{I} = \mathcal{I}(c)$ with $c \in \mathbb{R}_+^V$ cost vector
$\forall \mathcal{I} \in \mathfrak{I}$ have $\mathsf{Cost}_{\mathcal{I}} : \mathcal{S} \to \mathbb{R}_+$ obj fn	$\mathsf{Cost}_{\mathcal{I}}(U) := \sum_{i \in U} c_i = c(U)$

$\Pi = (\mathcal{S}, \mathfrak{I})$ (min) problem	$\mathrm{VC}(G)$ for **fixed** G
$\mathcal{S}$ set of solutions	$U \subseteq V$ vertex cover
$\mathfrak{I}$ set of instances	$\mathcal{I} = \mathcal{I}(c)$ with $c \in \mathbb{R}_+^V$ cost vector
$\forall \mathcal{I} \in \mathfrak{I}$ have $\mathrm{Cost}_{\mathcal{I}} : \mathcal{S} \to \mathbb{R}_+$ obj fn	$\mathrm{Cost}_{\mathcal{I}}(U) := \sum_{i \in U} c_i = c(U)$

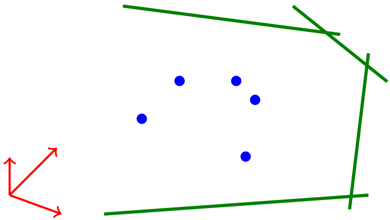


Setting up an LP relaxation:

1. pick d, system $Ax \geqslant b$ in $\mathbb{R}^d$
2. realize each $S \in \mathcal{S}$ as point x^S in $\mathbb{R}^d$ s.t. $\boxed{Ax^S \geqslant b}$
3. realize each $\mathcal{I} \in \mathfrak{I}$ as affine fn $f_{\mathcal{I}}(x)$ on $\mathbb{R}^d$ s.t. $\boxed{f_{\mathcal{I}}(x^S) = \mathsf{Cost}_{\mathcal{I}}(S)}$

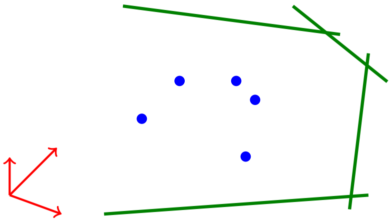
Setting up an LP relaxation:

1. pick d, system $Ax \geqslant b$ in $\mathbb{R}^d$
2. realize each $S \in \mathcal{S}$ as point x^S in $\mathbb{R}^d$ s.t. $\boxed{Ax^S \geqslant b}$
3. realize each $\mathcal{I} \in \mathfrak{I}$ as affine fn $f_{\mathcal{I}}(x)$ on $\mathbb{R}^d$ s.t. $\boxed{f_{\mathcal{I}}(x^S) = \mathsf{Cost}_{\mathcal{I}}(S)}$



Setting up an LP relaxation:

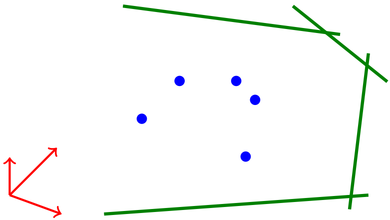
1. pick d, system $Ax \geqslant b$ in $\mathbb{R}^d$
2. realize each $S \in \mathcal{S}$ as point x^S in $\mathbb{R}^d$ s.t. $\boxed{Ax^S \geqslant b}$
3. realize each $\mathcal{I} \in \mathfrak{I}$ as affine fn $f_{\mathcal{I}}(x)$ on $\mathbb{R}^d$ s.t. $\boxed{f_{\mathcal{I}}(x^S) = \mathsf{Cost}_{\mathcal{I}}(S)}$



$$\text{Instance } \mathcal{I} \xrightarrow{f_{\mathcal{I}}(x)} \boxed{Ax \geqslant b} \xrightarrow{\mathsf{LB}} \mathsf{LP}(\mathcal{I}) := \min\{f_{\mathcal{I}}(x) \mid Ax \geqslant b\}$$

Setting up an LP relaxation:

1. pick d, system $Ax \geqslant b$ in $\mathbb{R}^d$
2. realize each $S \in \mathcal{S}$ as point x^S in $\mathbb{R}^d$ s.t. $\boxed{Ax^S \geqslant b}$
3. realize each $\mathcal{I} \in \mathfrak{I}$ as affine fn $f_{\mathcal{I}}(x)$ on $\mathbb{R}^d$ s.t. $\boxed{f_{\mathcal{I}}(x^S) = \mathsf{Cost}_{\mathcal{I}}(S)}$

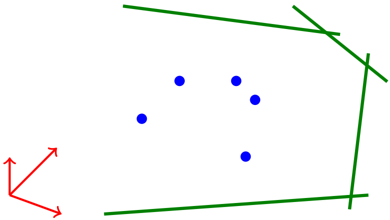


$$\text{Instance } \mathcal{I} \xrightarrow{f_{\mathcal{I}}(x)} \boxed{Ax \geqslant b} \xrightarrow{\text{LB}} \text{LP}(\mathcal{I}) := \min\{f_{\mathcal{I}}(x) \mid Ax \geqslant b\}$$

