Nonlinear Discrete Optimization

Shmuel Onn

Technion – Israel Institute of Technology

http://ie.technion.ac.il/~onn

Billerafest 2008 - conference in honor of Lou Billera's 65th birthday

(Update on Lecture Series given at CRM Montréal)

Based on several papers joint with several co-authors including
Berstein, De Loera, Hemmecke, Lee, Rothblum, Weismantel, Wynn

Framework for Nonlinear Discrete Optimization

Shmuel Onn

Framework for Nonlinear Discrete Optimization

The set of feasible points is a subset S of Z^n suitably presented, e.g.

Framework for Nonlinear Discrete Optimization

The set of feasible points is a subset S of Z^n suitably presented, e.g.

- by a system of inequalities

Framework for Nonlinear Discrete Optimization

The set of feasible points is a subset S of Z^n suitably presented, e.g.

- by a system of inequalities

  (often linear inequalities, giving "integer programming")

Framework for Nonlinear Discrete Optimization

The set of feasible points is a subset S of Z^n suitably presented, e.g.

- by a system of inequalities
  (often linear inequalities, giving "integer programming")
- by a suitable oracle (membership, linear optimization, etc.)

Framework for Nonlinear Discrete Optimization

The set of feasible points is a subset S of Z^n suitably presented, e.g.

- by a system of inequalities

  (often linear inequalities, giving "integer programming")

- by a suitable oracle (membership, linear optimization, etc.)

  (often S is {0,1}-valued, giving "combinatorial optimization")

Framework for Nonlinear Discrete Optimization

The set of feasible points is a subset S of Z^n suitably presented, e.g.

- by a system of inequalities
  (often linear inequalities, giving "integer programming")

- by a suitable oracle (membership, linear optimization, etc.)
  (often S is {0,1}-valued, giving "combinatorial optimization")

The objective function is parameterized as $f(w_1x, \ldots, w_dx)$, where $w_1x, \ldots, w_dx$ are linear forms and f is a real valued function on R^d.

Framework for Nonlinear Discrete Optimization

The set of feasible points is a subset S of Z^n suitably presented, e.g.

- by a system of inequalities
  (often linear inequalities, giving "integer programming")
- by a suitable oracle (membership, linear optimization, etc.)
  (often S is {0,1}-valued, giving "combinatorial optimization")

The objective function is parameterized as $f(w_1x, \ldots, w_dx)$, where $w_1x, \ldots, w_dx$ are linear forms and f is a real valued function on R^d.

The problem can be interpreted as multi-objective discrete optimization, where the goal is to optimize the "balancing" by f of d linear criteria:

$$\min \text{ or } \max \{f(w_1x, \ldots, w_dx) : x \text{ in } S\} .$$

Shmuel Onn

Framework for Nonlinear Discrete Optimization

The set of feasible points is a subset S of Z^n suitably presented, e.g.

- by a system of inequalities
  (often linear inequalities, giving "integer programming")
- by a suitable oracle (membership, linear optimization, etc.)
  (often S is {0,1}-valued, giving "combinatorial optimization")

The objective function is parameterized as $f(w_1x, \ldots, w_dx)$, where $w_1x, \ldots, w_dx$ are linear forms and f is a real valued function on R^d.

The problem can be interpreted as multi-objective discrete optimization, where the goal is to optimize the "balancing" by f of d linear criteria:

$$\min \text{ or } \max \{f(w_1x, \ldots, w_dx) : x \text{ in } S\} .$$

It is generally intractable even for fixed $d=1$ and f the identity on R (e.g., it may be NP-hard or even require exponential oracle-time).

Shmuel Onn

Some Applications and Examples Where Our Theory Provides Polynomial Time Algorithms

- Vector Partitioning and Clustering:

- Shaped partition problems (SIAM Opt.)
- Partition problems with convex objectives (Math. OR)

Vector Partitioning

Partition m items evaluated by k criteria to p players, to maximize social utility that is function of the sums of vectors of items each player gets.

Example: Consider m=6 items, k=2 criteria, p=3 players

The criteria-item matrix is:

items

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 9 & 16 & 25 & 36 \end{bmatrix}$$ criteria

Each player should get 2 items

The nonlinear function on k x p matrices is $f(X) = \sum X_{ij}^3$

The matrix of a partition such as π = (34, 56, 12) is:

players

$$A^\pi = \begin{bmatrix} 7 & 11 & 3 \\ 25 & 61 & 5 \end{bmatrix}$$ criteria

The social utility of π is $f(A^\pi) = 244432$

Shmuel Onn

All 90 partitions π of items {1, …,6} To 3 players where each player gets 2 items

The optimal partition is:

