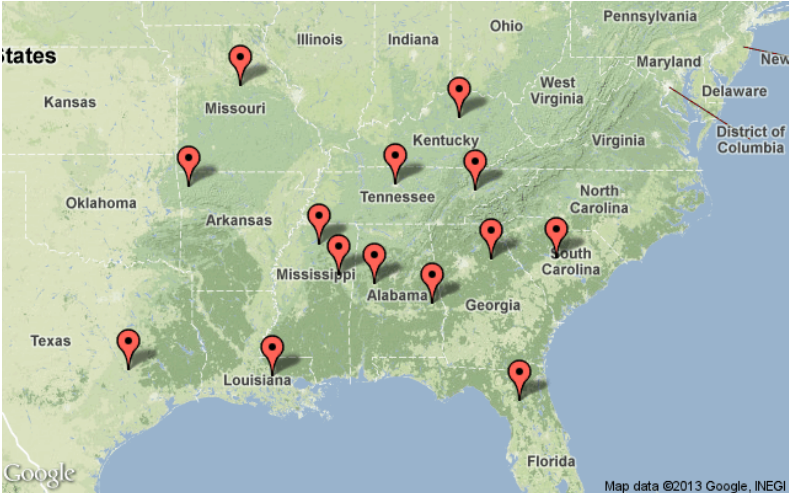
05-03-20 SPP seminar

Extended formulations
and a connection to communication protocols

Manuel Aprile (ULB)

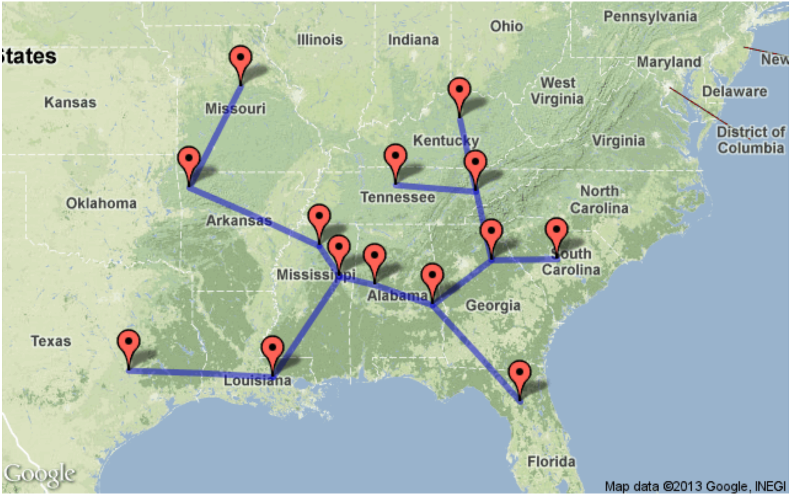
Yuri Faenza (Columbia University)

The minimum weight spanning tree problem



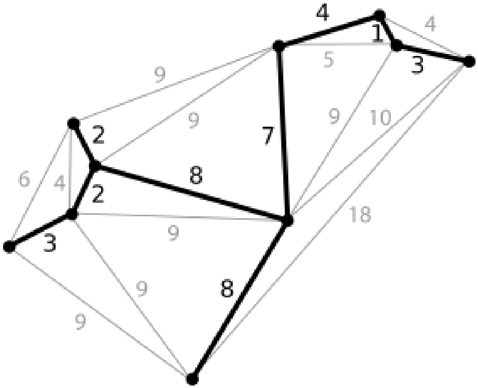
https://pynash.org/2013/03/05/treeification/

The minimum weight spanning tree problem



https://pynash.org/2013/03/05/treeification/

The minimum weight spanning tree problem



Goal: connect all the vertices in the graph G using edges of minimum total weight. The desired subgraph is a **spanning tree** of G.

Combinatorial algorithms

For problems like finding the minimum weight spanning tree, there are fast "ad hoc" algorithms.

However, such algorithms can rarely be adapted when extra constraints arise (e.g. “find the spanning tree of minimum weight such that node A and B are at distance at most 5”).

In practice, we need more flexible algorithms...

Linear programming approach

Idea: associate to each solution of our problem to a point in $\mathbb{R}^d$ and describe the convex hull with a linear system.

$$\text{spanning tree } T \quad \longrightarrow \quad \chi^T_e = \begin{cases} 1 & \text{if } e \in T \\ 0 & \text{otherwise} \end{cases}$$

$$\mathrm{STP}(G) = \mathrm{conv}\{\chi^T : T \text{ is a spanning tree of } G\} = \{x \in \mathbb{R}^E : Ax \leq b\}$$

Then the problem can be formulated as a **linear program** (LP):

$$\begin{array}{ll} \max & \langle c, x \rangle \\ \text{subject to} & Ax \leqslant b, \quad x \in \mathbb{R}^d \end{array}$$

Spanning tree polytope

Graph $G = (V, E)$

$$\mathrm{STP}(G) = \Big\{ x \in \mathbb{R}^E : \\ x(E(U)) \leq |U| - 1 \qquad \forall U \subset V \\ x \geq 0 \\ x(E) = |V| - 1 \Big\}$$

Problem: our description $Ax \leq b$ has exponential size!

