Solving Curved Linear Programs via the Shadow Simplex Method

Daniel Dadush[1] Nicolai Hähnle[2]

[1]Centrum Wiskunde & Informatica (CWI)

[2]Bonn Universität

CWI Scientific Meeting 01/15

Outline

1. Introduction
   - Linear Programming and its Applications
   - The Simplex Method
   - Results
2. The Shadow Simplex Method
   - Pivot Rule
   - 3-Step Shadow Simplex Path
3. Conclusions

Outline

Linear Programming

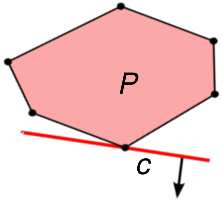
- Linear Programming (LP): linear constraints & linear objective with continuous variables.

$$\max \quad c^T x$$
$$\text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n \quad (A \text{ has } m \text{ rows, } n \text{ columns})$$

Linear Programming

- Linear Programming (LP): linear constraints & linear objective with continuous variables.

$$\max \quad c^T x$$
$$\text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n \quad (A \text{ has } m \text{ rows, } n \text{ columns})$$

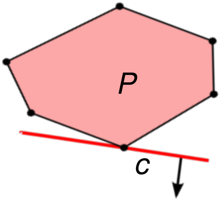


Linear Programming

- Linear Programming (LP): linear constraints & linear objective with continuous variables.

$$\max \quad c^T x$$

$$\text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n \quad (A \text{ has } m \text{ rows, } n \text{ columns})$$

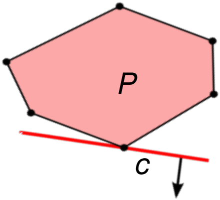


- Amazingly versatile modeling language.

Linear Programming

- Linear Programming (LP): linear constraints & linear objective with continuous variables.

$$\max \quad c^T x$$
$$\text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n \quad (A \text{ has } m \text{ rows, } n \text{ columns})$$



- Amazingly versatile modeling language.
- Generally provides a “relaxed” view of a desired optimization problem, but can be solved in polynomial time!

Mixed Integer Programming

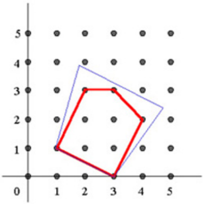
- Mixed Integer Programming (MIP): models both continuous and discrete choices.

$$\max \quad c^T x + d^T y$$
$$\text{subject to } Ax + By \leq b, \quad x \in \mathbb{R}^{n_1}, y \in \mathbb{Z}^{n_2}$$

Mixed Integer Programming

- Mixed Integer Programming (MIP): models both continuous and discrete choices.

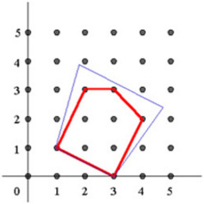
$$\max \quad c^T x + d^T y$$
$$\text{subject to } Ax + By \leq b, \quad x \in \mathbb{R}^{n_1}, y \in \mathbb{Z}^{n_2}$$



Mixed Integer Programming

- Mixed Integer Programming (MIP): models both continuous and discrete choices.

$$\max \quad c^T x + d^T y$$
$$\text{subject to } Ax + By \leq b, \quad x \in \mathbb{R}^{n_1}, y \in \mathbb{Z}^{n_2}$$

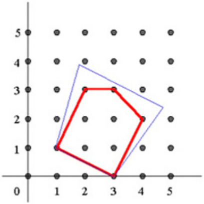


- One of the most successful modeling language for many real world applications. While instances can be extremely hard to solve (MIP is NP-hard), many practical instances are not.

Mixed Integer Programming

- Mixed Integer Programming (MIP): models both continuous and discrete choices.

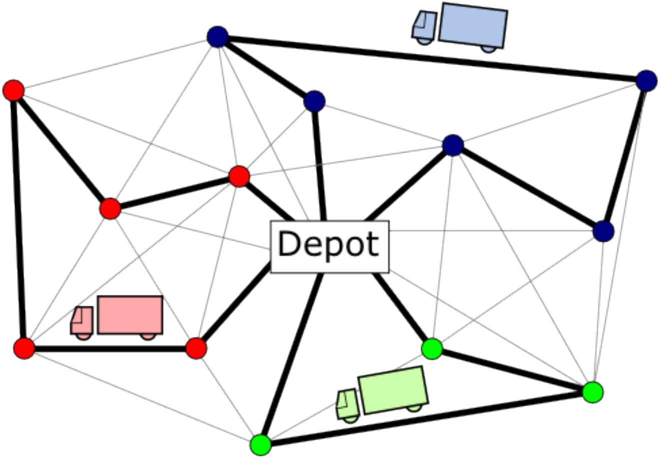
$$\max \quad c^T x + d^T y$$
$$\text{subject to } Ax + By \leq b, \quad x \in \mathbb{R}^{n_1}, y \in \mathbb{Z}^{n_2}$$



- Many sophisticated software packages exist for these models (CPLEX, Gurobi, etc.). MIP solving is now considered a *mature and practical* technology.