Always: $\mathrm{LP}(\mathcal{I}) \leqslant \mathsf{OPT}(\mathcal{I})$

Setting up an LP relaxation:

1. pick d, system $Ax \geqslant b$ in $\mathbb{R}^d$
2. realize each $S \in \mathcal{S}$ as point x^S in $\mathbb{R}^d$ s.t. $\boxed{Ax^S \geqslant b}$
3. realize each $\mathcal{I} \in \mathfrak{I}$ as affine fn $f_{\mathcal{I}}(x)$ on $\mathbb{R}^d$ s.t. $\boxed{f_{\mathcal{I}}(x^S) = \mathsf{Cost}_{\mathcal{I}}(S)}$



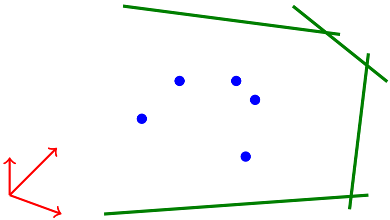
$$\text{Instance } \mathcal{I} \xrightarrow{f_{\mathcal{I}}(x)} \boxed{Ax \geqslant b} \xrightarrow{\mathsf{LB}} \mathsf{LP}(\mathcal{I}) := \min\{f_{\mathcal{I}}(x) \mid Ax \geqslant b\}$$

Always: $\mathrm{LP}(\mathcal{I}) \leqslant \mathsf{OPT}(\mathcal{I})$

ρ-**apx LP if**: $\mathsf{OPT}(\mathcal{I}) \leqslant \rho\,\mathrm{LP}(\mathcal{I})$

Setting up an LP relaxation:

1. pick d, system $Ax \geqslant b$ in $\mathbb{R}^d$
2. realize each $S \in \mathcal{S}$ as point x^S in $\mathbb{R}^d$ s.t. $\boxed{Ax^S \geqslant b}$
3. realize each $\mathcal{I} \in \mathfrak{I}$ as affine fn $f_{\mathcal{I}}(x)$ on $\mathbb{R}^d$ s.t. $\boxed{f_{\mathcal{I}}(x^S) = \mathsf{Cost}_{\mathcal{I}}(S)}$



$$\text{Instance } \mathcal{I} \xrightarrow{f_{\mathcal{I}}(x)} \boxed{Ax \geqslant b} \xrightarrow{\text{LB}} \text{LP}(\mathcal{I}) := \min\{f_{\mathcal{I}}(x) \mid Ax \geqslant b\}$$

Always: $\mathrm{LP}(\mathcal{I}) \leqslant \mathsf{OPT}(\mathcal{I})$

ρ-**apx LP if**: $\mathsf{OPT}(\mathcal{I}) \leqslant \rho\,\mathrm{LP}(\mathcal{I})$

Similar for **max** problems, e.g., $\mathrm{IS}(G)$ = max independent set problem on G

Definition

Size of LP relaxation $Ax \geqslant b := \#$ **inequalities** of $Ax \geqslant b$

Definition

Size of LP relaxation $Ax \geqslant b := \#$ **inequalities** of $Ax \geqslant b$

Main question

Tradeoff between **quality** ρ and **size** for LPs of a given problem?

Definition

Size of LP relaxation $Ax \geqslant b := \#$ **inequalities** of $Ax \geqslant b$

Main question

Tradeoff between **quality** ρ and **size** for LPs of a given problem?

Theorem (BFPS'15)

$\forall \epsilon > 0$, $\forall n_0$, $\exists$ *graph* G *with* $n \geqslant n_0$ *vertices such that*

- *every* $(2-\epsilon)$*-apx LP relaxation for* $\mathrm{VC}(G)$ *has size* $n^{\Omega\left(\frac{\log n}{\log\log n}\right)}$
- *every* $(1/\epsilon)$*-apx LP relaxation for* $\mathrm{IS}(G)$ *has size* $n^{\Omega\left(\frac{\log n}{\log\log n}\right)}$

Definition

Size of LP relaxation $Ax \geqslant b := \#$ **inequalities** of $Ax \geqslant b$

Main question

Tradeoff between **quality** ρ and **size** for LPs of a given problem?

Theorem (BFPS'15)

$\forall \epsilon > 0$, $\forall n_0$, $\exists$ *graph* G *with* $n \geqslant n_0$ *vertices such that*

- *every* $(2 - \epsilon)$*-apx LP relaxation for* $\mathrm{VC}(G)$ *has size* $n^{\Omega\left(\frac{\log n}{\log \log n}\right)}$
- *every* $(1/\epsilon)$*-apx LP relaxation for* $\mathrm{IS}(G)$ *has size* $n^{\Omega\left(\frac{\log n}{\log \log n}\right)}$

- Answers questions by Singh'10, CLRS'13

Definition

Size of LP relaxation $Ax \geqslant b := \#$ **inequalities** of $Ax \geqslant b$

Main question

Tradeoff between **quality** ρ and **size** for LPs of a given problem?

