Can linear programs solve NP-hard problems?

Ronald de Wolf

Centrum Wiskunde & Informatica

Universiteit van Amsterdam

Linear programs

Linear programs

- A factory produces 2 types of goods

Linear programs

- A factory produces 2 types of goods
  - unit of type 1 gives profit € 1; type 2 gives profit € 4

Linear programs

- A factory produces 2 types of goods
  - unit of type 1 gives profit € 1; type 2 gives profit € 4

  It has 4 kinds of resources: 650 of R_1, 100 of R_2,...

Linear programs

- A factory produces 2 types of goods
  - unit of type 1 gives profit € 1; type 2 gives profit € 4

  It has 4 kinds of resources: 650 of R_1, 100 of R_2,...
  - type 1 uses 34 units of R_1, type 2 uses 16 units of R_1

Linear programs

- A factory produces 2 types of goods
  - unit of type 1 gives profit € 1; type 2 gives profit € 4

  It has 4 kinds of resources: 650 of R_1, 100 of R_2,. . .
  - type 1 uses 34 units of R_1, type 2 uses 16 units of R_1
- Maximizing profit is linear program

Linear programs

- A factory produces 2 types of goods
  - unit of type 1 gives profit € 1; type 2 gives profit € 4

  It has 4 kinds of resources: 650 of R_1, 100 of R_2,. . .
  - type 1 uses 34 units of R_1, type 2 uses 16 units of R_1
- Maximizing profit is linear program:

$$\begin{array}{ll} \max & x_1 + 4x_2 \\ \text{s.t.} & 34x_1 + 16x_2 \le 650 \\ & \dots \qquad\qquad \le 100 \\ & \dots \\ & x_1, x_2 \ge 0 \end{array}$$

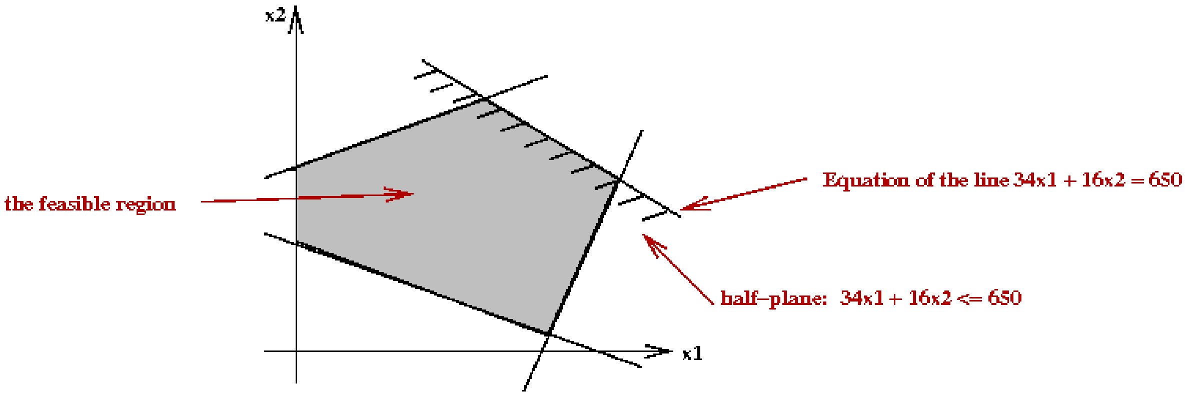
Linear programs

- A factory produces 2 types of goods
  - unit of type 1 gives profit € 1; type 2 gives profit € 4

  It has 4 kinds of resources: 650 of R_1, 100 of R_2,...
  - type 1 uses 34 units of R_1, type 2 uses 16 units of R_1
- Maximizing profit is linear program:

$$\begin{array}{ll} \max & x_1 + 4x_2 \\ \text{s.t.} & 34x_1 + 16x_2 \leq 650 \\ & \ldots \qquad\qquad \leq 100 \\ & \ldots \\ & x_1, x_2 \geq 0 \end{array}$$