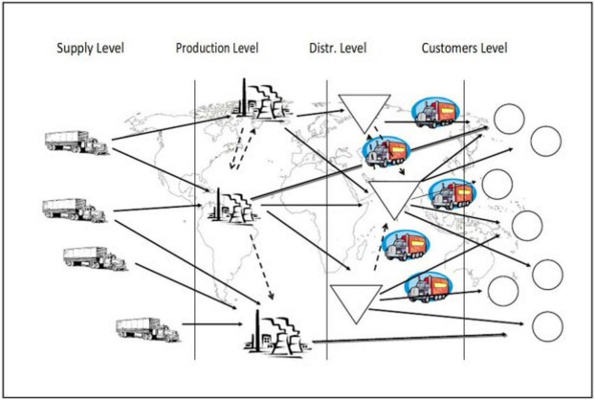
Sample Applications

- Routing delivery / pickup trucks for customers.



Sample Applications

- Optimizing supply chain logistics.



Standard Framework for Solving MIPs

- Relax integrality of the variables.

Standard Framework for Solving MIPs

- Relax integrality of the variables.

$$\max \quad c^T x + d^T y$$
$$\text{subject to } Ax + By \leq b, \quad x \in \mathbb{R}^{n_1}, y \in \mathbb{Z}^{n_2}$$

Standard Framework for Solving MIPs

- Relax integrality of the variables.

$$\begin{aligned} \max \quad & c^T x + d^T y \\ & \text{subject to } Ax + By \leq b, \quad x \in \mathbb{R}^{n_1}, y \in \mathbb{R}^{n_2} \end{aligned}$$

Standard Framework for Solving MIPs

- Relax integrality of the variables.

$$\max \quad c^T x + d^T y$$
$$\text{subject to } Ax + By \leq b, \quad x \in \mathbb{R}^{n_1}, y \in \mathbb{R}^{n_2}$$

- Solve the LP.

Standard Framework for Solving MIPs

- Relax integrality of the variables.

$$\max \quad c^T x + d^T y \quad \text{subject to } Ax + By \leq b, \quad x \in \mathbb{R}^{n_1}, y \in \mathbb{R}^{n_2}$$

- Solve the LP.
- Add extra constraints to tighten the LP or “guess” the values of some of the integer variables. Repeat.

Standard Framework for Solving MIPs

- Relax integrality of the variables.

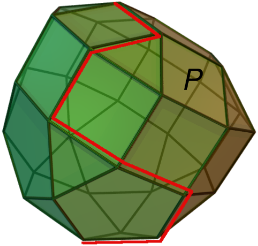
$$\max \quad c^T x + d^T y$$
$$\text{subject to } Ax + By \leq b, \quad x \in \mathbb{R}^{n_1}, y \in \mathbb{R}^{n_2}$$

- Solve the LP.
- Add extra constraints to tighten the LP or “guess” the values of some of the integer variables. Repeat.
- Need to solve a lot of LPs quickly.

Linear Programming via the Simplex Method

$$\max \quad c^T x$$
$$\text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n$$

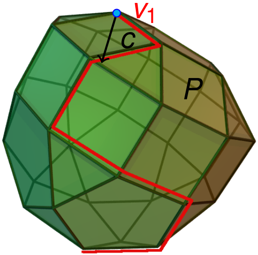
Simplex Method: move from vertex to vertex along the graph of P until the optimal solution is found.



Linear Programming via the Simplex Method

$$\max \quad c^T x$$
$$\text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n$$

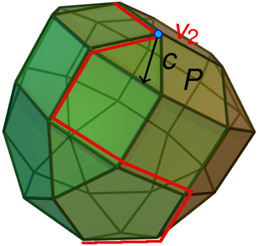
Simplex Method: move from vertex to vertex along the graph of P until the optimal solution is found.



Linear Programming via the Simplex Method

$$\max \quad c^T x \qquad \text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n$$

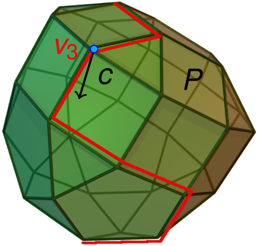
Simplex Method: move from vertex to vertex along the graph of P until the optimal solution is found.



Linear Programming via the Simplex Method

$$\max \quad c^T x$$
$$\text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n$$

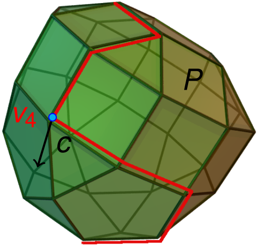
Simplex Method: move from vertex to vertex along the graph of P until the optimal solution is found.



Linear Programming via the Simplex Method

$$\max \quad c^T x$$
$$\text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n$$

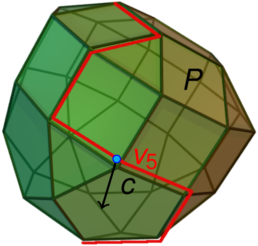
Simplex Method: move from vertex to vertex along the graph of P until the optimal solution is found.