Theorem (BFPS'15)

$\forall \epsilon > 0$, $\forall n_0$, $\exists$ *graph* G *with* $n \geqslant n_0$ *vertices such that*

- *every* $(2-\epsilon)$*-apx LP relaxation for* $\mathrm{VC}(G)$ *has size* $n^{\Omega\left(\frac{\log n}{\log\log n}\right)}$
- *every* $(1/\epsilon)$*-apx LP relaxation for* $\mathrm{IS}(G)$ *has size* $n^{\Omega\left(\frac{\log n}{\log\log n}\right)}$

- Answers questions by Singh'10, CLRS'13
- "Geometric" limitations to what small LPs can express

Definition

Size of LP relaxation $Ax \geqslant b$:= # **inequalities** of $Ax \geqslant b$

Main question

Tradeoff between **quality** ρ and **size** for LPs of a given problem?

Theorem (BFPS'15)

$\forall \epsilon > 0$, $\forall n_0$, $\exists$ *graph* G *with* $n \geqslant n_0$ *vertices such that*

- *every* $(2-\epsilon)$*-apx LP relaxation for* $\mathrm{VC}(G)$ *has size* $n^{\Omega\left(\frac{\log n}{\log \log n}\right)}$
- *every* $(1/\epsilon)$*-apx LP relaxation for* $\mathrm{IS}(G)$ *has size* $n^{\Omega\left(\frac{\log n}{\log \log n}\right)}$

- Answers questions by Singh'10, CLRS'13
- "Geometric" limitations to what small LPs can express
- Strong consequence of NP $\not\subseteq$ P/poly that can be proved without assuming NP $\not\subseteq$ P/poly

Remark

We just care about the *existence or non-existence* of LP relaxations:

- **do not bound** the time it takes to write down the LP (important)
- **do not bound** the coefficients! (not so important)

Remark

We just care about the *existence or non-existence* of LP relaxations:

- **do not bound** the time it takes to write down the LP (important)
- **do not bound** the coefficients! (not so important)

From FJ'14:

$\exists\ O(n)$**-size** $O(\sqrt{n})$**-apx LP relaxation for IS**(G) **for any graph** G

$$\begin{array}{ll} \max & \sum_{i \in V} w_i x_i \\ \text{s.t.} & 0 \leqslant x_i \leqslant y_j \leqslant 1 \quad \forall i \in S_j \\ & \sum_j y_j \leqslant \lfloor \sqrt{2n} \rfloor \end{array}$$

Remark

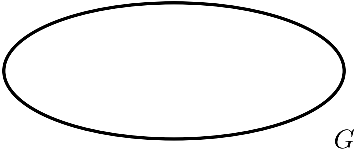
We just care about the *existence or non-existence* of LP relaxations:

- **do not bound** the time it takes to write down the LP (important)
- **do not bound** the coefficients! (not so important)

From FJ'14:

$\exists$ $O(n)$**-size** $O(\sqrt{n})$**-apx LP relaxation for IS**(G) **for any graph** G

$$\begin{array}{ll} \max & \sum_{i \in V} w_i x_i \\ \text{s.t.} & 0 \leqslant x_i \leqslant y_j \leqslant 1 \quad \forall i \in S_j \\ & \sum_j y_j \leqslant \lfloor \sqrt{2n} \rfloor \end{array}$$



G

Remark

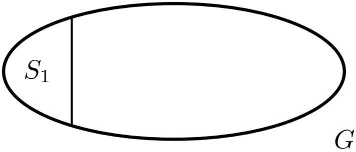
We just care about the *existence or non-existence* of LP relaxations:

- **do not bound** the time it takes to write down the LP (important)
- **do not bound** the coefficients! (not so important)

From FJ'14:

$\exists$ $O(n)$-**size** $O(\sqrt{n})$-**apx LP relaxation for IS**(G) **for any graph** G

$$\begin{array}{ll} \max & \sum_{i \in V} w_i x_i \\ \text{s.t.} & 0 \leqslant x_i \leqslant y_j \leqslant 1 \quad \forall i \in S_j \\ & \sum_j y_j \leqslant \lfloor \sqrt{2n} \rfloor \end{array}$$



Remark

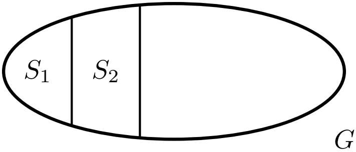
We just care about the *existence or non-existence* of LP relaxations:

- **do not bound** the time it takes to write down the LP (important)
- **do not bound** the coefficients! (not so important)

From FJ'14:

$\exists$ $O(n)$**-size** $O(\sqrt{n})$**-apx LP relaxation for IS**(G) **for any graph** G

$$\begin{array}{ll} \max & \sum_{i \in V} w_i x_i \\ \text{s.t.} & 0 \leqslant x_i \leqslant y_j \leqslant 1 \quad \forall i \in S_j \\ & \sum_j y_j \leqslant \lfloor \sqrt{2n} \rfloor \end{array}$$



Remark

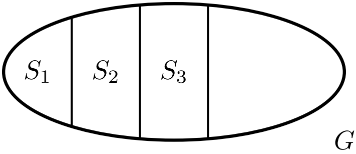
We just care about the *existence or non-existence* of LP relaxations:

- **do not bound** the time it takes to write down the LP (important)
- **do not bound** the coefficients! (not so important)

From FJ'14:

$\exists$ $O(n)$-**size** $O(\sqrt{n})$-**apx LP relaxation for IS**(G) **for any graph** G

$$\begin{array}{ll} \max & \sum_{i \in V} w_i x_i \\ \text{s.t.} & 0 \leqslant x_i \leqslant y_j \leqslant 1 \quad \forall i \in S_j \\ & \sum_j y_j \leqslant \lfloor \sqrt{2n} \rfloor \end{array}$$



Remark

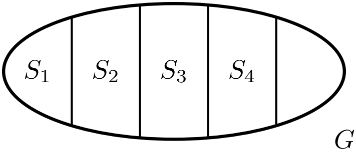
We just care about the *existence or non-existence* of LP relaxations:

- **do not bound** the time it takes to write down the LP (important)
- **do not bound** the coefficients! (not so important)

From FJ'14:

$\exists$ $O(n)$**-size** $O(\sqrt{n})$**-apx LP relaxation for IS**(G) **for any graph** G

$$
\begin{array}{ll}
\max & \sum_{i \in V} w_i x_i \\
\text{s.t.} & 0 \leqslant x_i \leqslant y_j \leqslant 1 \quad \forall i \in S_j \\
& \sum_j y_j \leqslant \lfloor \sqrt{2n} \rfloor
\end{array}
$$



Remark

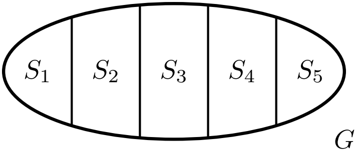
We just care about the *existence or non-existence* of LP relaxations:

- **do not bound** the time it takes to write down the LP (important)
- **do not bound** the coefficients! (not so important)

From FJ'14:

$\exists$ $O(n)$**-size** $O(\sqrt{n})$**-apx LP relaxation for IS**(G) **for any graph** G

$$\begin{array}{ll} \max & \sum_{i \in V} w_i x_i \\ \text{s.t.} & 0 \leqslant x_i \leqslant y_j \leqslant 1 \quad \forall i \in S_j \\ & \sum_j y_j \leqslant \lfloor \sqrt{2n} \rfloor \end{array}$$



Constraint satisfaction problems

Parameters of the problem:

- number of variables n
- *domain size* R
- *arity* k

Constraint satisfaction problems

Parameters of the problem:

- number of variables n
- *domain size* R
- *arity* k

Definition (Max-CSP(n, R, k))

- **Solutions:** assignments $x \in [R]^n$ of values to the vars $x_1, \ldots, x_n$
- **Instances:** sets $\mathcal{I} := \{P_1, \ldots, P_m\}$ of Boolean predicates $P_i : [R]^n \to \{0,1\} : x \mapsto P_i(x)$, each depending on k variables
- **Objective function:** $\mathsf{Val}_{\mathcal{I}}(x) := \frac{1}{m} \sum_{i=1}^{m} P_i(x)$

Constraint satisfaction problems

Parameters of the problem:

- number of variables n
- *domain size* R
- *arity* k

Definition (Max-CSP(n, R, k))

- **Solutions:** assignments $x \in [R]^n$ of values to the vars $x_1, \ldots, x_n$
- **Instances:** sets $\mathcal{I} := \{P_1, \ldots, P_m\}$ of Boolean predicates $P_i : [R]^n \to \{0, 1\} : x \mapsto P_i(x)$, each depending on k variables
- **Objective function:** $\mathsf{Val}_{\mathcal{I}}(x) := \frac{1}{m} \sum_{i=1}^{m} P_i(x)$

Example (Max-Cut(n))

- arity $k = 2$
- domain size $R = 2$ and domain $\{0, 1\}$
- predicates of the form $P(x) = x_i \oplus x_j$

Recent results

BPZ'14 prove super-polynomial lower bounds for

- $(1.5 - \epsilon)$-apx LP relaxations of $\mathrm{VC}(G)$
- $(2 - \epsilon)$-apx LP relaxations of $\mathrm{IS}(G)$

Recent results

BPZ'14 prove super-polynomial lower bounds for

- $(1.5 - \epsilon)$-apx LP relaxations of $\mathrm{VC}(G)$
- $(2 - \epsilon)$-apx LP relaxations of $\mathrm{IS}(G)$

by combining

- lower bound of CLRS'13 for $(2 - \epsilon)$-apx LP relaxations of $\textsc{Max-Cut}$
- reductions from $\textsc{Max-Cut}(n)$ to $\mathrm{VC}(G)$ and $\mathrm{IS}(G)$