- Feasible region is a polytope



A bit of history of LPs

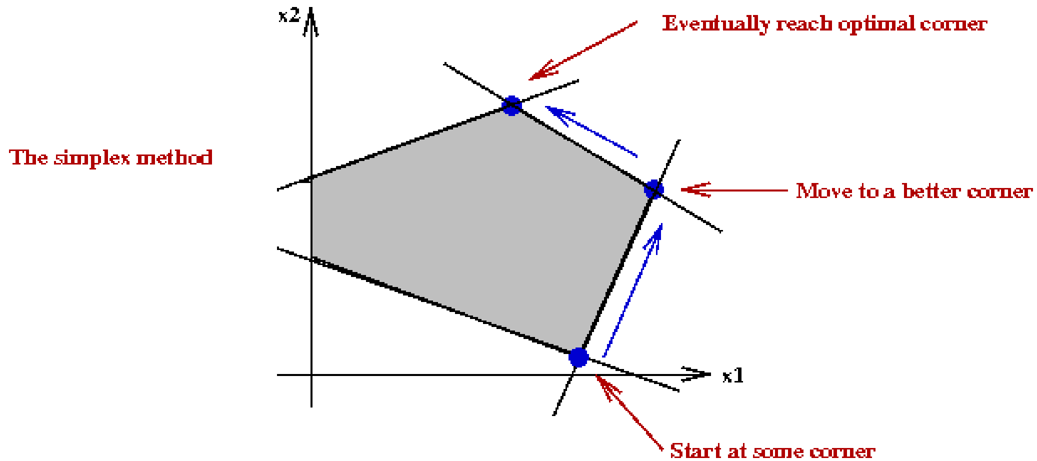
A bit of history of LPs

- Simplex algorithm: developed by Kantorovich and Dantzig for use in WWII, published by Dantzig in 1947.

A bit of history of LPs

- Simplex algorithm: developed by Kantorovich and Dantzig for use in WWII, published by Dantzig in 1947.

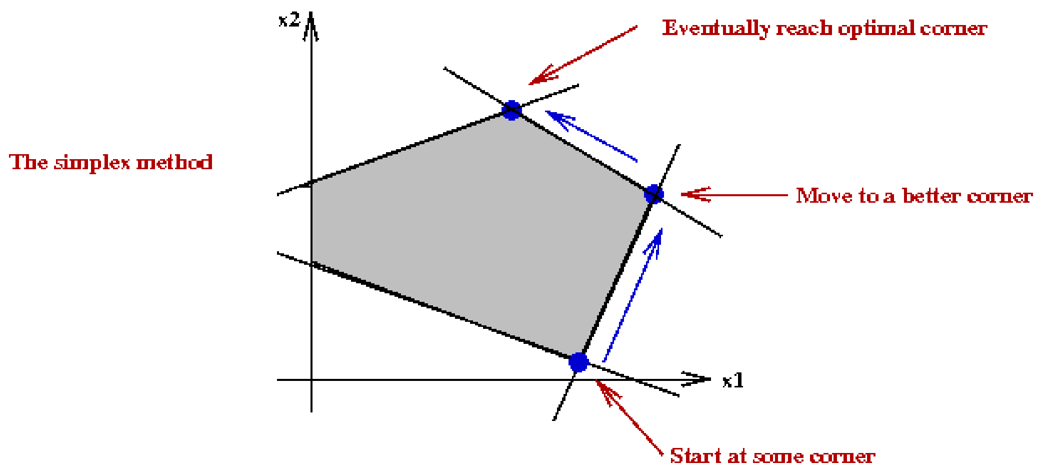
  Walks along vertices of polytope, optimizing objective



A bit of history of LPs

- Simplex algorithm: developed by Kantorovich and Dantzig for use in WWII, published by Dantzig in 1947.

  Walks along vertices of polytope, optimizing objective

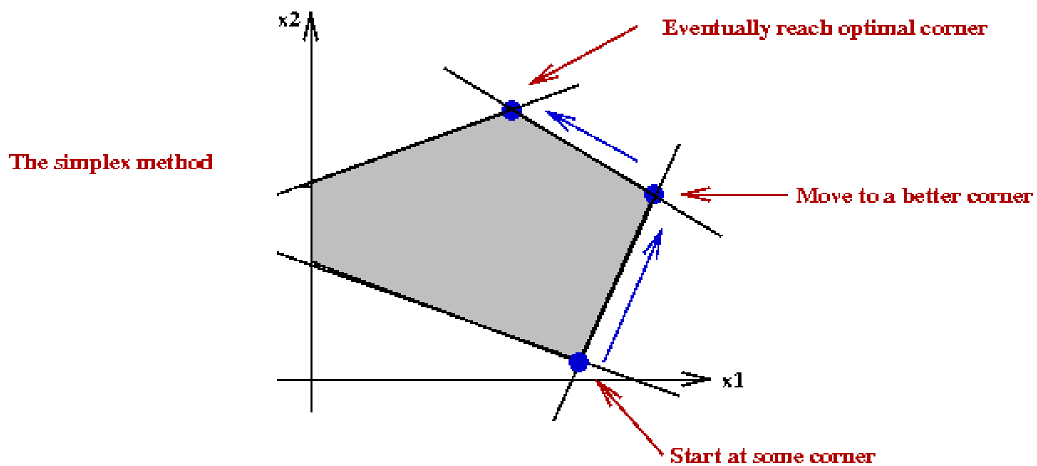


  Usually efficient in practice, worst-case exponential time

A bit of history of LPs

- Simplex algorithm: developed by Kantorovich and Dantzig for use in WWII, published by Dantzig in 1947.