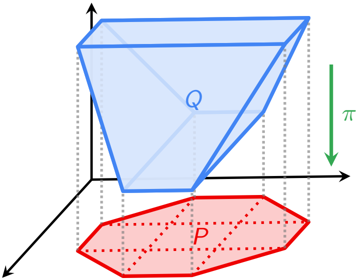
Extended formulations

Let $P = \{x \in \mathbb{R}^d : Ax \leq b\}$ be a polytope.

Definition

$Q = \{A'x + Cy \leq b'\}$ is an **extended formulation** for P if there exists a projection $\pi : \mathbb{R}^{d+k} \to \mathbb{R}^d$ such that $\pi(Q) = P$.

Q has **higher** dimension but **less** facets!

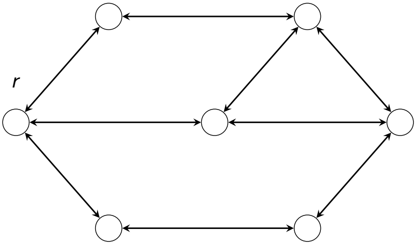


$$
\begin{aligned}
P &= \pi(Q) \\
&= \{x \in \mathbb{R}^d \mid \exists\, y \in \mathbb{R}^k : (x, y) \in Q\} \\
&= \{x \in \mathbb{R}^d \mid \exists\, y \in \mathbb{R}^k : Ax + Cy \leqslant b\}
\end{aligned}
$$

Wong's extended formulation

Graph $G = (V, E)$

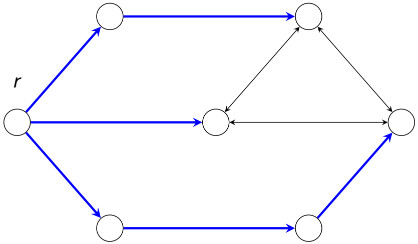
- Bidirect the edges, fix root r
- Spanning trees $=$ r-arborescences
- r-arborescence $=$ union of r-v flow for each $v \in V \setminus \{r\}$.



Wong's extended formulation

Graph $G = (V, E)$

- Bidirect the edges, fix root r
- Spanning trees $=$ r-arborescences
- r-arborescence $=$ union of r-v flow for each $v \in V \setminus \{r\}$.



Wong's extended formulation

$$
\begin{aligned}
\mathrm{STP}(G) = \Big\{ x \in \mathbb{R}^E \mid \exists\, c \in \mathbb{R}^{\vec{E}}, \phi^v \in \mathbb{R}^{\vec{E}} \ \forall\, v \in V - r : \\
& \phi^v(\delta^{\mathrm{out}}(r)) - \phi^v(\delta^{\mathrm{in}}(r)) = 1 && \forall\, v \in V - r \\
& \phi^v(\delta^{\mathrm{out}}(u)) - \phi^v(\delta^{\mathrm{in}}(u)) = 0 && \forall\, u \in V - r - v \\
& \mathbf{0} \le \phi^v \le c && \forall\, v \in V - r \\
& x_{uv} = c_{(u,v)} + c_{(v,u)} && \forall\, uv \in E \\
& x(E) = |V| - 1 && \Big\}
\end{aligned}
$$

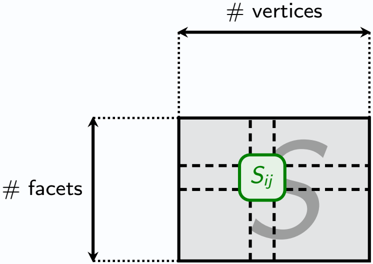
$\mathrm{STP}(G)$ has $\approx 2^{|V|}$ facets...

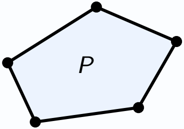
but admits an **extended formulation** of size $O(|V| \cdot |E|)$.

Slack matrix

Consider $P = \mathrm{conv}(\{x^{(1)}, \dots, x^{(m)}\}) = \{x \in \mathbb{R}^d \mid Ax \leqslant b\}$.

Definition



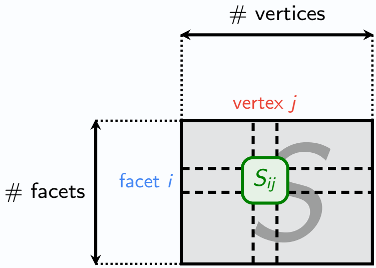


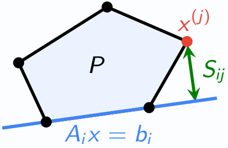
$$\boxed{S_{ij} := b_i - A_i^T x^{(j)}}$$

Slack matrix

Consider $P = \mathrm{conv}(\{x^{(1)}, \ldots, x^{(m)}\}) = \{x \in \mathbb{R}^d \mid Ax \leqslant b\}$.