π = (34, 56, 12)

with optimal utility:

players

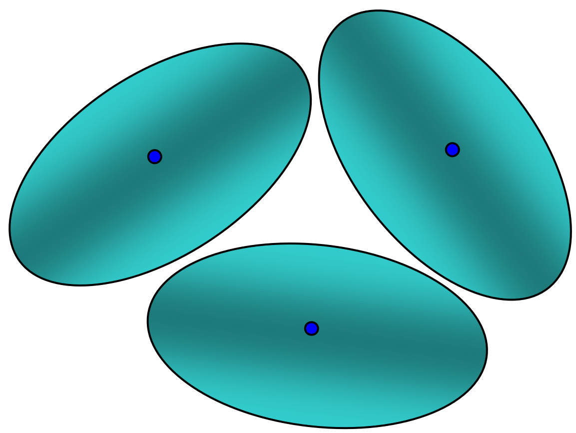
$$A^{\pi} = \begin{bmatrix} 7 & 11 & 3 \\ 25 & 61 & 5 \end{bmatrix}$$ criteria

$f(A^{\pi}) = 244432$

[12 34 56]	[12 35 46]	[12 36 45]	[12 45 36]	[12 46 35]	[12 56 34]
[13 24 56]	[13 25 46]	[13 26 45]	[13 45 26]	[13 46 25]	[13 56 24]
[14 23 56]	[14 25 36]	[14 26 35]	[14 35 26]	[14 36 25]	[14 56 23]
[15 23 46]	[15 24 36]	[15 26 34]	[15 34 26]	[15 36 24]	[15 46 23]
[16 23 45]	[16 24 35]	[16 25 34]	[16 34 25]	[16 35 24]	[16 45 23]
[23 14 56]	[23 15 46]	[23 16 45]	[23 45 16]	[23 46 15]	[23 56 14]
[24 13 56]	[24 15 36]	[24 16 35]	[24 35 16]	[24 36 15]	[24 56 13]
[25 13 46]	[25 14 36]	[25 16 34]	[25 34 16]	[25 36 14]	[25 46 13]
[26 13 45]	[26 14 35]	[26 15 34]	[26 34 15]	[26 35 14]	[26 45 13]
[34 12 56]	[34 15 26]	[34 16 25]	[34 25 16]	[34 26 15]	**[34 56 12]**
[35 12 46]	[35 14 26]	[35 16 24]	[35 24 16]	[35 26 14]	[35 46 12]
[36 12 45]	[36 14 25]	[36 15 24]	[36 24 15]	[36 25 14]	[36 45 12]
[45 12 36]	[45 13 26]	[45 16 23]	[45 23 16]	[45 26 13]	[45 36 12]
[46 12 35]	[46 13 25]	[46 15 23]	[46 23 15]	[46 25 13]	[46 35 12]
[56 12 34]	[56 13 24]	[56 14 23]	[56 23 14]	[56 24 13]	[56 34 12]

Minimum Variance Clustering

Given m points $v^1, \ldots, v^m$ in R^k, group them into p (balanced) clusters so as to minimize the sum of cluster variances .



P=3, k=3, m large

Shmuel Onn

- Matroids and Their Applications:

- Matroids and Their Applications:

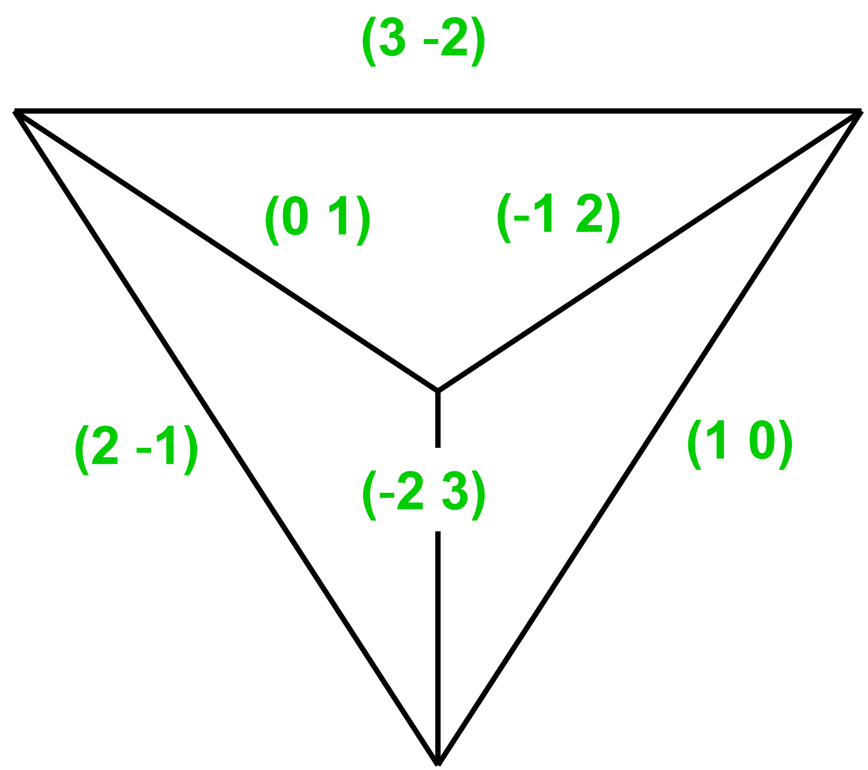
- Spanning trees, polymatroids, intersections of matroids:

- Matroids and Their Applications:

- Spanning trees, polymatroids, intersections of matroids:

Example - spanning trees:

Consider $n=6$, the graph $G=K_4$, $d=2$, weights w, and the Euclidean norm (squared) $f(x) = |x|^2 = x_1^2 + x_2^2$