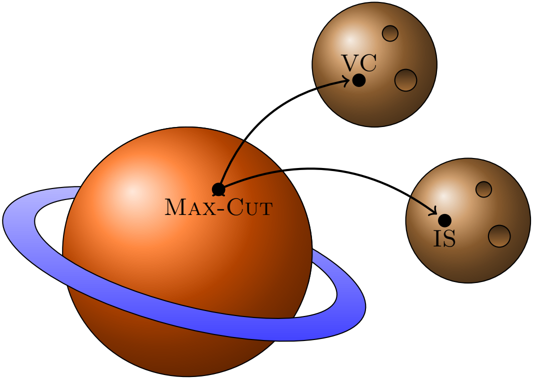
Recent results

BPZ'14 prove super-polynomial lower bounds for

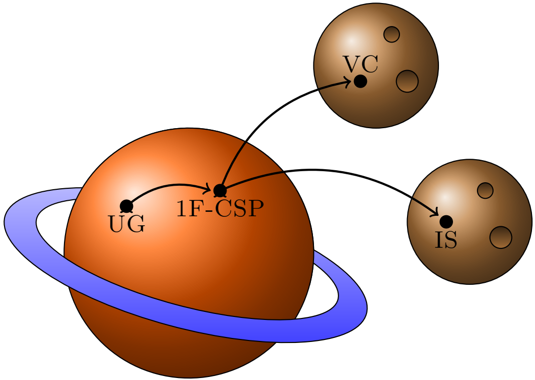
- $(1.5 - \epsilon)$-apx LP relaxations of $\mathrm{VC}(G)$
- $(2 - \epsilon)$-apx LP relaxations of $\mathrm{IS}(G)$

by combining

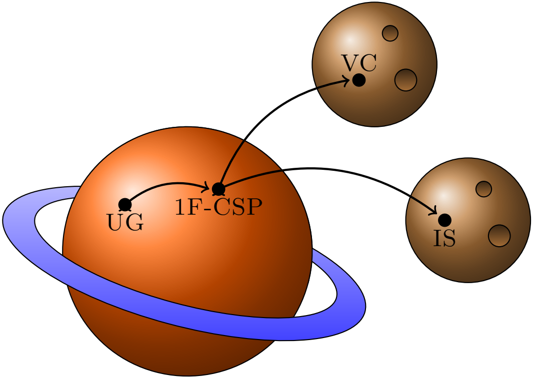
- lower bound of CLRS'13 for $(2 - \epsilon)$-apx LP relaxations of $\textsc{Max-Cut}$
- reductions from $\textsc{Max-Cut}(n)$ to $\mathrm{VC}(G)$ and $\mathrm{IS}(G)$



Our proof



Our proof



We use:

- a different starting point: UG (unique games)
- CMM'09: UG fools $O(n^{\delta})$ rounds of Sherali-Adams
- a different leaving point: $\mathrm{1F\text{-}CSP}$ (one free bit CSPs)
- CLRS'13: Sherali-Adams is "optimal" among all LPs for CSPs

One free bit CSPs

Definition (1F-CSP(n, k))

- **Solutions:** assignments $x \in \{0,1\}^n$
- **Instances:** sets $\mathcal{I} := \{P_1, \ldots, P_m\}$ of Boolean predicates $P_i : \{0,1\}^n \to \{0,1\} : x \mapsto P_i(x)$, each depending on k of the n variables, and each having exactly **two** satisfying assignment on these k bits
- **Objective function:** $\mathsf{Val}_{\mathcal{I}}(x) := \frac{1}{m} \sum_{i=1}^{m} P_i(x)$

Rem:

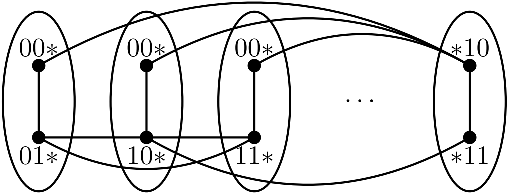
- $\textsc{Max-Cut}(n)$ is a special case for $k = 2$

FGLSS graphs

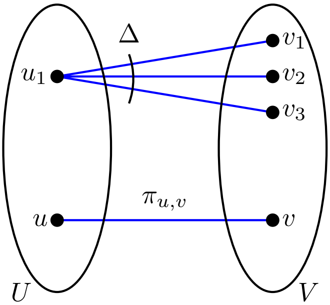
after Feige, Goldwasser, Lovász, Safra and Szegedy '96

$G^*(n,k)$ is the graph with:

- two vertices per one free bit predicate $P : \{0,1\}^S \to \{0,1\}$ where $S \subseteq [n]$, $|S| = k$
- one vertex for each partial assignment $\alpha \in \{0,1\}^S$ satisfying P
- edges between conflicting partial assignments



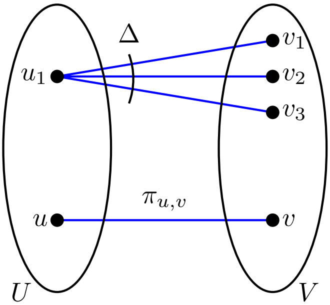
Unique games



Given:

- $G = (U, V, E)$ bipartite graph, Δ-regular
- domain $[R]$
- permutation $\pi_{u,v} : [R] \to [R]$ for each edge $uv \in E$

Unique games



Given:

- $G = (U, V, E)$ bipartite graph, Δ-regular
- domain $[R]$
- permutation $\pi_{u,v} : [R] \to [R]$ for each edge $uv \in E$