Definition (Slack matrix of P)





$$\boxed{S_{ij} := b_i - A_i^T x^{(j)}}$$

Yannakakis' Theorem

Theorem (Yannakakis, 1989)

Let S be a slack-matrix of $P = \{Ax \leq b\}$. If $S = TU$, with T, U nonnegative, then

$$\{Ax + Ty = b, y \geq 0\}$$

is an extended formulation of P.

Yannakakis' Theorem

Theorem (Yannakakis, 1989)

Let S be a slack-matrix of $P = \{Ax \leq b\}$. If $S = TU$, with T, U nonnegative, then

$$\{Ax + Ty = b, y \geq 0\}$$

is an extended formulation of P.

Factorization of rank **r** $\rightarrow$ EF with **r** inequalities, BUT...

What about the equations?

Yannakakis' Theorem

Theorem (Yannakakis, 1989)

Let S be a slack-matrix of $P = \{Ax \leq b\}$. If $S = TU$, with T, U nonnegative, then

$$\{Ax + Ty = b, y \geq 0\}$$

is an extended formulation of P.

Factorization of rank **r** $\rightarrow$ EF with **r** inequalities, BUT...

What about the equations?

Most of them are redundant. Hence, there is an EF $\{A'x + T'y = b', y \geq 0\}$ with A', T', b' small ($\leq$ **r** equations)

Problem: how to find it directly (i.e. without writing A, T explicitly) ?

Deterministic communication protocols

$f : X \times Y \to \{0, 1\}$ boolean function (matrix).

Two players:

Alice knows $x \in X$

Bob knows $y \in Y$

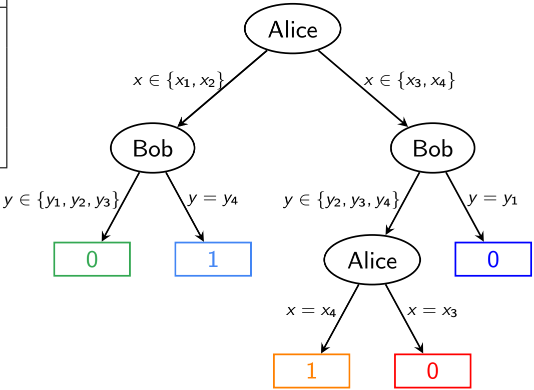
want to compute $f(x, y)$ by exchanging bits.

	y_1	y_2	y_3	y_4
x_1	0	0	0	1
x_2	0	0	0	1
x_3	0	0	0	0
x_4	0	1	1	1

Goal: Minimize # bits exchanged.

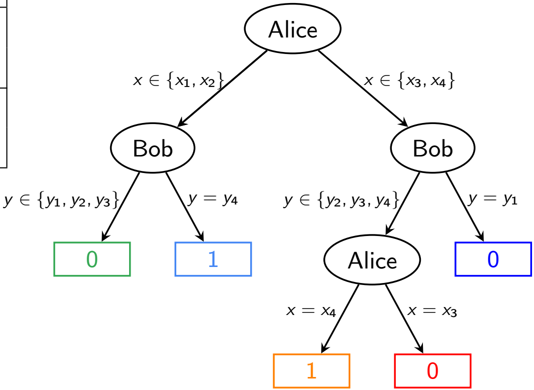
Deterministic communication protocols (cont.)

	y_1	y_2	y_3	y_4
x_1	0	0	0	1
x_2	0	0	0	1
x_3	0	0	0	0
x_4	0	1	1	1



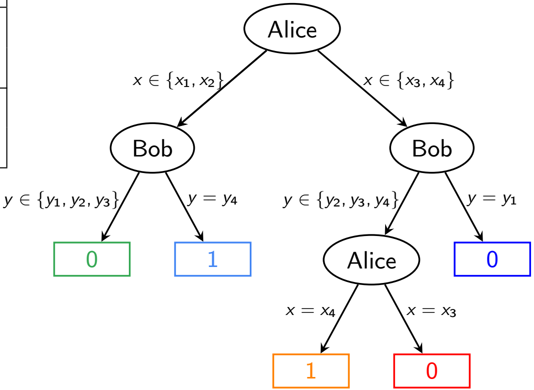
Deterministic communication protocols (cont.)

	y_1	y_2	y_3	y_4
x_1	0	0	0	1
x_2	0	0	0	1
x_3	0	0	0	0
x_4	0	1	1	1



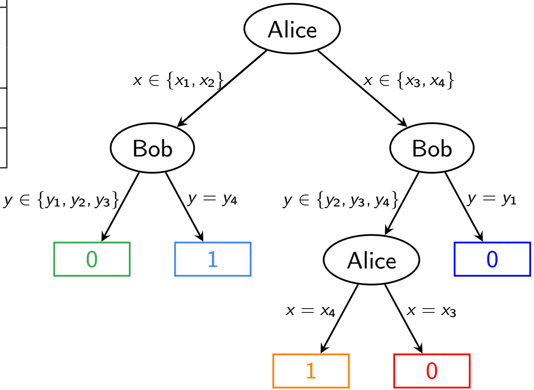
Deterministic communication protocols (cont.)