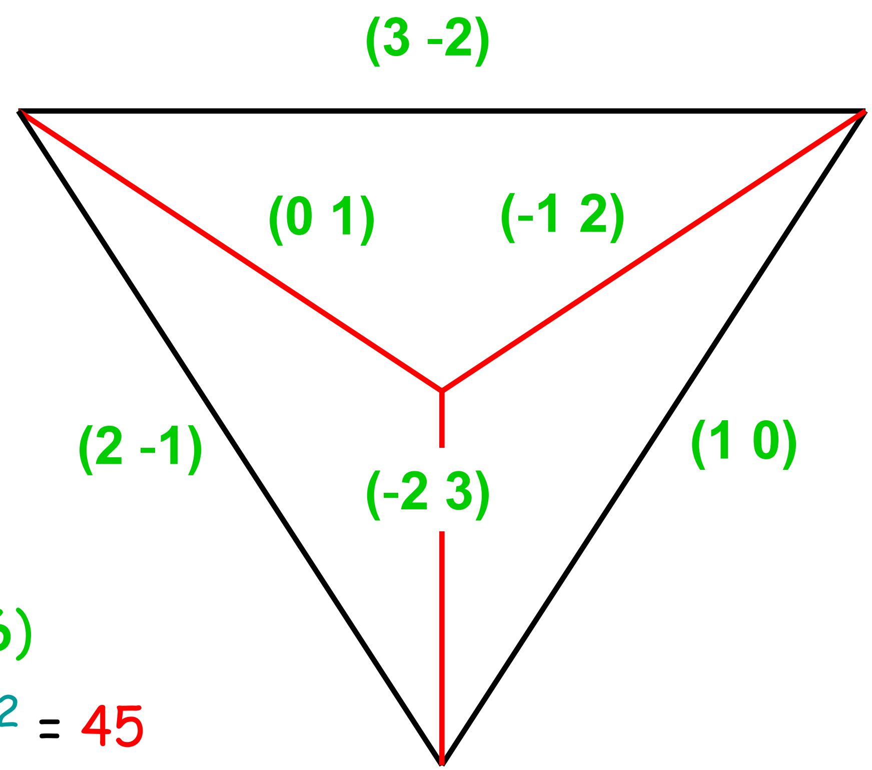
Shmuel Onn

- Matroids and Their Applications:

- Spanning trees, polymatroids, intersections of matroids:

Example - spanning trees:

Consider $n=6$, the graph $G=K_4$, $d=2$, weights w, and the Euclidean norm (squared) $f(x) = |x|^2 = x_1^2 + x_2^2$



The optimal tree x has

$w(x) = (0\ \ 1) + (-1\ \ 2) + (-2\ \ 3) = (-3\ \ 6)$

and objective value $f(w(x)) = |-3\ \ 6|^2 = 45$

Shmuel Onn

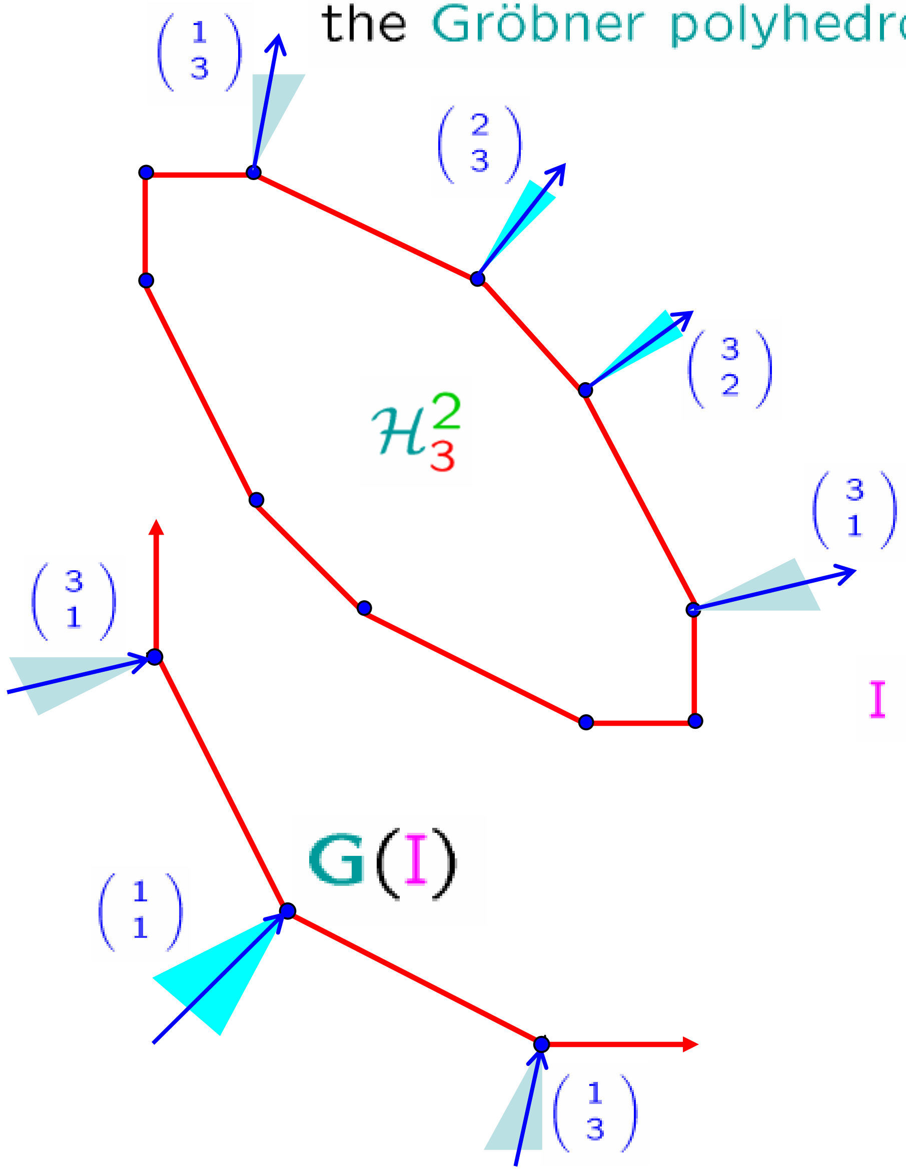
- Matroids and Their Applications:

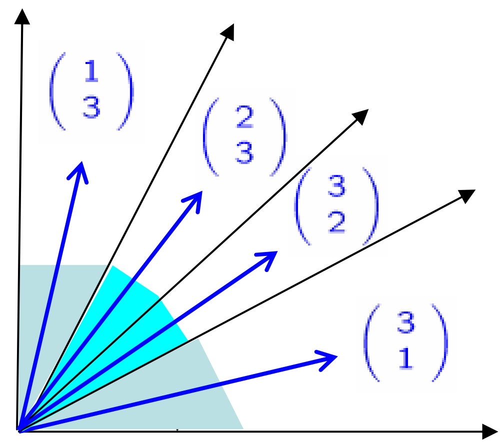
- Spanning trees, polymatroids, intersections of matroids:

- Systems of polynomial equations:
  simultaneous computation of universal Gröbner bases
  for all ideals on the Hilbert Scheme

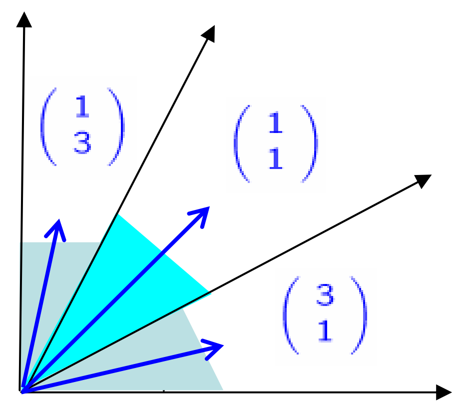
Example: The Hilbert Zonotope $\mathcal{H}_3^2$

It refines the Hilbert polyhedron $\mathbf{H}(\mathrm{Hilb}_3^2)$ and the Gröbner polyhedron $\mathbf{G}(\mathrm{I})$ of every $\mathrm{I} \in \mathrm{Hilb}_3^2$.





$$\mathrm{I} \;=\; \langle x^3 - 3x^2 + 2x,\, y - x^2 \rangle \;\in\; \mathrm{Hilb}_3^2$$