UG (unique games) is the $\textsc{Max-CSP}$ with

- arity $k = 2$
- one predicate per edge $uv \in E$ that is true iff $\pi_{u,v}(x_u) = x_v$

Unique games conjecture

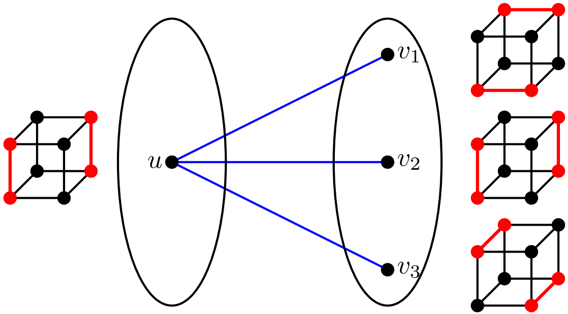
Conjecture (Khot'02)

For every $\epsilon, \delta > 0$, there exists a sufficiently large domain size $R = R(\epsilon, \delta)$ such that the following promise problem is NP-hard. Given a UG instance $(G, [R], (\pi_{uv})_{uv \in E})$, distinguish between the following two cases:

1. $\exists$ assignment satisfying at least a $(1 - \epsilon)$-fraction of the edges;
2. $\nexists$ assignment satisfying more than a δ-fraction of the edges.

The Bansal-Khot PCP and 1F-CSP's

For every bipartite, Δ-regular UG instance $(G, [R], (\pi_{uv})_{uv \in E})$ we obtain a $1\mathrm{F}\text{-}\mathrm{CSP}$ instance by reinterpreting the PCP due to BK'09



Reductions

Consider:

- $\Pi_1 = (\mathcal{S}_1, \mathfrak{I}_1)$ **max** problem
- $\Pi_2 = (\mathcal{S}_2, \mathfrak{I}_2)$ **min** problem

Reductions

Consider:

- $\Pi_1 = (\mathcal{S}_1, \mathfrak{I}_1)$ **max** problem
- $\Pi_2 = (\mathcal{S}_2, \mathfrak{I}_2)$ **min** problem

A *reduction from* Π_1 *to* Π_2 is defined by two maps

1. $\mathfrak{I}_1 \to \mathfrak{I}_2 : \mathcal{I}_1 \mapsto \mathcal{I}_2$
2. $\mathcal{S}_1 \to \mathcal{S}_2 : S_1 \mapsto S_2$

subject to:

$$\mathsf{Val}_{\mathcal{I}_1}(S_1) = \underbrace{\mu(\mathcal{I}_1)}_{\text{affine shift}} - \underbrace{\alpha(\mathcal{I}_1)}_{\geqslant 0} \cdot \mathrm{Cost}_{\mathcal{I}_2}(S_2) \qquad \forall \mathcal{I}_1 \in \mathfrak{I}_1, S_1 \in \mathcal{S}_1$$

Reductions

Consider:

- $\Pi_1 = (\mathcal{S}_1, \mathfrak{I}_1)$ **max** problem
- $\Pi_2 = (\mathcal{S}_2, \mathfrak{I}_2)$ **min** problem

A *reduction from* Π_1 *to* Π_2 is defined by two maps

1. $\mathfrak{I}_1 \to \mathfrak{I}_2 : \mathcal{I}_1 \mapsto \mathcal{I}_2$
2. $\mathcal{S}_1 \to \mathcal{S}_2 : S_1 \mapsto S_2$

subject to:

$$\mathsf{Val}_{\mathcal{I}_1}(S_1) = \underbrace{\mu(\mathcal{I}_1)}_{\text{affine shift}} - \underbrace{\alpha(\mathcal{I}_1)}_{\geqslant 0} \cdot \mathrm{Cost}_{\mathcal{I}_2}(S_2) \qquad \forall \mathcal{I}_1 \in \mathfrak{I}_1, S_1 \in \mathcal{S}_1$$

The reduction is *exact* if additionally

$$\mathsf{OPT}(\mathcal{I}_1) = \mu(\mathcal{I}_1) - \alpha(\mathcal{I}_1) \cdot \mathsf{OPT}(\mathcal{I}_2) \qquad \forall \mathcal{I}_1 \in \mathfrak{I}_1$$

Definition

For $c \geqslant s > 0$, an LP relaxation is a (c,s)-apx for a **max** problem $\Pi = (\mathcal{S}, \mathfrak{I})$ if

$$\forall \mathcal{I} \in \mathfrak{I} : \qquad \mathsf{OPT}(\mathcal{I}) \leqslant s \quad \Longrightarrow \quad \mathrm{LP}(\mathcal{I}) \leqslant c$$

Definition

For $c \geqslant s > 0$, an LP relaxation is a (c, s)-apx for a **max** problem $\Pi = (\mathcal{S}, \mathfrak{I})$ if

$$\forall \mathcal{I} \in \mathfrak{I} : \qquad \mathsf{OPT}(\mathcal{I}) \leqslant s \quad \Longrightarrow \quad \mathrm{LP}(\mathcal{I}) \leqslant c$$

Intuitively:

> *"Whenever the optimum is small, the LP knows that the optimum is somewhat small."*