	y_1	y_2	y_3	y_4
x_1	0	0	0	1
x_2	0	0	0	1
x_3	0	0	0	0
x_4	0	1	1	1



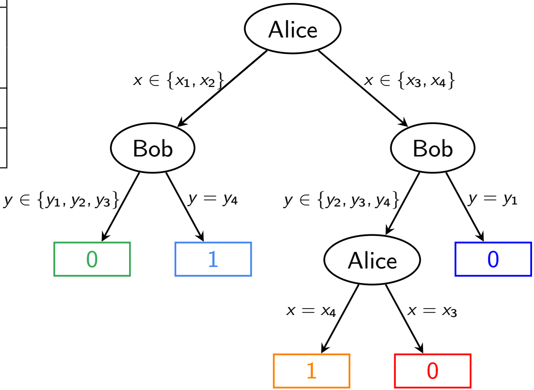
Deterministic communication protocols (cont.)

	y_1	y_2	y_3	y_4
x_1	0	0	0	1
x_2	0	0	0	1
x_3	0	0	0	0
x_4	0	1	1	1



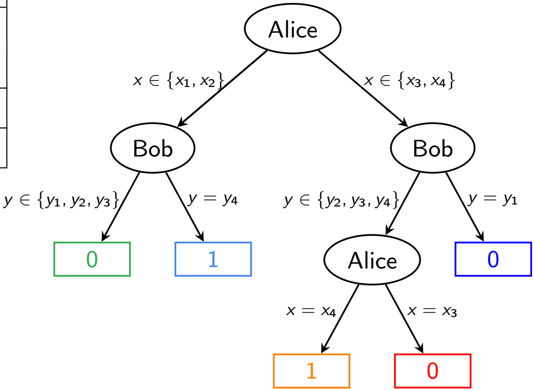
Deterministic communication protocols (cont.)

	y_1	y_2	y_3	y_4
x_1	0	0	0	1
x_2	0	0	0	1
x_3	0	0	0	0
x_4	0	1	1	1



Deterministic communication protocols (cont.)

	y_1	y_2	y_3	y_4
x_1	0	0	0	1
x_2	0	0	0	1
x_3	0	0	0	0
x_4	0	1	1	1



Complexity of the protocol = height of the tree.

$\#$ of rectangles = $\#$ of leaves $\leq 2^{\text{height}}$

EFs and communication complexity

Theorem Yannakakis, 1989

Let P be a polytope, and S its slack matrix. Assume there exists a deterministic protocol of complexity c for S. Then there is an EF of P of size $\leq 2^c$.

Deterministic protocol

$\downarrow$

Factorization of the slack matrix

$\downarrow$

Extended Formulation

EFs and communication complexity

Theorem Yannakakis, 1989

Let P be a polytope, and S its slack matrix. Assume there exists a deterministic protocol of complexity c for S. Then there is an EF of P of size $\leq 2^c$.

$$\begin{array}{c}\text{Deterministic protocol}\\ \downarrow \\ \text{Factorization of the slack matrix} \\ \downarrow \\ \text{Extended Formulation}\end{array}$$

Can we skip the factorization step and directly get our EF?

EFs and communication complexity

Theorem Yannakakis, 1989

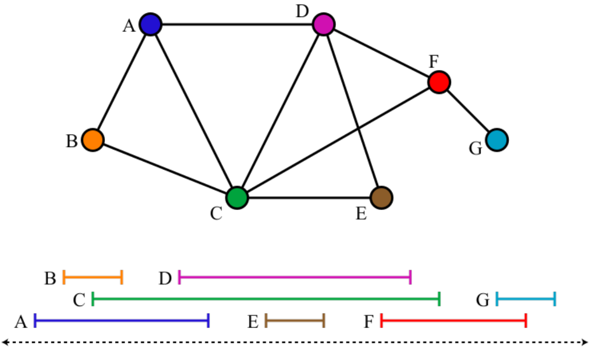
Let P be a polytope, and S its slack matrix. Assume there exists a deterministic protocol of complexity c for S. Then there is an EF of P of size $\leq 2^c$.

$$
\begin{tikzcd}
\text{Deterministic protocol} \arrow[d] \arrow[dd, bend left] \\
\text{Factorization of the slack matrix} \arrow[d] \\
\text{Extended Formulation}
\end{tikzcd}
$$

Can we skip the factorization step and directly get our EF?

Yes! We give an algorithm that, given a protocol and some information, outputs a corresponding EF in linear time.