  Walks along vertices of polytope, optimizing objective



  Usually efficient in practice, worst-case exponential time

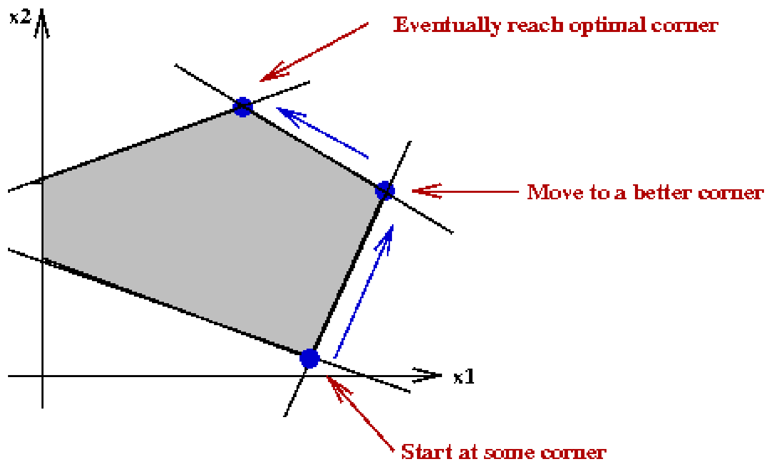
- Ellipsoid method (Khachiyan'79): takes polynomial time in the worst case, but is not practical

A bit of history of LPs

- Simplex algorithm: developed by Kantorovich and Dantzig for use in WWII, published by Dantzig in 1947.

  Walks along vertices of polytope, optimizing objective

  The simplex method

  Usually efficient in practice, worst-case exponential time
- Ellipsoid method (Khachiyan'79): takes polynomial time in the worst case, but is not practical
- Interior point method (Karmarkar'84): reasonably efficient in practice

Can we solve NP-hard problems by LP?

Can we solve NP-hard problems by LP?

- Famous **NP**-hard problem: traveling salesman problem
  Find shortest cycle that goes through each of n cities exactly once

Can we solve NP-hard problems by LP?

- Famous **NP**-hard problem: traveling salesman problem
  Find shortest cycle that goes through each of n cities exactly once
- A polynomial-size LP for TSP would show **P** $=$ **NP**

Can we solve NP-hard problems by LP?

- Famous **NP**-hard problem: traveling salesman problem
  Find shortest cycle that goes through each of n cities exactly once
- A polynomial-size LP for TSP would show **P** $=$ **NP**
- Swart'86–87 claimed to have found such LPs

Can we solve NP-hard problems by LP?

- Famous **NP**-hard problem: traveling salesman problem
  Find shortest cycle that goes through each of n cities exactly once
- A polynomial-size LP for TSP would show **P** $=$ **NP**
- Swart'86–87 claimed to have found such LPs
- Yannakakis'88: symmetric LPs for TSP are exponential

Can we solve NP-hard problems by LP?

- Famous **NP**-hard problem: traveling salesman problem
  Find shortest cycle that goes through each of n cities exactly once
- A polynomial-size LP for TSP would show **P** $=$ **NP**
- Swart'86–87 claimed to have found such LPs
- Yannakakis'88: symmetric LPs for TSP are exponential
- Swart's LPs were symmetric, so they couldn't work

Can we solve NP-hard problems by LP?

- Famous **NP**-hard problem: traveling salesman problem
  Find shortest cycle that goes through each of n cities exactly once
- A polynomial-size LP for TSP would show **P** $=$ **NP**
- Swart'86–87 claimed to have found such LPs
- Yannakakis'88: symmetric LPs for TSP are exponential
- Swart's LPs were symmetric, so they couldn't work
- 20-year open problem: what about non-symmetric LP?

Can we solve NP-hard problems by LP?