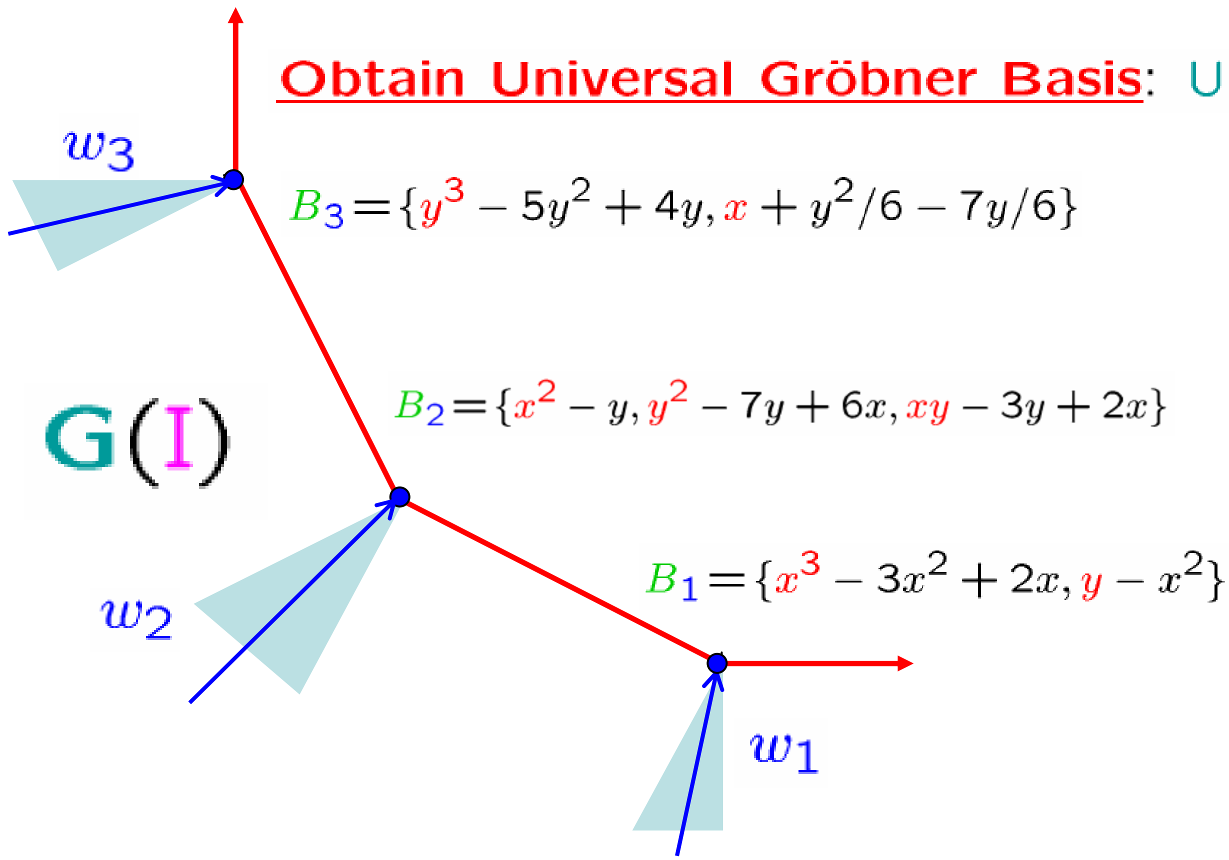


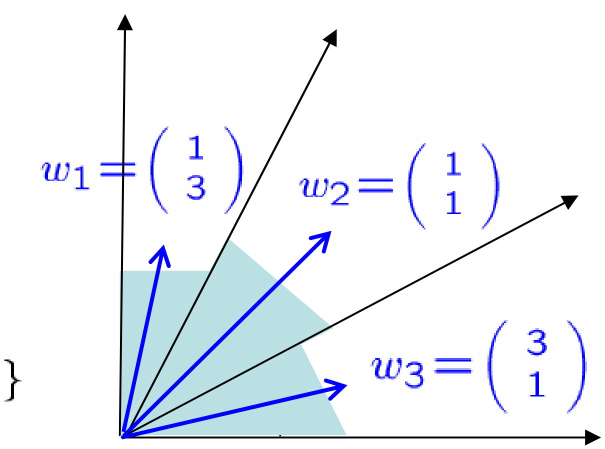
Gröbner Polyhedra

Each generic $w \in \mathbf{R}_+^d$ gives monomial order on $\mathbf{C}[x_1, \ldots, x_d]$ by the value $w \cdot v$ of the exponent $v \in \mathbf{N}^d$ of the monomial $\prod_i x_i^{v_i}$.

A Gröbner polyhedron $\mathbf{G}(\mathrm{I})$ of an ideal I is one with vertices in bijection with reduced Gröbner bases of I, where the RGB of v equals RGB_w whenever w is minimized over $\mathbf{G}(\mathrm{I})$ uniquely at v.

Obtain Universal Gröbner Basis: $\mathrm{UGB}(\mathrm{I}) = B_1 \cup B_2 \cup B_3$





- Matroids and Their Applications:

- Spanning trees, polymatroids, intersections of matroids:

- Systems of polynomial equations:
  simultaneous computation of universal Gröbner bases
  for all ideals on the Hilbert Scheme

- Experimental design and learning:
  finding optimal multivariate polynomial model
  that fits experiment-results or learning-queries

- Matroids and Their Applications:

- Spanning trees, polymatroids, intersections of matroids:

- Systems of polynomial equations:
  simultaneous computation of universal Gröbner bases
  for all ideals on the Hilbert Scheme

- Experimental design and learning:
  finding optimal multivariate polynomial model
  that fits experiment-results or learning-queries

- Convex matroid optimization (SIAM Disc. Math.)
- Convex combinatorial optimization (Disc. Comp. Geom.)
- Cutting corners (Adv. App. Math.)
- The Hilbert zonotope and universal Gröbner bases (Adv. App. Math.)
- Nonlinear matroid optimization and experimental design (SIAM Disc. Math.)
- Nonlinear optimization over a weighted independence system (submitted)

- Multiway Tables and Their Applications:

- Multiway Tables and Their Applications:

- Privacy and confidentiality in statistical data bases

- Multiway Tables and Their Applications:

- Privacy and confidentiality in statistical data bases
- Congestion avoiding (multiway) transportation

- Multiway Tables and Their Applications:

- Privacy and confidentiality in statistical data bases
- Congestion avoiding (multiway) transportation
- Error correcting codes

- Multiway Tables and Their Applications:

- Privacy and confidentiality in statistical data bases
- Congestion avoiding (multiway) transportation
- Error correcting codes
- **Scheme** for (nonlinear) optimization over **any integer program**

- Multiway Tables and Their Applications:

- Privacy and confidentiality in statistical data bases
- Congestion avoiding (multiway) transportation
- Error correcting codes
- **Scheme** for (nonlinear) optimization over **any integer program**