Linear Programming via the Simplex Method

$$\max \quad c^T x$$
$$\text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n$$

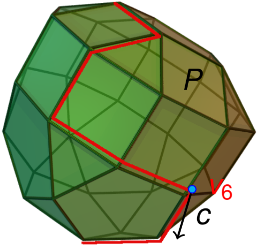
Simplex Method: move from vertex to vertex along the graph of P until the optimal solution is found.



Linear Programming via the Simplex Method

$$\max \quad c^T x \qquad \text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n$$

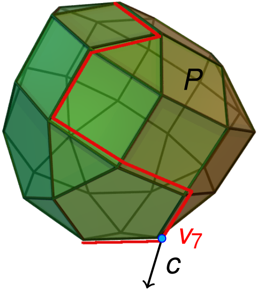
Simplex Method: move from vertex to vertex along the graph of P until the optimal solution is found.



Linear Programming via the Simplex Method

$$\max \quad c^T x$$
$$\text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n$$

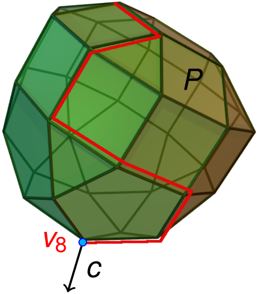
Simplex Method: move from vertex to vertex along the graph of P until the optimal solution is found.



Linear Programming via the Simplex Method

$$\max \quad c^T x$$
$$\text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n$$

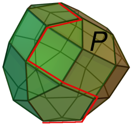
Simplex Method: move from vertex to vertex along the graph of P until the optimal solution is found.



Linear Programming via the Simplex Method

$$\max \quad c^T x$$
$$\text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n$$

A has n columns, m rows.

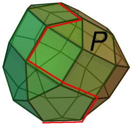


Question

Why is simplex so popular?

Linear Programming via the Simplex Method

$$\max \quad c^T x$$
$$\text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n$$
A has n columns, m rows.



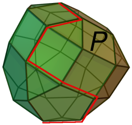
Question

Why is simplex so popular?

- "Easy" to reoptimize when adding an extra constraint or variable.

Linear Programming via the Simplex Method

$$\max \quad c^T x$$
$$\text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n$$
A has n columns, m rows.



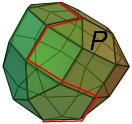
Question

Why is simplex so popular?

- “Easy” to reoptimize when adding an extra constraint or variable.
- Vertex solutions are often “nice” (e.g. sparse, easy to interpret).

Linear Programming via the Simplex Method

$$\max \quad c^T x$$
$$\text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n$$
A has n columns, m rows.



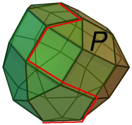
Question

Why is simplex so popular?

- "Easy" to reoptimize when adding an extra constraint or variable.
- Vertex solutions are often "nice" (e.g. sparse, easy to interpret).
- Simplex pivots implementable using "simple" linear algebra. Worst case $O(mn)$ time per iteration.

Linear Programming via the Simplex Method

$$\max \quad c^T x$$
$$\text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n$$
A has n columns, m rows.

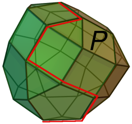


Problem

No known pivot rule is proven to converge in polynomial time!!!

Linear Programming via the Simplex Method

$$\max \quad c^T x$$
$$\text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n$$
A has n columns, m rows.



Problem

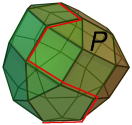
No known pivot rule is proven to converge in polynomial time!!!

Simplex lower bounds:

- Klee-Minty (1972): designed "deformed cubes", providing worst case examples for many pivot rules.
- Friedmann et al. (2011): systematically designed bad examples using Markov decision processes.
- In these examples, the pivot rule is tricked into taking an (sub)exponentially long path, even though short paths exists.

Linear Programming via the Simplex Method

$$\max \quad c^T x$$
$$\text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n$$
A has n columns, m rows.



Problem

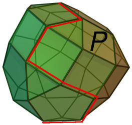
No known pivot rule is proven to converge in polynomial time!!!

Simplex upper bounds:

- Kalai (1992), Matousek-Sharir-Welzl (1992-96): Random facet rule requires $2^{O(\sqrt{n \log m})}$ pivots on expectation.

Linear Programming via the Simplex Method

$$\max \quad c^T x$$
$$\text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n$$
A has n columns, m rows.



Problem

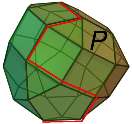
No known pivot rule is proven to converge in polynomial time!!!