- Famous **NP**-hard problem: traveling salesman problem
  Find shortest cycle that goes through each of n cities exactly once
- A polynomial-size LP for TSP would show **P** $=$ **NP**
- Swart'86–87 claimed to have found such LPs
- Yannakakis'88: symmetric LPs for TSP are exponential
- Swart's LPs were symmetric, so they couldn't work
- 20-year open problem: what about non-symmetric LP?
- Sometimes non-symmetry helps a lot! (Kaibel et al'10)

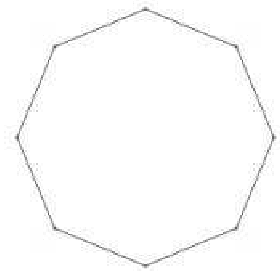
Can we solve NP-hard problems by LP?

- Famous **NP**-hard problem: traveling salesman problem
  Find shortest cycle that goes through each of n cities exactly once
- A polynomial-size LP for TSP would show **P** $=$ **NP**
- Swart'86–87 claimed to have found such LPs
- Yannakakis'88: symmetric LPs for TSP are exponential
- Swart's LPs were symmetric, so they couldn't work
- 20-year open problem: what about non-symmetric LP?
- Sometimes non-symmetry helps a lot! (Kaibel et al'10)
- Fiorini, Massar, Pokutta, Tiwary, dW (STOC'12):
  any LP for TSP needs exponential size

Polytopes and optimization problems

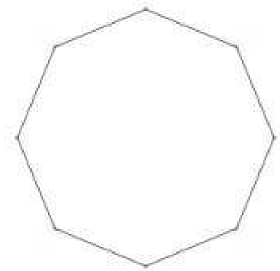
Polytopes and optimization problems

- Polytope P: convex hull of finite set of points in $\mathbb{R}^d$



Polytopes and optimization problems

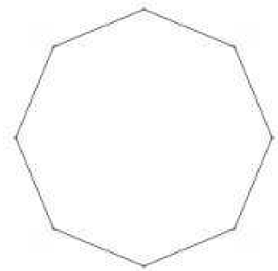
- Polytope P: convex hull of finite set of points in $\mathbb{R}^d$



$\Leftrightarrow$ bounded intersection of finitely many halfspaces

Polytopes and optimization problems

- Polytope P: convex hull of finite set of points in $\mathbb{R}^d$

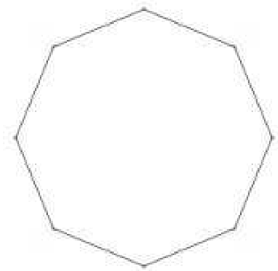


  $\Leftrightarrow$ bounded intersection of finitely many halfspaces

- Can be written as system of linear inequalities:
  $P = \{x \in \mathbb{R}^d \mid Ax \leq b\}$

Polytopes and optimization problems

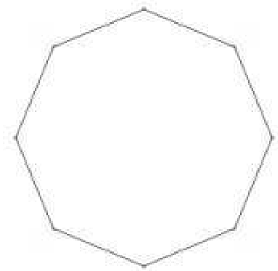
- Polytope P: convex hull of finite set of points in $\mathbb{R}^d$



  $\Leftrightarrow$ bounded intersection of finitely many halfspaces
- Can be written as system of linear inequalities:
  $P = \{x \in \mathbb{R}^d \mid Ax \leq b\}$
  Different systems "$Ax \leq b$" can define the same P

Polytopes and optimization problems

- Polytope P: convex hull of finite set of points in $\mathbb{R}^d$

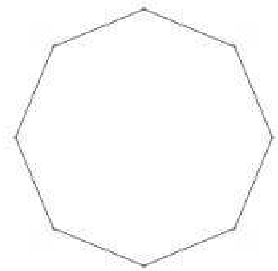


  $\Leftrightarrow$ bounded intersection of finitely many halfspaces

- Can be written as system of linear inequalities:
  $P = \{x \in \mathbb{R}^d \mid Ax \leq b\}$
  Different systems "$Ax \leq b$" can define the same P
  The size of P is the minimal number of inequalities

Polytopes and optimization problems

- Polytope P: convex hull of finite set of points in $\mathbb{R}^d$

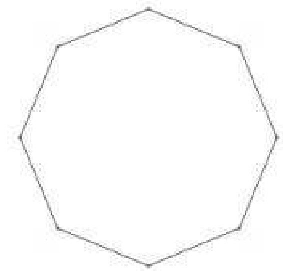


  $\Leftrightarrow$ bounded intersection of finitely many halfspaces

- Can be written as system of linear inequalities:
  $P = \{x \in \mathbb{R}^d \mid Ax \leq b\}$
  Different systems "$Ax \leq b$" can define the same P
  The size of P is the minimal number of inequalities