The maximum stable set problem

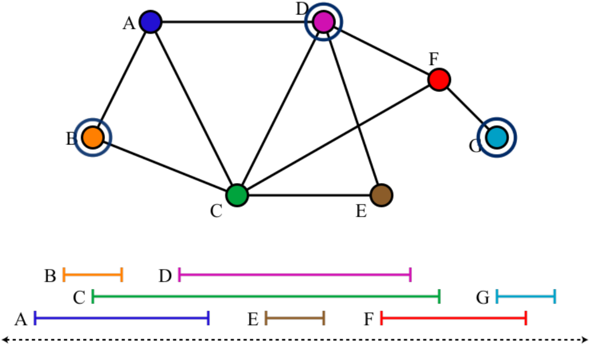


By David Eppstein - https://commons.wikimedia.org/w/index.php?curid=3001223

Goal: find the largest set of jobs that do not interfere with each other.

The desired subgraph is a **stable set** of G (=no edges).

The maximum stable set problem



By David Eppstein - https://commons.wikimedia.org/w/index.php?curid=3001223

Goal: find the largest set of jobs that do not interfere with each other.
The desired subgraph is a **stable set** of G (=no edges).

STAB(G), G perfect

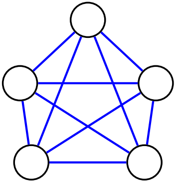
Perfect graph: graph without induced odd cycles or odd anticycles of length ≥ 5.

Let $STAB(G) = \text{conv}\{\chi^S \;:\; S \text{ is a stable set in G}\}$.

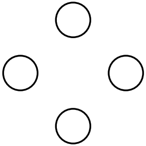
Theorem (Chvàtal, 1974)

$G = (V, E)$ is perfect if and only if

$$\text{STAB}(G) = \{x \in \mathbb{R}^V : \begin{array}{lcl} x & \geq & 0 \\ \sum_{v \in C} x_v & \leq & 1 \text{ for all cliques } C \text{ of } G\} \end{array}$$



Clique



Stable set

STAB(G), G perfect

Perfect graph: graph without induced odd cycles or odd anticycles of length ≥ 5.

Let $STAB(G) = \text{conv}\{\chi^S \;:\; S$ is a stable set in G$\}$.

Theorem (Chvàtal, 1974)

$G = (V, E)$ is perfect if and only if

$$\text{STAB}(G) = \{x \in \mathbb{R}^V : \begin{array}{lll} x & \geq & 0 \\ \sum_{v \in C} x_v & \leq & 1 \text{ for all cliques } C \text{ of } G\} \end{array}$$

Exponential number of inequalities $\Rightarrow$ no use for a polytime algorithm.

Open question

Is there a polynomial size extended formulation (EF) for STAB(G)?

;

STAB(G), G perfect

Theorem (Yannakakis, 1989)

Let G be a perfect graph on n vertices. There is a deterministic protocol for the slack matrix of STAB(G) of complexity $O(\log^2 n)$, hence an EF of size $n^{O(\log n)}$.

But as seen before, writing down the corresponding EF takes exponential time.

STAB(G), G perfect

Theorem (Yannakakis, 1989)

Let G be a perfect graph on n vertices. There is a deterministic protocol for the slack matrix of STAB(G) of complexity $O(\log^2 n)$, hence an EF of size $n^{O(\log n)}$.

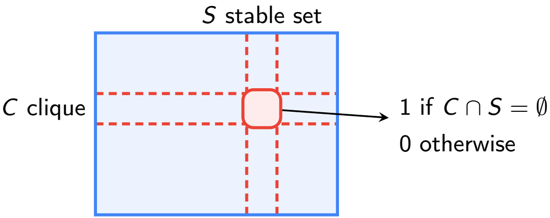
But as seen before, writing down the corresponding EF takes exponential time.

We give an algorithm to write down the EF efficiently (i.e. in time $n^{O(\log n)}$).

Yannakakis' protocol

$$STAB(G) = \{x \in \mathbb{R}^V : \begin{array}{rcl} x & \geq & 0 \\ \sum_{v \in C} x_v & \leq & 1 \text{ for all cliques } C \text{ of } G \end{array}\}$$

Slack matrix (ignoring nonnegativity inequalities):



Alice gets a clique C, Bob gets a stable set S

Goal: decide whether $C \cap S = \emptyset$

Yannakakis' protocol

Alice: if $\exists\, v \in C$ of degree $\leq \frac{n}{2}$, send v, else send 0

Bob: If $v \in S$, then $S \cap C \neq \emptyset \rightarrow$ output 0

Else, restrict G (and S) to $N(v)$ Since $C \subseteq N(v)$

Bob: If $\exists\, u \in S$ of degree $> \frac{n}{2}$, send u, else send 0

Repeat...

Note: If both Alice and Bob send 0, then $C \cap S = \emptyset \rightarrow$ output 1

The graph shrinks by half at every stage, so Alice and Bob communicate $O(\log n)$ vertices. $\implies$ at most $O(\log^2 n)$ bits exchanged!