Polynomial bounds for random or smoothed linear programs:

Borgwardt (82), Smale (83), Adler (83), Todd (83), Haimovich (83), Meggido (86), Adler-Shamir-Karp (86,87), Spielman-Teng (01,04), Spielman-Deshpande (05), Spielman-Kelner (06), Vershynin (06)

Linear Programming via the Simplex Method

$$\max \quad c^T x$$
$$\text{subject to } Ax \leq b, \quad x \in \mathbb{R}^n$$
A has n columns, m rows.



Problem

No known pivot rule is proven to converge in polynomial time!!!

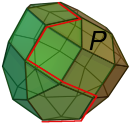
Polynomial bounds for random or smoothed linear programs:

Borgwardt (82), Smale (83), Adler (83), Todd (83), Haimovich (83), Meggido (86), Adler-Shamir-Karp (86,87), Spielman-Teng (01,04), Spielman-Deshpande (05), Spielman-Kelner (06), Vershynin (06)

These works rely on the *shadow simplex method*.

Linear Programming and the Hirsch Conjecture

$$P = \{x \in \mathbb{R}^n : Ax \leq b\}, \quad A \in \mathbb{R}^{m \times n}$$



P lives in $\mathbb{R}^n$ (ambient dimension is n) and has m constraints.

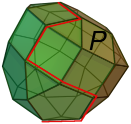
Besides the computational efficiency of the simplex method, an even more basic question is not understood:

Question

How can we bound the length of paths on the graph of P? I.e. how to bound the ***diameter*** *of P?*

Linear Programming and the Hirsch Conjecture

$$P = \{x \in \mathbb{R}^n : Ax \le b\},$$
$$A \in \mathbb{R}^{m \times n}$$



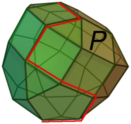
P lives in $\mathbb{R}^n$ (ambient dimension is n) and has m constraints.

Conjecture (Polynomial Hirsch Conjecture)

The diameter of P is bounded by a polynomial in the dimension n and number of constraints m.

Linear Programming and the Hirsch Conjecture

$$P = \{x \in \mathbb{R}^n : Ax \leq b\},$$
$$A \in \mathbb{R}^{m \times n}$$



P lives in $\mathbb{R}^n$ (ambient dimension is n) and has m constraints.

Conjecture (Polynomial Hirsch Conjecture)

The diameter of P is bounded by a polynomial in the dimension n and number of constraints m.

Diameter upper bounds:

- Barnette, Larman (1974): $\frac{1}{3}2^{n-2}(m - n + \frac{5}{2})$.
- Kalai, Kleitman (1992), Todd (2014): $(m - n)^{\log n}$.

Well-Conditioned Polytopes

$$P = \{x \in \mathbb{R}^n \ : \ Ax \leq b\}, \ \ A \in \mathbb{Z}^{m \times n}$$

Definition (Bounded Subdeterminants)

An integer matrix A has subdeterminants bounded by Δ if every square submatrix B of A satisfies $|\det(B)| \leq \Delta$.

Well-Conditioned Polytopes

$$P = \{x \in \mathbb{R}^n \ : \ Ax \leq b\}, \quad A \in \mathbb{Z}^{m \times n}$$

Definition (Bounded Subdeterminants)

An integer matrix A has subdeterminants bounded by Δ if every square submatrix B of A satisfies $|\det(B)| \leq \Delta$.

- Diameter Bound:
  - Bonifas, Di Summa, Eisenbrand, Hähnle, Niemeier (2012): non-constructive $O(n^{3.5}\Delta^2 \log n\Delta)$ bound.
  - Brunsch,Röglin (2013): shadow simplex method finds paths of length $O(mn^3\Delta^4)$.

Well-Conditioned Polytopes

$$P = \{x \in \mathbb{R}^n \ : \ Ax \leq b\}, \ \ A \in \mathbb{Z}^{m \times n}$$

Definition (Bounded Subdeterminants)

An integer matrix A has subdeterminants bounded by Δ if every square submatrix B of A satisfies $|\det(B)| \leq \Delta$.

- Diameter Bound:
  - Bonifas, Di Summa, Eisenbrand, Hähnle, Niemeier (2012): non-constructive $O(n^{3.5}\Delta^2 \log n\Delta)$ bound.
  - Brunsch,Röglin (2013): shadow simplex method finds paths of length $O(mn^3\Delta^4)$.
- Optimization:
  - Dyer, Frieze (1994): $\Delta = 1$ (totally unimodular) random walk simplex uses $O(m^{16}n^6 \log(mn)^3)$ pivots.
  - Eisenbrand, Vempala (2014): random walk simplex uses $O(m \, \mathrm{poly}(n, \Delta))$ pivots.

Faster Shadow Simplex

$$P = \{x \in \mathbb{R}^n \ : \ Ax \leq b\}, \ \ A \in \mathbb{Z}^{m \times n}$$

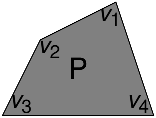
Subdeterminants of A bounded by Δ.