- TSP polytope: each cycle in K_n is vector $\in \{0,1\}^{\binom{n}{2}}$

Polytopes and optimization problems

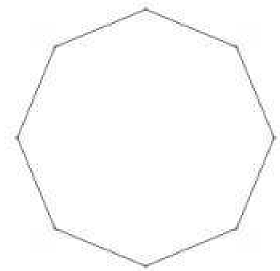


- Polytope P: convex hull of finite set of points in $\mathbb{R}^d$

  $\Leftrightarrow$ bounded intersection of finitely many halfspaces
- Can be written as system of linear inequalities:
  $P = \{x \in \mathbb{R}^d \mid Ax \leq b\}$
  Different systems "$Ax \leq b$" can define the same P
  The size of P is the minimal number of inequalities
- TSP polytope: each cycle in K_n is vector $\in \{0,1\}^{\binom{n}{2}}$.
  TSP(n) is the convex hull of all those vectors

Polytopes and optimization problems

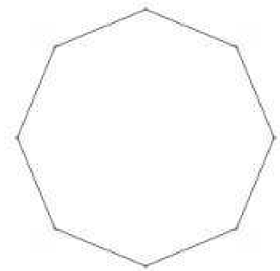
- Polytope P: convex hull of finite set of points in $\mathbb{R}^d$



  $\Leftrightarrow$ bounded intersection of finitely many halfspaces
- Can be written as system of linear inequalities:
  $P = \{x \in \mathbb{R}^d \mid Ax \leq b\}$
  Different systems "$Ax \leq b$" can define the same P
  The size of P is the minimal number of inequalities
- TSP polytope: each cycle in K_n is vector $\in \{0,1\}^{\binom{n}{2}}$.
  TSP(n) is the convex hull of all those vectors
- Solving TSP w.r.t. weight function w_{ij}:
  minimize the linear function $\sum_{i,j} w_{ij}x_{ij}$ over $x \in$ TSP(n)

Polytopes and optimization problems

- Polytope P: convex hull of finite set of points in $\mathbb{R}^d$



  $\Leftrightarrow$ bounded intersection of finitely many halfspaces
- Can be written as system of linear inequalities:
  $P = \{x \in \mathbb{R}^d \mid Ax \leq b\}$
  Different systems "$Ax \leq b$" can define the same P
  The size of P is the minimal number of inequalities
- TSP polytope: each cycle in K_n is vector $\in \{0,1\}^{\binom{n}{2}}$.
  TSP(n) is the convex hull of all those vectors
- Solving TSP w.r.t. weight function w_{ij}:
  minimize the linear function $\sum_{i,j} w_{ij} x_{ij}$ over $x \in$ TSP(n)
- TSP(n) has exponential size

Extended formulations of polytopes

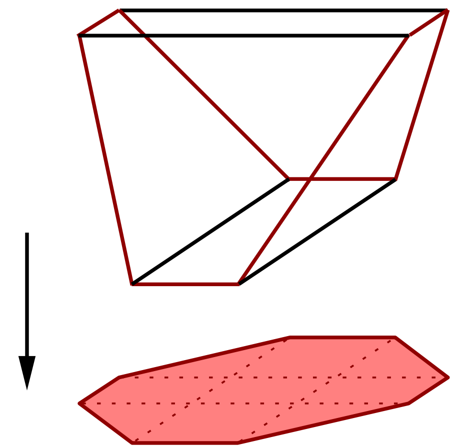
Extended formulations of polytopes

- Sometimes extra variables/dimensions can reduce size very much.

Extended formulations of polytopes

- Sometimes extra variables/dimensions can reduce size very much.

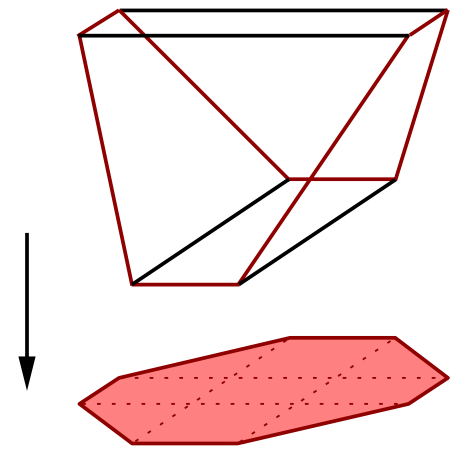
  Regular n-gon in $\mathbb{R}^2$ has size n, but is the projection of polytope Q in higher dimension, of size $O(\log n)$



Extended formulations of polytopes

- Sometimes extra variables/dimensions can reduce size very much.