Definition

For $c \geqslant s > 0$, an LP relaxation is a (c, s)-apx for a **max** problem $\Pi = (\mathcal{S}, \mathfrak{I})$ if

$$\forall \mathcal{I} \in \mathfrak{I} : \qquad \mathsf{OPT}(\mathcal{I}) \leqslant s \quad \Longrightarrow \quad \mathrm{LP}(\mathcal{I}) \leqslant c$$

Intuitively:

> *"Whenever the optimum is small, the LP knows that the optimum is somewhat small."*

Rem:

- c is the *completeness*
- s is the *soundness*

Theorem (BPZ'14)

Let Π_1 be a max problem and let Π_2 be a min problem. Suppose $\exists$ an **exact** *reduction from Π_1 to Π_2 with constant affine shift $\mu(\mathcal{I}_1) = \mu$. Then, every ρ_2-apx LP relaxation for Π_2 gives a (c_1, s_1)-apx LP relaxation for Π_1, of the same size, where*

$$\rho_2 = \frac{\mu - s_1}{\mu - c_1}$$

Theorem (BPZ'14)

Let Π_1 be a max problem and let Π_2 be a min problem. Suppose $\exists$ an **exact** *reduction from Π_1 to Π_2 with constant affine shift $\mu(\mathcal{I}_1) = \mu$. Then, every ρ_2-apx LP relaxation for Π_2 gives a (c_1, s_1)-apx LP relaxation for Π_1, of the same size, where*

$$\rho_2 = \frac{\mu - s_1}{\mu - c_1}$$

Proof. Consider ρ_2-apx LP relaxation $Ax \geq b$ for Π_2 with realizations

- x^{S_2} for $S_2 \in \mathcal{S}_2$
- $f_{\mathcal{I}_2} : \mathbb{R}^d \to \mathbb{R}$ for $\mathcal{I}_2 \in \mathfrak{I}_2$

Use same LP $Ax \geq b$ for Π_1 with realizations

- $x^{S_1} := x^{S_2}$ with $S_1 \mapsto S_2$
- $f_{\mathcal{I}_1}(x) := \mu - \alpha(\mathcal{I}_1) f_{\mathcal{I}_2}(x)$ with $\mathcal{I}_1 \mapsto \mathcal{I}_2$

We claim this is a (c_1, s_1)-apx for Π_1

Proof (continued). Using:

$$\mathsf{OPT}(\mathcal{I}_2) \leqslant \rho_2 \, \mathrm{LP}(\mathcal{I}_2) \qquad \text{and} \qquad \mathsf{OPT}(\mathcal{I}_1) = \mu - \alpha(\mathcal{I}_1)\, \mathsf{OPT}(\mathcal{I}_2)$$

we have:

$$\mathrm{LP}(\mathcal{I}_1)$$

Proof (continued). Using:

$$\mathsf{OPT}(\mathcal{I}_2) \leqslant \rho_2 \, \mathrm{LP}(\mathcal{I}_2) \qquad \text{and} \qquad \mathsf{OPT}(\mathcal{I}_1) = \mu - \alpha(\mathcal{I}_1) \, \mathsf{OPT}(\mathcal{I}_2)$$

we have:

$$\mathrm{LP}(\mathcal{I}_1) = \max\{\mu - \alpha(\mathcal{I}_1) f_{\mathcal{I}_2}(x) \mid Ax \geqslant b\}$$

Proof (continued). Using:

$$\mathsf{OPT}(\mathcal{I}_2) \leqslant \rho_2 \, \mathrm{LP}(\mathcal{I}_2) \qquad \text{and} \qquad \mathsf{OPT}(\mathcal{I}_1) = \mu - \alpha(\mathcal{I}_1) \, \mathsf{OPT}(\mathcal{I}_2)$$

we have:

$$\begin{aligned} \mathrm{LP}(\mathcal{I}_1) &= \max\{\mu - \alpha(\mathcal{I}_1) f_{\mathcal{I}_2}(x) \mid Ax \geqslant b\} \\ &= \mu - \alpha(\mathcal{I}_1) \min\{f_{\mathcal{I}_2}(x) \mid Ax \geqslant b\} \end{aligned}$$

Proof (continued). Using:

$$\mathsf{OPT}(\mathcal{I}_2) \leqslant \rho_2 \, \mathrm{LP}(\mathcal{I}_2) \qquad \text{and} \qquad \mathsf{OPT}(\mathcal{I}_1) = \mu - \alpha(\mathcal{I}_1) \, \mathsf{OPT}(\mathcal{I}_2)$$

we have:

$$\begin{aligned} \mathrm{LP}(\mathcal{I}_1) &= \max\{\mu - \alpha(\mathcal{I}_1) f_{\mathcal{I}_2}(x) \mid Ax \geqslant b\} \\ &= \mu - \alpha(\mathcal{I}_1) \min\{f_{\mathcal{I}_2}(x) \mid Ax \geqslant b\} \\ &= \mu - \alpha(\mathcal{I}_1) \, \mathrm{LP}(\mathcal{I}_2) \end{aligned}$$