Theorem (D., Hähnle 2014+)

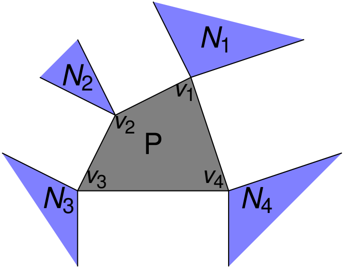
- *Diameter is bounded by* $O(n^3 \Delta^2 \ln(n\Delta))$.
- *Can solve LP using* $O(n^5 \Delta^2 \ln(n\Delta))$ *pivots on expectation.*

Based on a new analysis and variant of the **shadow simplex method.**

τ-wide Polyhedra

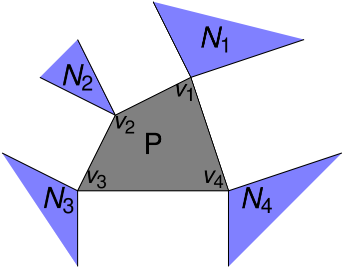


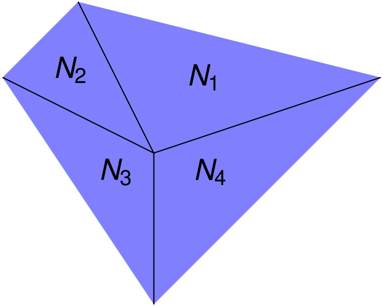
τ-wide Polyhedra



- Normal cone at v is all objectives maximized at v.

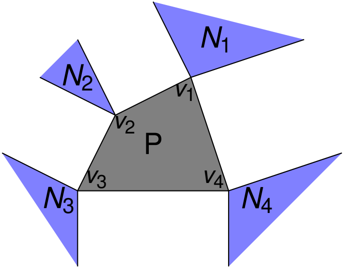
τ-wide Polyhedra

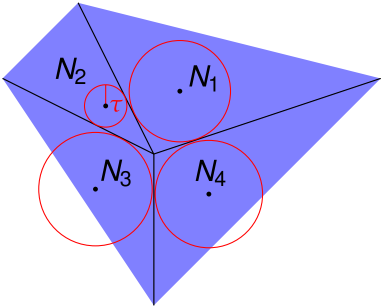




- Normal cone at v is all objectives maximized at v.
- Normal fan is the set of normal cones.

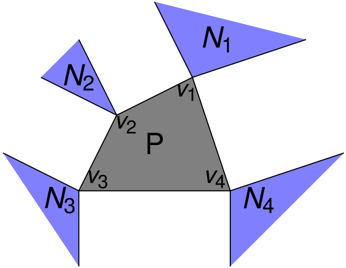
τ-wide Polyhedra

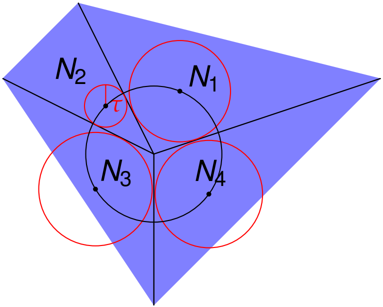




- Normal cone at *v* is all objectives maximized at *v*.
- Normal fan is the set of normal cones.
- *P* is τ-wide if each normal cone contains a unit ball of radius τ centered on the unit sphere.

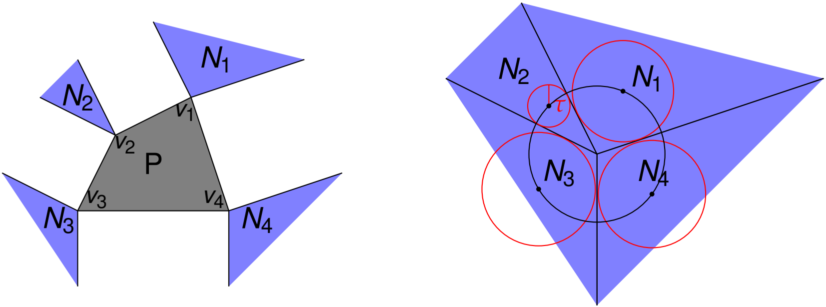
τ-wide Polyhedra





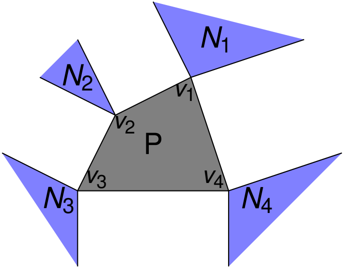
- Normal cone at v is all objectives maximized at v.
- Normal fan is the set of normal cones.
- P is τ-wide if each normal cone contains a unit ball of radius τ centered on the unit sphere.

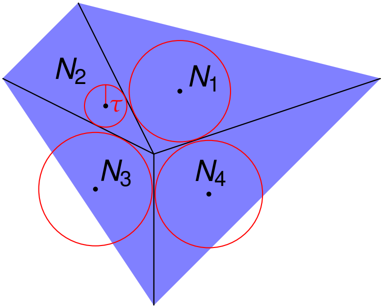
τ-wide Polyhedra



- P has subdeterminant bound $\Delta \Rightarrow P$ is τ-wide for $\tau = 1/(n\Delta)^2$.
- Normal cone at v is all objectives maximized at v.
- Normal fan is the set of normal cones.
- P is τ-wide if each normal cone contains a unit ball of radius τ centered on the unit sphere.