- The complexity of 3-way tables (SIAM Comp.)
- Markov bases of 3-way tables (J. Symb. Comp.)
- All linear and integer programs are slim 3-way programs (SIAM Opt.)
- N-fold integer programming (Disc. Opt. in memory of Dantzig)
- Graver complexity of integer programming (Annals Combin.)
- Nonlinear bipartite matching (Disc. Opt.)
- Convex integer maximization (J. Pure App. Algebra)
- Convex integer minimization (submitted)

Some Geometric Methods:

Convex Discrete Maximization

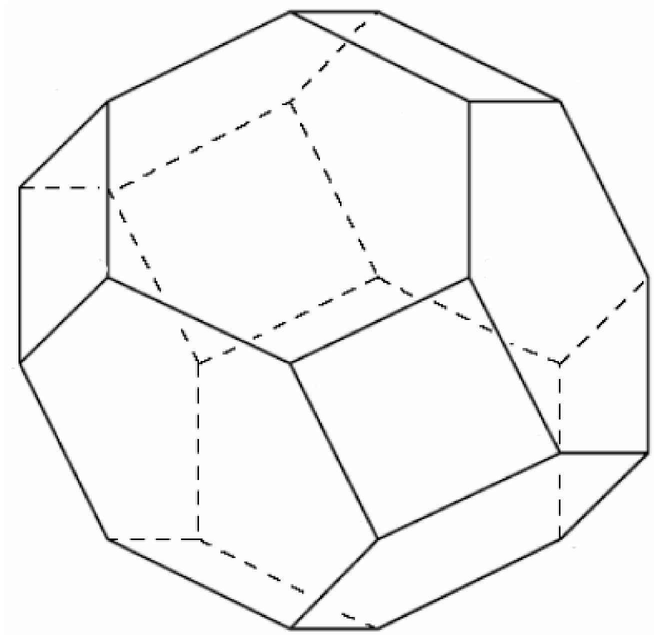
Convex Discrete Maximization

Consider the convex hull $P = \text{conv}\{S\}$ of the feasible set S in Z^n.
When we can control the edge-directions of P, we can reduce convex to linear maximization in strongly polynomial time.

Shmuel Onn

Convex Discrete Maximization

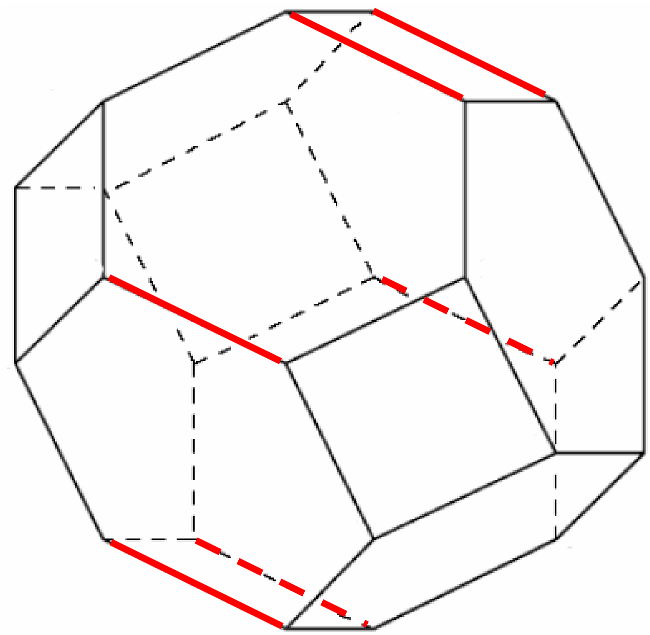
Consider the convex hull P = conv{S} of the feasible set S in Z^n.
When we can control the edge-directions of P, we can reduce convex to linear maximization in strongly polynomial time.



Shmuel Onn

Convex Discrete Maximization

Consider the convex hull $P = \text{conv}\{S\}$ of the feasible set S in Z^n.
When we can control the edge-directions of P, we can reduce convex to linear maximization in strongly polynomial time.