  Regular n-gon in $\mathbb{R}^2$ has size n, but is the projection of polytope Q in higher dimension, of size $O(\log n)$

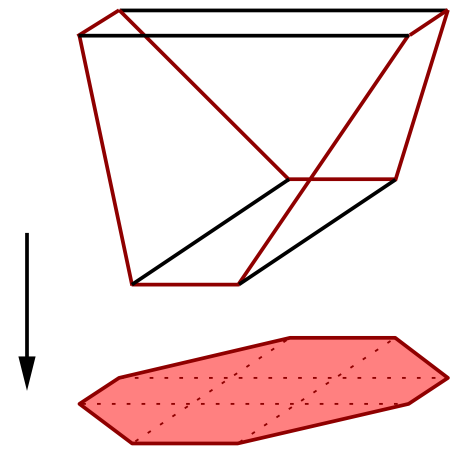


- Optimizing over P reduces to optimizing over Q. If Q has small size, this can be done efficiently!

Extended formulations of polytopes

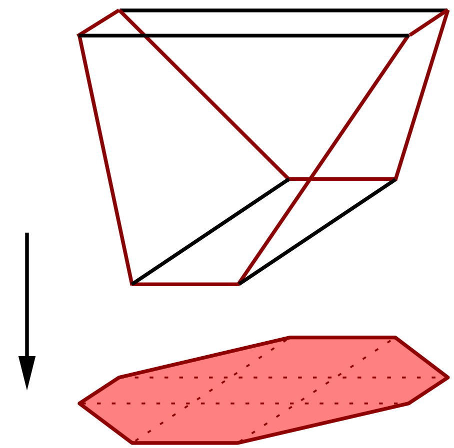
- Sometimes extra variables/dimensions can reduce size very much.

  Regular n-gon in $\mathbb{R}^2$ has size n, but is the projection of polytope Q in higher dimension, of size $O(\log n)$



- Optimizing over P reduces to optimizing over Q. If Q has small size, this can be done efficiently!
- How small can size(Q) be?

Extended formulations of polytopes



- Sometimes extra variables/dimensions can reduce size very much.

  Regular n-gon in $\mathbb{R}^2$ has size n, but is the projection of polytope Q in higher dimension, of size $O(\log n)$

- Optimizing over P reduces to optimizing over Q. If Q has small size, this can be done efficiently!
- How small can size(Q) be?

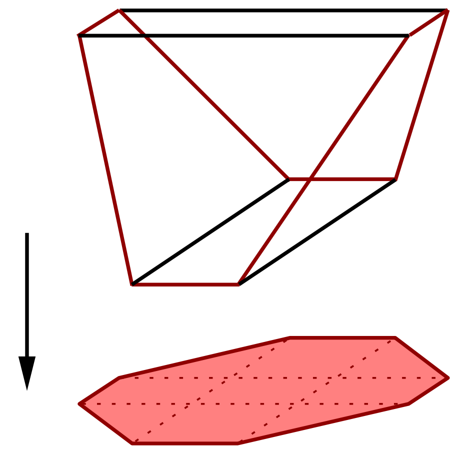
  Extension complexity:

  $xc(P) = \min\{\text{size}(Q) \mid Q \text{ projects to } P\}$

Extended formulations of polytopes

- Sometimes extra variables/dimensions can reduce size very much.

  Regular n-gon in $\mathbb{R}^2$ has size n, but is the projection of polytope Q in higher dimension, of size $O(\log n)$



- Optimizing over P reduces to optimizing over Q. If Q has small size, this can be done efficiently!
- How small can size(Q) be?

  Extension complexity:

  $xc(P) = \min\{\text{size}(Q) \mid Q \text{ projects to } P\}$

- Our goal: strong lower bounds on $xc(P)$ for interesting P

How to bound extension complexity?

How to bound extension complexity?

- Slack matrix S of a polytope $P = \text{conv}(V)$ with inequalities $\{A_i x \leq b_i\}$ and points $V = \{v_j\}$:

$$S_{ij} = b_i - A_i v_j$$

How to bound extension complexity?

- Slack matrix S of a polytope $P = \text{conv}(V)$ with inequalities $\{A_i x \leq b_i\}$ and points $V = \{v_j\}$:

$$S_{ij} = b_i - A_i v_j$$

NB: every entry is nonnegative; S is not unique

How to bound extension complexity?

- Slack matrix S of a polytope $P = \text{conv}(V)$ with inequalities $\{A_i x \leq b_i\}$ and points $V = \{v_j\}$:

$$S_{ij} = b_i - A_i v_j$$

NB: every entry is nonnegative; S is not unique

- Yannakakis'88: $xc(P) =$ positive rank of S

How to bound extension complexity?