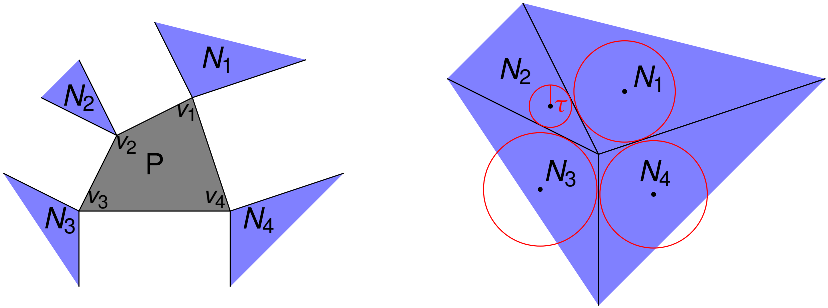
τ-wide Polyhedra





In 2 dimensions all interior angles are at most $\pi - 2\tau$, i.e. *sharp*.

τ-wide Polyhedra



In 2 dimensions all interior angles are at most $\pi - 2\tau$, i.e. *sharp*.

Theorem (D.-Hähnle 2014)

If P is τ-wide then the diameter of P is bounded by $O(n/\tau \ln(1/\tau))$.

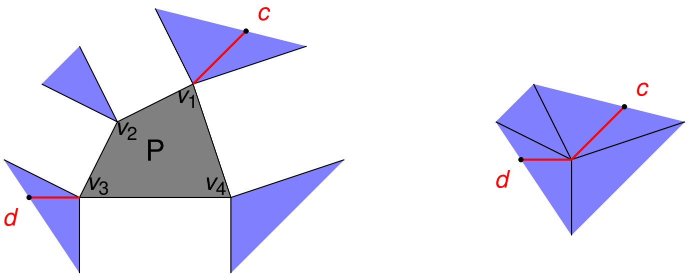
Outline

1. Introduction
   - Linear Programming and its Applications
   - The Simplex Method
   - Results

2. The Shadow Simplex Method
   - Pivot Rule
   - 3-Step Shadow Simplex Path

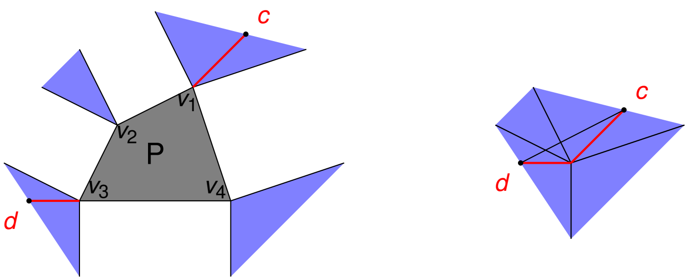
3. Conclusions

Pivot Rule



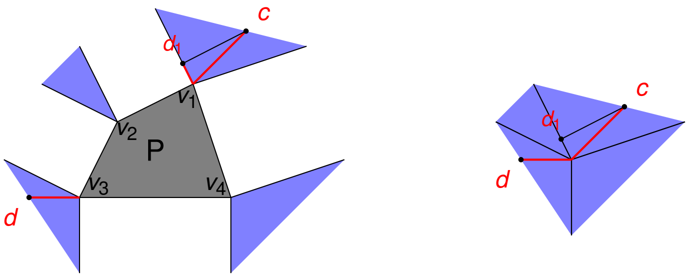
- Move from v_1 to v_3 by following $[c, d]$ through the normal fan.
- Pivot steps correspond to crossing facets of the normal fan.

Pivot Rule



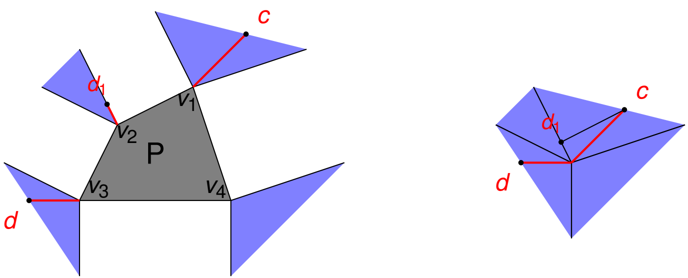
- Move from v_1 to v_3 by following $[c, d]$ through the normal fan.
- Pivot steps correspond to crossing facets of the normal fan.

Pivot Rule



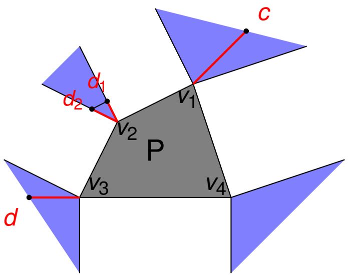
- Move from v_1 to v_3 by following $[c, d]$ through the normal fan.
- Pivot steps correspond to crossing facets of the normal fan.