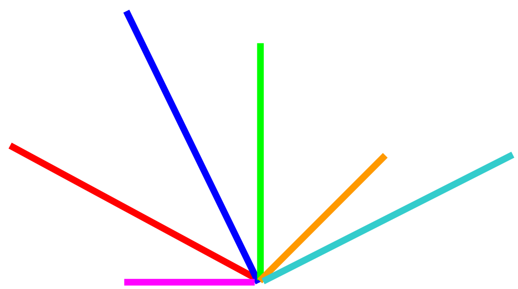


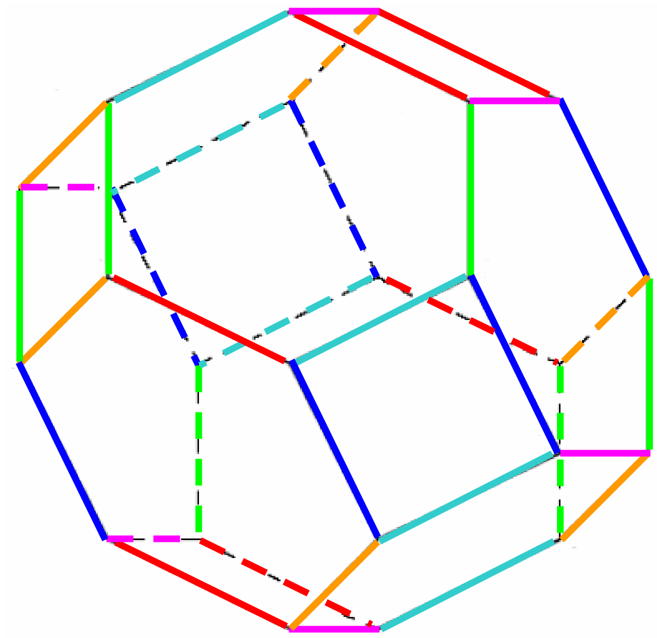
Shmuel Onn

Convex Discrete Maximization

Consider the convex hull **P** = conv{S} of the feasible set S in Z^n.

When we can control the edge-directions of **P**, we can reduce convex to linear maximization in strongly polynomial time.

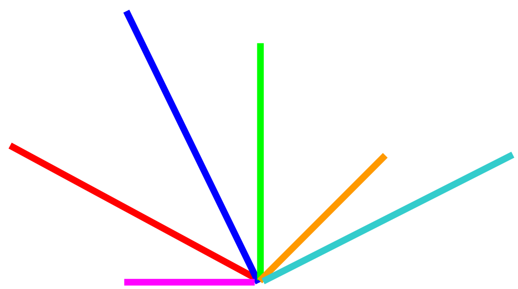


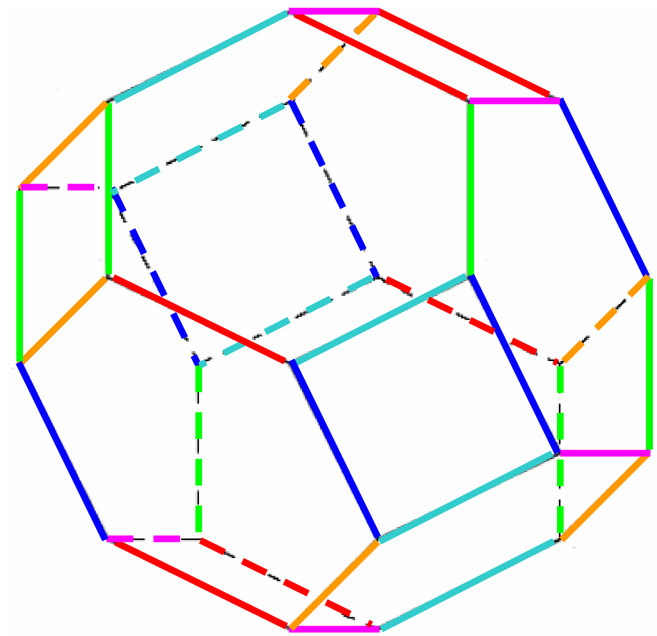


Shmuel Onn

Convex Discrete Maximization

Consider the convex hull $P = \text{conv}\{S\}$ of the feasible set S in Z^n.
When we can control the edge-directions of P, we can reduce convex to linear maximization in strongly polynomial time.





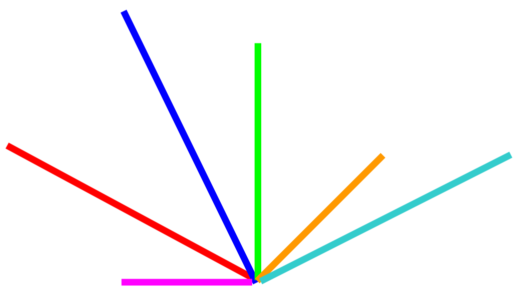
- Cubes: unit vectors 1_i

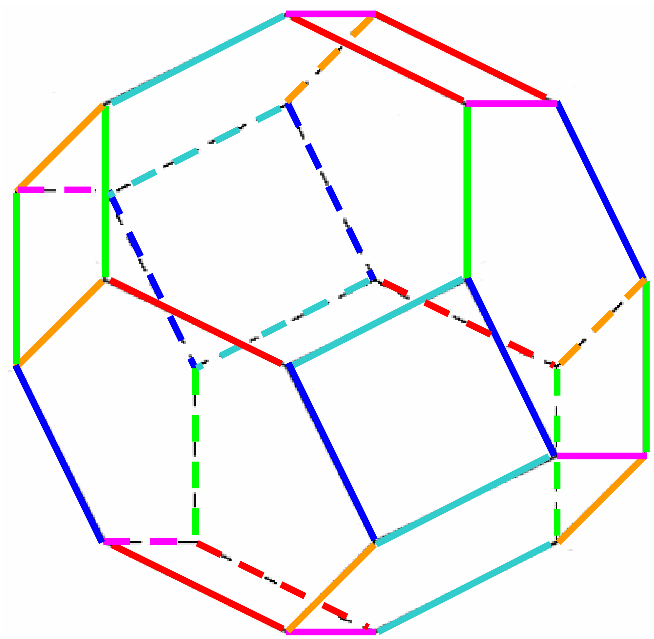
(e.g. 0-1 quadratic programming)

Shmuel Onn

Convex Discrete Maximization

Consider the convex hull $P = \text{conv}\{S\}$ of the feasible set S in Z^n.
When we can control the edge-directions of P, we can reduce convex to linear maximization in strongly polynomial time.