Proof (continued). Using:

$$\mathsf{OPT}(\mathcal{I}_2) \leqslant \rho_2\,\mathrm{LP}(\mathcal{I}_2) \qquad \text{and} \qquad \mathsf{OPT}(\mathcal{I}_1) = \mu - \alpha(\mathcal{I}_1)\,\mathsf{OPT}(\mathcal{I}_2)$$

we have:

$$\begin{aligned} \mathrm{LP}(\mathcal{I}_1) &= \max\{\mu - \alpha(\mathcal{I}_1) f_{\mathcal{I}_2}(x) \mid Ax \geqslant b\} \\ &= \mu - \alpha(\mathcal{I}_1)\min\{f_{\mathcal{I}_2}(x) \mid Ax \geqslant b\} \\ &= \mu - \alpha(\mathcal{I}_1)\,\mathrm{LP}(\mathcal{I}_2) \\ &\leqslant \mu - \frac{1}{\rho_2} \cdot \alpha(\mathcal{I}_1) \cdot \mathsf{OPT}(\mathcal{I}_2) \end{aligned}$$

Proof (continued). Using:

$$\mathsf{OPT}(\mathcal{I}_2) \leqslant \rho_2 \, \mathrm{LP}(\mathcal{I}_2) \qquad \text{and} \qquad \mathsf{OPT}(\mathcal{I}_1) = \mu - \alpha(\mathcal{I}_1) \, \mathsf{OPT}(\mathcal{I}_2)$$

we have:

$$\begin{aligned}
\mathrm{LP}(\mathcal{I}_1) &= \max\{\mu - \alpha(\mathcal{I}_1) f_{\mathcal{I}_2}(x) \mid Ax \geqslant b\} \\
&= \mu - \alpha(\mathcal{I}_1) \min\{f_{\mathcal{I}_2}(x) \mid Ax \geqslant b\} \\
&= \mu - \alpha(\mathcal{I}_1) \, \mathrm{LP}(\mathcal{I}_2) \\
&\leqslant \mu - \frac{1}{\rho_2} \cdot \alpha(\mathcal{I}_1) \cdot \mathsf{OPT}(\mathcal{I}_2) \\
&= \mu + \frac{\mu - c_1}{\mu - s_1} \cdot (\underbrace{\mathsf{OPT}(\mathcal{I}_1)}_{\leqslant s_1} - \mu)
\end{aligned}$$

Proof (continued). Using:

$$\mathsf{OPT}(\mathcal{I}_2) \leqslant \rho_2 \, \mathrm{LP}(\mathcal{I}_2) \qquad \text{and} \qquad \mathsf{OPT}(\mathcal{I}_1) = \mu - \alpha(\mathcal{I}_1) \, \mathsf{OPT}(\mathcal{I}_2)$$

we have:

$$\begin{aligned}
\mathrm{LP}(\mathcal{I}_1) &= \max\{\mu - \alpha(\mathcal{I}_1) f_{\mathcal{I}_2}(x) \mid Ax \geqslant b\} \\
&= \mu - \alpha(\mathcal{I}_1) \min\{f_{\mathcal{I}_2}(x) \mid Ax \geqslant b\} \\
&= \mu - \alpha(\mathcal{I}_1) \, \mathrm{LP}(\mathcal{I}_2) \\
&\leqslant \mu - \frac{1}{\rho_2} \cdot \alpha(\mathcal{I}_1) \cdot \mathsf{OPT}(\mathcal{I}_2) \\
&= \mu + \frac{\mu - c_1}{\mu - s_1} \cdot (\underbrace{\mathsf{OPT}(\mathcal{I}_1)}_{\leqslant s_1} - \mu) \\
&\leqslant \mu + \frac{\mu - c_1}{\mu - s_1} \cdot (s_1 - \mu)
\end{aligned}$$

Proof (continued). Using:

$$\mathsf{OPT}(\mathcal{I}_2) \leqslant \rho_2 \,\mathrm{LP}(\mathcal{I}_2) \qquad \text{and} \qquad \mathsf{OPT}(\mathcal{I}_1) = \mu - \alpha(\mathcal{I}_1)\,\mathsf{OPT}(\mathcal{I}_2)$$

we have:

$$\begin{aligned}
\mathrm{LP}(\mathcal{I}_1) &= \max\{\mu - \alpha(\mathcal{I}_1) f_{\mathcal{I}_2}(x) \mid Ax \geqslant b\} \\
&= \mu - \alpha(\mathcal{I}_1) \min\{f_{\mathcal{I}_2}(x) \mid Ax \geqslant b\} \\
&= \mu - \alpha(\mathcal{I}_1)\,\mathrm{LP}(\mathcal{I}_2) \\
&\leqslant \mu - \frac{1}{\rho_2} \cdot \alpha(\mathcal{I}_1) \cdot \mathsf{OPT}(\mathcal{I}_2) \\
&= \mu + \frac{\mu - c_1}{\mu - s_1} \cdot (\underbrace{\mathsf{OPT}(\mathcal{I}_1)}_{\leqslant s_1} - \mu) \\
&\leqslant \mu + \frac{\mu - c_1}{\mu - s_1} \cdot (s_1 - \mu) \\
&= c_1
\end{aligned}$$