Yannakakis' protocol

Alice: if $\exists\, v \in C$ of degree $\leq \frac{n}{2}$, send v, else send 0

Bob: If $v \in S$, then $S \cap C \neq \emptyset \rightarrow$ output 0

Else, restrict G (and S) to $N(v)$ Since $C \subseteq N(v)$

Bob: If $\exists\, u \in S$ of degree $> \frac{n}{2}$, send u, else send 0

Repeat...

Note: If both Alice and Bob send 0, then $C \cap S = \emptyset \rightarrow$ output 1

This protocol partitions the slack matrix in $2^{O(\log^2 n)} = n^{O(\log n)}$ rectangles, giving a factorization of the same size $\implies$

$$\text{STAB}(G) = \{x : \exists\, y \in \mathbb{R}^{n^{O(\log n)}} : Ax + Ty = b, y \geq 0\}$$

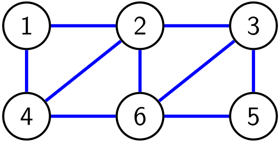
But T has a complex structure, we could not get rid of redundant equations for general G.

Our result: first ingredient

Lemma

Let $G(V, E)$ be a perfect graph, and let $v_1, \ldots, v_k \in V$. Let G_i be the subgraph of G induced by v_i and its neighbors, and G_0 be the subgraph of G induced by $V \setminus \{v_1, \ldots, v_k\}$. Then we have

$$\mathrm{STAB}(G) = \mathrm{STAB}(G_0) \cap \cdots \cap \mathrm{STAB}(G_k).$$



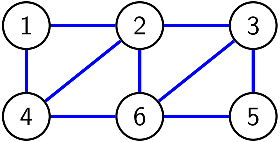
G

Our result: first ingredient

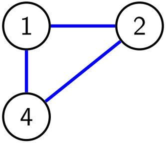
Lemma

Let $G(V, E)$ be a perfect graph, and let $v_1, \dots, v_k \in V$. Let G_i be the subgraph of G induced by v_i and its neighbors, and G_0 be the subgraph of G induced by $V \setminus \{v_1, \dots, v_k\}$. Then we have

$$\mathrm{STAB}(G) = \mathrm{STAB}(G_0) \cap \cdots \cap \mathrm{STAB}(G_k).$$



G



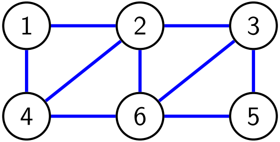
G_1

Our result: first ingredient

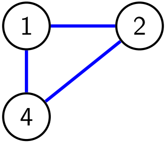
Lemma

Let $G(V, E)$ be a perfect graph, and let $v_1, \dots, v_k \in V$. Let G_i be the subgraph of G induced by v_i and its neighbors, and G_0 be the subgraph of G induced by $V \setminus \{v_1, \dots, v_k\}$. Then we have

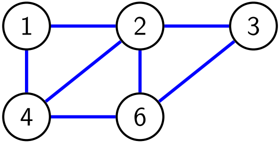
$$\mathrm{STAB}(G) = \mathrm{STAB}(G_0) \cap \cdots \cap \mathrm{STAB}(G_k).$$



G



G_1



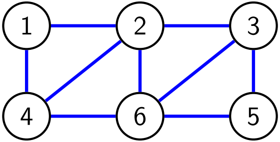
G_2

Our result: first ingredient

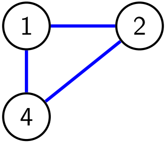
Lemma

Let $G(V, E)$ be a perfect graph, and let $v_1, \dots, v_k \in V$. Let G_i be the subgraph of G induced by v_i and its neighbors, and G_0 be the subgraph of G induced by $V \setminus \{v_1, \dots, v_k\}$. Then we have

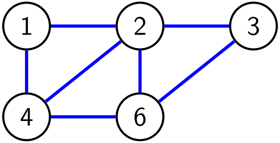
$$\mathrm{STAB}(G) = \mathrm{STAB}(G_0) \cap \cdots \cap \mathrm{STAB}(G_k).$$



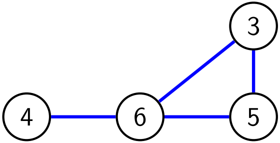
G



G_1



G_2



G_0

Our result: second ingredient

Lemma (Fulkerson 1972)

G is a perfect graph if and only if

$$\mathrm{STAB}(G) = \{x : x \geq 0, x^T y \leq 1 \; \forall \; y \in \mathrm{STAB}(\bar{G})\}.$$

Our result: second ingredient

Lemma (Fulkerson 1972)

G is a perfect graph if and only if

$$\mathrm{STAB}(G) = \{x : x \geq 0, x^T y \leq 1 \ \forall \ y \in \mathrm{STAB}(\bar{G})\}.$$

Lemma (Martin 1991, Weltge 2015)

Given a non-empty polyhedron Q, let $P = \{x : x^T y \leq 1 \ \forall \ y \in Q\}$. Given an EF for Q, we can efficiently get an EF for P (of roughly the same size).