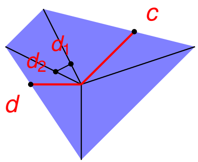
Pivot Rule



- Move from v_1 to v_3 by following $[c, d]$ through the normal fan.
- Pivot steps correspond to crossing facets of the normal fan.

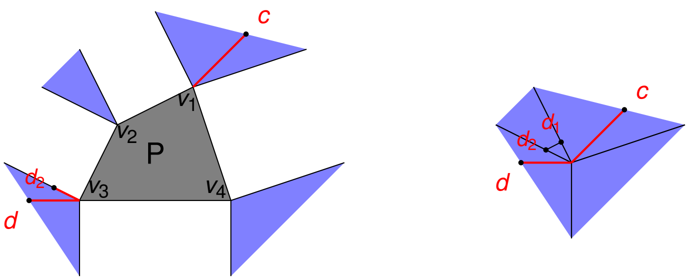
Pivot Rule





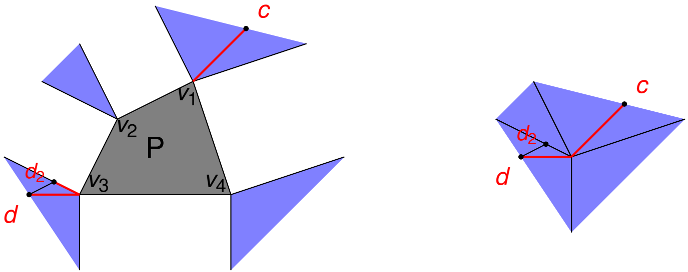
- Move from v_1 to v_3 by following $[c, d]$ through the normal fan.
- Pivot steps correspond to crossing facets of the normal fan.

Pivot Rule



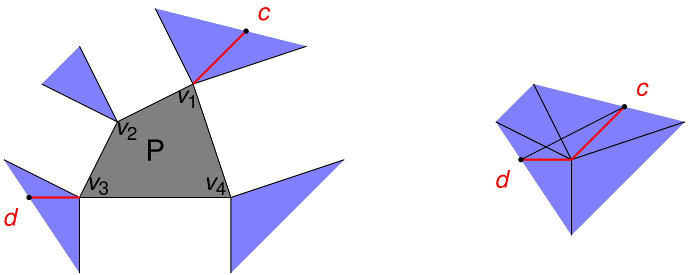
- Move from v_1 to v_3 by following $[c, d]$ through the normal fan.
- Pivot steps correspond to crossing facets of the normal fan.

Pivot Rule



- Move from v_1 to v_3 by following $[c, d]$ through the normal fan.
- Pivot steps correspond to crossing facets of the normal fan.

Pivot Rule

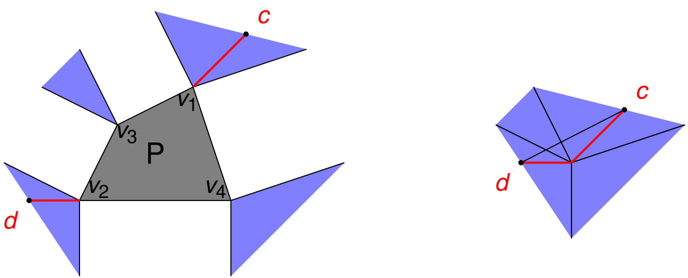


- Move from v_1 to v_3 by following $[c, d]$ through the normal fan.
- Pivot steps correspond to crossing facets of the normal fan.

Question

How can we bound the number of intersections with the normal fan?

Pivot Rule



Question

How can we bound the number of intersections with the normal fan?

Polynomial bounds for random and smoothed linear programs:

Borgwardt (82), Smale (83), Adler (83), Todd (83), Haimovich (83), Meggido (86), Adler-Shamir-Karp (86,87), Spielman-Teng (01,04), Spielman-Deshpande (05), Spielman-Kelner (06), Vershynin (06)

3-Step Shadow Simplex Path

- We use a 3-step shadow simplex path:

$$c \xrightarrow{(a)} c + X \xrightarrow{(b)} d + X \xrightarrow{(c)} d$$

where X is distributed proportional to $e^{-\|x\|}$.

3-Step Shadow Simplex Path

- We use a 3-step shadow simplex path:

$$c \xrightarrow{(a)} c + X \xrightarrow{(b)} d + X \xrightarrow{(c)} d$$

  where X is distributed proportional to $e^{-\|x\|}$.
- c is an objective maximized at the starting vertex.

3-Step Shadow Simplex Path

- We use a 3-step shadow simplex path:

$$c \quad \xrightarrow{(a)} \quad c + X \quad \xrightarrow{(b)} \quad d + X \quad \xrightarrow{(c)} \quad d$$

  where X is distributed proportional to $e^{-\|x\|}$.
- c is an objective maximized at the starting vertex.
- d is the LP objective function.