- Slack matrix S of a polytope $P = \text{conv}(V)$ with inequalities $\{A_i x \leq b_i\}$ and points $V = \{v_j\}$:

$$S_{ij} = b_i - A_i v_j$$

  NB: every entry is nonnegative; S is not unique
- Yannakakis'88: $xc(P) =$ positive rank of S

  Instead of considering all Q, can focus on slack matrix!

How to bound extension complexity?

- Slack matrix S of a polytope $P = \text{conv}(V)$ with inequalities $\{A_i x \le b_i\}$ and points $V = \{v_j\}$:

  $$S_{ij} = b_i - A_i v_j$$

  NB: every entry is nonnegative; S is not unique
- Yannakakis'88: $xc(P) =$ positive rank of S

  Instead of considering all Q, can focus on slack matrix!
- We can use nondeterministic communication complexity to lower bound $\text{rank}_+(S)$

How to bound extension complexity?

- Slack matrix S of a polytope $P = \text{conv}(V)$ with inequalities $\{A_i x \leq b_i\}$ and points $V = \{v_j\}$:

$$S_{ij} = b_i - A_i v_j$$

  NB: every entry is nonnegative; S is not unique
- Yannakakis'88: $xc(P) =$ positive rank of S

  Instead of considering all Q, can focus on slack matrix!
- We can use nondeterministic communication complexity to lower bound $\text{rank}_+(S)$
- Big problem until now: which polytope to analyze, and how to analyze its slack matrix?

Lower bound for correlation polytope

Lower bound for correlation polytope

- Correlation polytope: COR$(n) = $ conv$\{bb^T \mid b \in \{0,1\}^n\}$

Lower bound for correlation polytope

- Correlation polytope: COR$(n) = \text{conv}\{bb^T \mid b \in \{0,1\}^n\}$
- Occurs naturally in NP-hard problems, e.g. MaxClique

Lower bound for correlation polytope

- Correlation polytope: COR$(n) =$ conv$\{bb^T \mid b \in \{0,1\}^n\}$
- Occurs naturally in NP-hard problems, e.g. MaxClique
- We can find 2^n valid constraints (indexed by $a \in \{0,1\}^n$) whose slacks w.r.t. vertex bb^T are $M_{ab} = (1 - a^T b)^2$

Lower bound for correlation polytope

- Correlation polytope: COR$(n) =$ conv$\{bb^T \mid b \in \{0,1\}^n\}$
- Occurs naturally in NP-hard problems, e.g. MaxClique
- We can find 2^n valid constraints (indexed by $a \in \{0,1\}^n$) whose slacks w.r.t. vertex bb^T are $M_{ab} = (1 - a^T b)^2$
- Nondeterministic communication complexity of M was already analyzed in the quantum computing! (dW'00)

Lower bound for correlation polytope

- Correlation polytope: COR$(n) =$ conv$\{bb^T \mid b \in \{0,1\}^n\}$
- Occurs naturally in NP-hard problems, e.g. MaxClique
- We can find 2^n valid constraints (indexed by $a \in \{0,1\}^n$) whose slacks w.r.t. vertex bb^T are $M_{ab} = (1 - a^T b)^2$
- Nondeterministic communication complexity of M was already analyzed in the quantum computing! (dW'00)
- Take slack matrix S for COR(n), with 2^n vertices bb^T for columns, 2^n a-constraints for first 2^n rows, remaining facets for other rows

$$S = \left[\begin{array}{ccc} & \vdots & \\ \cdots & M_{ab} & \cdots \\ & \vdots & \\ \hline & \vdots & \end{array}\right]$$

Lower bound for correlation polytope

- Correlation polytope: COR$(n) =$ conv$\{bb^T \mid b \in \{0,1\}^n\}$
- Occurs naturally in NP-hard problems, e.g. MaxClique
- We can find 2^n valid constraints (indexed by $a \in \{0,1\}^n$) whose slacks w.r.t. vertex bb^T are $M_{ab} = (1 - a^T b)^2$
- Nondeterministic communication complexity of M was already analyzed in the quantum computing! (dW'00)
- Take slack matrix S for COR(n), with 2^n vertices bb^T for columns, 2^n a-constraints for first 2^n rows, remaining facets for other rows

$$S = \left[\begin{array}{ccc} & \vdots & \\ \cdots & M_{ab} & \cdots \\ & \vdots & \\ \hline & \vdots & \end{array}\right]$$