Our result: second ingredient

Lemma (Fulkerson 1972)

G is a perfect graph if and only if

$$\mathrm{STAB}(G) = \{x : x \geq 0, x^T y \leq 1 \ \forall\, y \in \mathrm{STAB}(\bar{G})\}.$$

Lemma (Martin 1991, Weltge 2015)

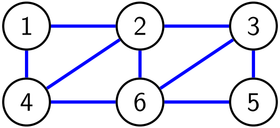
Given a non-empty polyhedron Q, let $P = \{x : x^T y \leq 1 \ \forall\, y \in Q\}$. If $Q = \{y : \exists\, z : Ay + Bz \leq b, Cy + Dz = d\}$, then

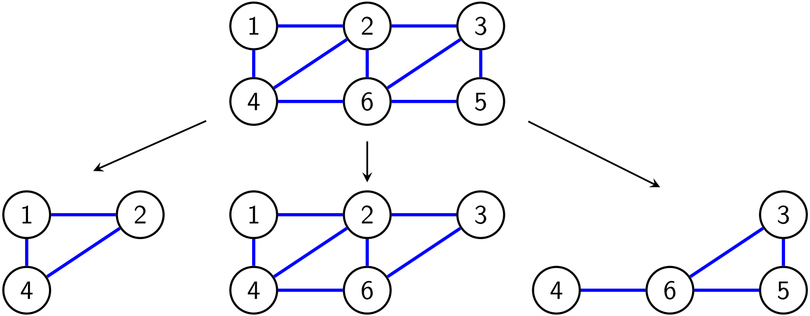
$$P = \{x : \exists \lambda \geq 0, \mu : A^T \lambda + C^T \mu = x,$$

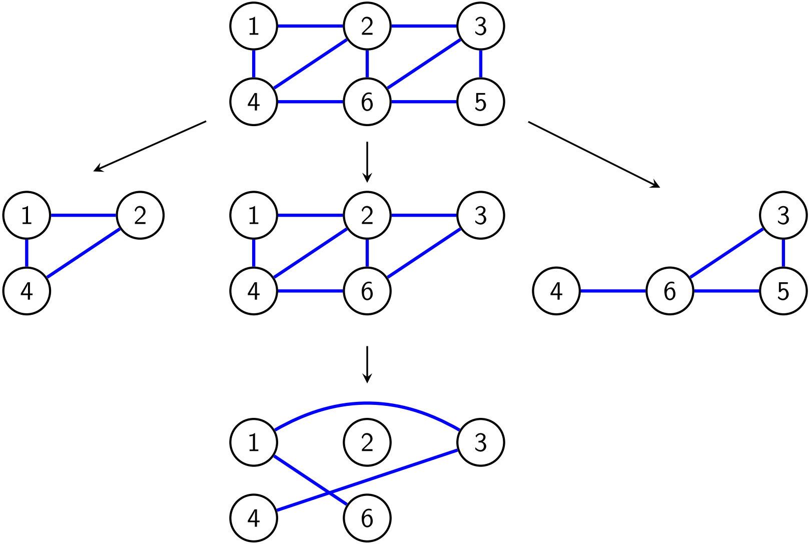
$$B^T \lambda + D^T \mu = 0, b^T \lambda + d^T \mu \leq 1\}.$$

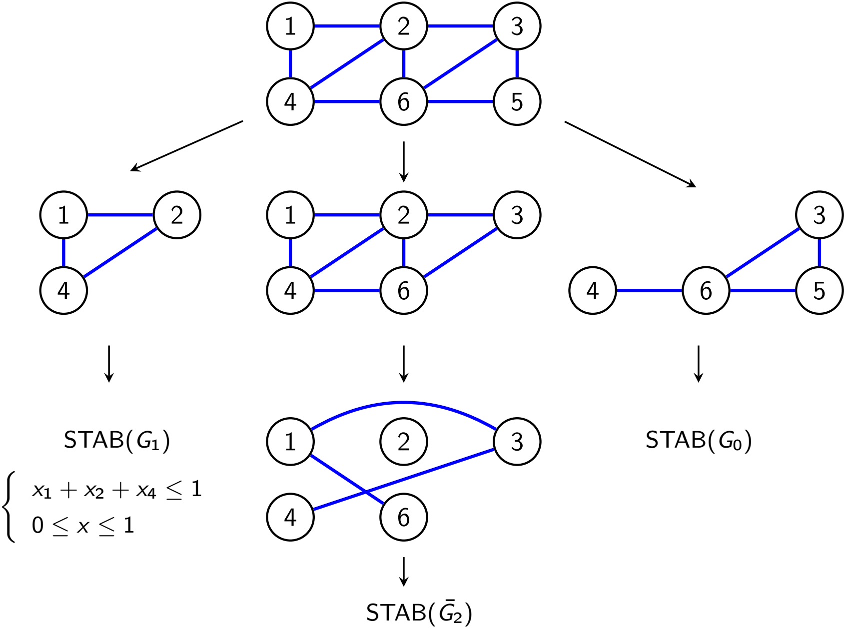
Corollary

Given an EF for $\mathrm{STAB}(G)$, G perfect, we can efficiently obtain an extended formulation for $\mathrm{STAB}(\bar{G})$ (of roughly the same size).