3-Step Shadow Simplex Path

- We use a 3-step shadow simplex path:

$$c \xrightarrow{(a)} c + X \xrightarrow{(b)} d + X \xrightarrow{(c)} d$$

  where X is distributed proportional to $e^{-\|x\|}$.
- c is an objective maximized at the starting vertex.
- d is the LP objective function.

Theorem (D.-Hähnle 2014)

Assume P is an n-dimensional τ-wide polytope.

- *Phase (b). Expected number of crossings of $[c + X, d + X]$ with normal fan is $O(\|c - d\| / \tau)$.*

3-Step Shadow Simplex Path

- We use a 3-step shadow simplex path:

$$c \xrightarrow{(a)} c + X \xrightarrow{(b)} d + X \xrightarrow{(c)} d$$

  where X is distributed proportional to $e^{-\|x\|}$.
- c is an objective maximized at the starting vertex.
- d is the LP objective function.

Theorem (D.-Hähnle 2014)

Assume P is an n-dimensional τ-wide polytope.

- *Phase (b). Expected number of crossings of $[c + X, d + X]$ with normal fan is $O(\|c - d\| / \tau)$.*
- *Phase (a+c). Expected number of crossings of $[c + \alpha X, c + X]$ (same for d), for $\alpha \in (0, 1]$, with normal fan is $O(n/\tau \ln(1/\alpha))$.*

3-Step Shadow Simplex Path

- We use a 3-step shadow simplex path:

$$c \xrightarrow{(a)} c + X \xrightarrow{(b)} d + X \xrightarrow{(c)} d$$

  where X is distributed proportional to $e^{-\|x\|}$.
- c is an objective maximized at the starting vertex.
- d is the LP objective function.

Theorem (D.-Hähnle 2014)

Assume P is an n-dimensional τ-wide polytope.

- *Phase (b). Expected number of crossings of $[c + X, d + X]$ with normal fan is $O(\|c - d\| / \tau)$.*
- *Phase (a+c). Expected number of crossings of $[c + \alpha X, c + X]$ (same for d), for $\alpha \in (0, 1]$, with normal fan is $O(n/\tau \ln(1/\alpha))$.*

Main ingredient for all our results.

Outline

Navigation over the Voronoi Graph

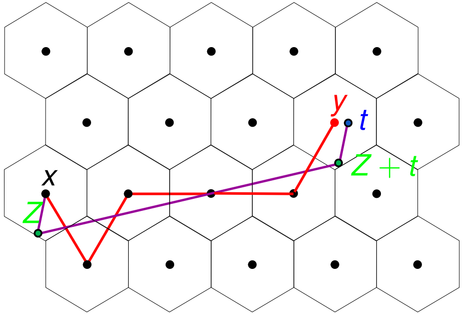


Figure: Randomized Straight Line algorithm

- Closest Vector Problem (CVP): Find closest lattice vector y to t.

Navigation over the Voronoi Graph

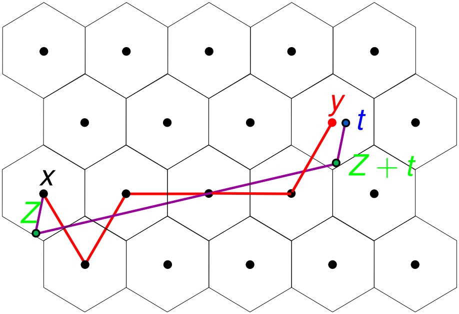


Figure: Randomized Straight Line algorithm

- Closest Vector Problem (CVP): Find closest lattice vector y to t.
- Can reduce CVP to efficient navigation over the Voronoi graph (Sommer,Feder,Shalvi 09; Micciancio,Voulgaris 10-13).

Navigation over the Voronoi Graph

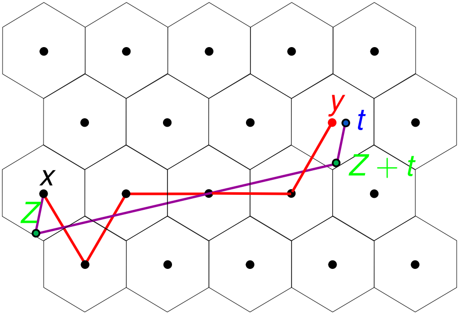


Figure: Randomized Straight Line algorithm

- Closest Vector Problem (CVP): Find closest lattice vector y to t.
- Can move between "nearby" lattice points using a polynomial number of steps (Bonifas, D. 14).

Summary

- New and simpler analysis and variant of the Shadow Simplex method.
- Improved diameter bounds and simplex algorithm for *curved polyhedra*.

Summary

- New and simpler analysis and variant of the Shadow Simplex method.
- Improved diameter bounds and simplex algorithm for *curved polyhedra*.

Thank you!