- $xc($COR$(n))$

Lower bound for correlation polytope

- Correlation polytope: COR(n) = conv$\{bb^T \mid b \in \{0,1\}^n\}$
- Occurs naturally in NP-hard problems, e.g. MaxClique
- We can find 2^n valid constraints (indexed by $a \in \{0,1\}^n$) whose slacks w.r.t. vertex bb^T are $M_{ab} = (1 - a^T b)^2$
- Nondeterministic communication complexity of M was already analyzed in the quantum computing! (dW'00)
- Take slack matrix S for COR(n), with 2^n vertices bb^T for columns, 2^n a-constraints for first 2^n rows, remaining facets for other rows

$$S = \left[\begin{array}{ccc} & \vdots & \\ \cdots & M_{ab} & \cdots \\ & \vdots & \\ \hline & \vdots & \end{array}\right]$$

- xc(COR(n)) = rank$_+(S)$

Lower bound for correlation polytope

- Correlation polytope: COR$(n) =$ conv$\{bb^T \mid b \in \{0,1\}^n\}$
- Occurs naturally in NP-hard problems, e.g. MaxClique
- We can find 2^n valid constraints (indexed by $a \in \{0,1\}^n$) whose slacks w.r.t. vertex bb^T are $M_{ab} = (1 - a^T b)^2$
- Nondeterministic communication complexity of M was already analyzed in the quantum computing! (dW'00)
- Take slack matrix S for COR(n), with 2^n vertices bb^T for columns, 2^n a-constraints for first 2^n rows, remaining facets for other rows

$$S = \left[\begin{array}{ccc} & \vdots & \\ \cdots & M_{ab} & \cdots \\ & \vdots & \\ \hline & \vdots & \end{array}\right]$$

- xc(COR$(n)) =$ rank$_+(S) \geq$ rank$_+(M)$

Lower bound for correlation polytope

- Correlation polytope: COR$(n) =$ conv$\{bb^T \mid b \in \{0,1\}^n\}$
- Occurs naturally in NP-hard problems, e.g. MaxClique
- We can find 2^n valid constraints (indexed by $a \in \{0,1\}^n$) whose slacks w.r.t. vertex bb^T are $M_{ab} = (1 - a^T b)^2$
- Nondeterministic communication complexity of M was already analyzed in the quantum computing! (dW'00)
- Take slack matrix S for COR(n), with 2^n vertices bb^T for columns, 2^n a-constraints for first 2^n rows, remaining facets for other rows

$$S = \left[\begin{array}{ccc} & \vdots & \\ \cdots & M_{ab} & \cdots \\ & \vdots & \\ \hline & \vdots & \end{array}\right]$$

- $xc(\text{COR}(n)) = \text{rank}_+(S) \geq \text{rank}_+(M) \geq 2^{\Omega(n)}$

Consequences for other polytopes

Consequences for other polytopes

- Via reduction from the correlation polytope:

Consequences for other polytopes

- Via reduction from the correlation polytope:
  - TSP-polytope has extension complexity $\geq 2^{\sqrt{n}}$

Consequences for other polytopes

- Via reduction from the correlation polytope:
  - TSP-polytope has extension complexity $\geq 2^{\sqrt{n}}$
    Recently improved to $\geq 2^{n}$ by Rothvoss

Consequences for other polytopes

- Via reduction from the correlation polytope:
  - TSP-polytope has extension complexity $\geq 2^{\sqrt{n}}$
    Recently improved to $\geq 2^n$ by Rothvoss
  - CUT-polytope has extension complexity $\geq 2^n$

Consequences for other polytopes

- Via reduction from the correlation polytope:
  - TSP-polytope has extension complexity $\geq 2^{\sqrt{n}}$
    Recently improved to $\geq 2^n$ by Rothvoss
  - CUT-polytope has extension complexity $\geq 2^n$
  - For specific graphs, the stable-set polytope has extension complexity $\geq 2^n$

Consequences for other polytopes

- Via reduction from the correlation polytope:
  - TSP-polytope has extension complexity $\geq 2^{\sqrt{n}}$
    Recently improved to $\geq 2^n$ by Rothvoss
  - CUT-polytope has extension complexity $\geq 2^n$
  - For specific graphs, the stable-set polytope has extension complexity $\geq 2^n$
- So every linear program based on extended formulations needs exponentially many constraints

Consequences for other polytopes

- Via reduction from the correlation polytope:
  - TSP-polytope has extension complexity $\geq 2^{\sqrt{n}}$
    Recently improved to $\geq 2^n$ by Rothvoss
  - CUT-polytope has extension complexity $\geq 2^n$
  - For specific graphs, the stable-set polytope has extension complexity $\geq 2^n$
- So every linear program based on extended formulations needs exponentially many constraints
- This rules out many efficient algorithms for NP-hard problems, and refutes all P=NP “proofs” à la Swart