1
2
3
4
6
5




1
2
3
4
6
5
1
2
4
1
2
3
4
6
3
4
6
5
1
2
3
4
6


1
2
3
4
6
5
1
2
4
1
2
3
4
6
3
4
6
5
STAB(G_1)
$\begin{cases} x_1 + x_2 + x_4 \leq 1 \\ 0 \leq x \leq 1 \end{cases}$
1
2
3
4
6
STAB($\bar{G}_2$)
STAB(G_0)

Main algorithm

Input: G on n nodes

Let $v_1, \dots, v_k \in V$ be the nodes with degree $\leq \frac{n}{2}$

if $k \geq \frac{n}{2}$ **then**

Recurse on $G_1, \dots, G_k, G_0$ $\qquad G_i = G[N^+(v_i)]$

$G_0 = G[V \setminus v_1, \dots, v_k]$

else

Repeat with the complement $\bar{G}$

Main algorithm

Input: G on n nodes

Let $v_1, \ldots, v_k \in V$ be the nodes with degree $\leq \frac{n}{2}$

if $k \geq \frac{n}{2}$ **then**

Recurse on $G_1, \ldots, G_k, G_0$ $\quad$ $G_i = G[N^+(v_i)]$

$G_0 = G[V \setminus v_1, \ldots, v_k]$

else

Repeat with the complement $\bar{G}$

...Until our graphs have constant size. Then reconstruct the formulation for STAB(G) using the previous Lemmas.

Main algorithm

Input: G on n nodes

Let $v_1, \dots, v_k \in V$ be the nodes with degree $\leq \frac{n}{2}$

if $k \geq \frac{n}{2}$ **then**

Recurse on $G_1, \dots, G_k, G_0$ $\qquad$ $G_i = G[N^+(v_i)]$

$G_0 = G[V \setminus v_1, \dots, v_k]$

else

Repeat with the complement $\bar{G}$

...Until our graphs have constant size. Then reconstruct the formulation for STAB(G) using the previous Lemmas.

- If G has $< \frac{n}{2}$ 'low degree' nodes, then $\bar{G}$ has $\geq \frac{n}{2}$ 'low degree' nodes.

Main algorithm

Input: G on n nodes

Let $v_1, \ldots, v_k \in V$ be the nodes with degree $\leq \frac{n}{2}$

if $k \geq \frac{n}{2}$ **then**

Recurse on $G_1, \ldots, G_k, G_0$ $\qquad$ $G_i = G[N^+(v_i)]$

$G_0 = G[V \setminus v_1, \ldots, v_k]$

else

Repeat with the complement $\bar{G}$

...Until our graphs have constant size. Then reconstruct the formulation for STAB(G) using the previous Lemmas.

- If G has $< \frac{n}{2}$ 'low degree' nodes, then $\bar{G}$ has $\geq \frac{n}{2}$ 'low degree' nodes.
- We recurse on at most n graphs of size at most $n/2 \implies n^{O(\log n)}$ total running time.

A general result

Theorem

Assume that there is a deterministic protocol, described by a tree τ, that partitions the slack matrix of P into rectangles $\mathcal{R} = \{R_1, \dots, R_k\}$. Then there is an algorithm that, given τ and a representation of $\mathcal{R}$, outputs an extended formulation of P in linear time in the size of the input.

A general result

Theorem

Assume that there is a deterministic protocol, described by a tree τ, that partitions the slack matrix of P into rectangles $\mathcal{R} = \{R_1, \ldots, R_k\}$. Then there is an algorithm that, given τ and a representation of $\mathcal{R}$, outputs an extended formulation of P in linear time in the size of the input.

- Our method is flexible: we can start from approximate EFs at the bottom and get an approximate EF of P.
- In particular, our method yields a relaxation of $\mathrm{STAB}(G)$ for non-perfect graphs G.

Conclusion

We give an algorithm to turn deterministic protocols into extended formulations in output-efficient time.

In particular we give an output-efficient algorithm to construct a quasipolynomial size EF for STAB(G), G perfect.

Open question

Can we extend this to randomized protocols?

(See Faenza, Fiorini, Grappe, Tiwary 2015)

Open question

Is there a polynomial size extended formulation for STAB(G), G perfect?

Conclusion

We give an algorithm to turn deterministic protocols into extended formulations in output-efficient time.

In particular we give an output-efficient algorithm to construct a quasipolynomial size EF for STAB(G), G perfect.

Open question

Can we extend this to randomized protocols?

(See Faenza, Fiorini, Grappe, Tiwary 2015)

Open question

Is there a polynomial size extended formulation for STAB(G), G perfect?

Thank you for your attention!