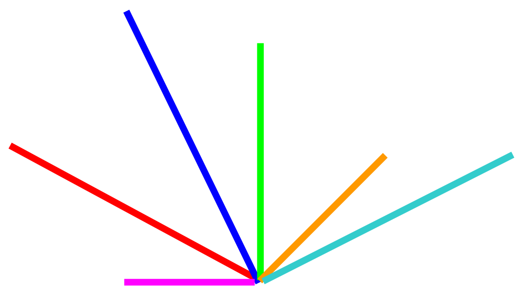


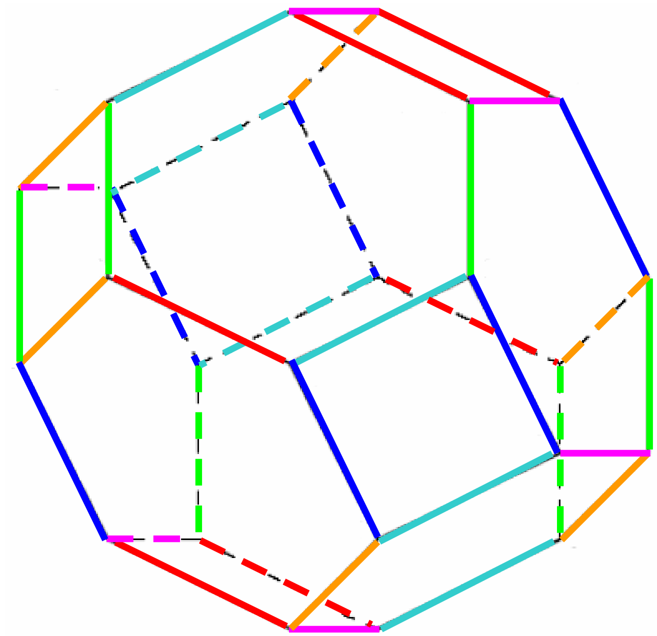
- Cubes: unit vectors 1_i (e.g. 0-1 quadratic programming)
- Matroid polytopes: pairs $1_i - 1_j$ (e.g. spanning trees}

Shmuel Onn

Convex Discrete Maximization

Consider the convex hull $P = \mathrm{conv}\{S\}$ of the feasible set S in Z^n.
When we can control the edge-directions of P, we can reduce convex to linear maximization in strongly polynomial time.





- Cubes: unit vectors 1_i (e.g. 0-1 quadratic programming)
- Matroid polytopes: pairs $1_i - 1_j$ (e.g. spanning trees}
- Transportations: $n \times p$ circuit matrices (e.g. partitioning, clustering)

Shmuel Onn

Convex Discrete Maximization

Theorem: Fix any d. Then for any S in Z^n endowed with a set E that covers all edge-directions of conv{S}, and any convex f presented by a comparison oracle, the convex discrete maximization problem

$$\max \{ f(w_1 x, \ldots, w_d x) \ : \ x \text{ in } S \}$$

reduces to strongly polynomially linear counterparts over S.

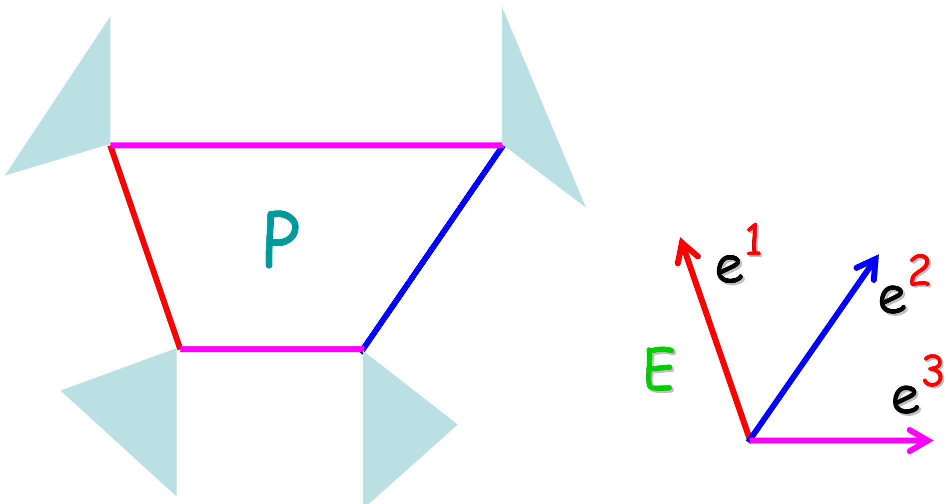
Proof: preliminaries on zonotopes

Lemma: If $E = \{e^1, \ldots, e^m\}$ covers all edge-directions of a polytope P then the zonotope $Z = [-1, 1]\, e^1 + \ldots + [-1, 1]\, e^m$ is a refinement of P.

(Minkowsky, Grunbaum , ...,)

Proof: preliminaries on zonotopes

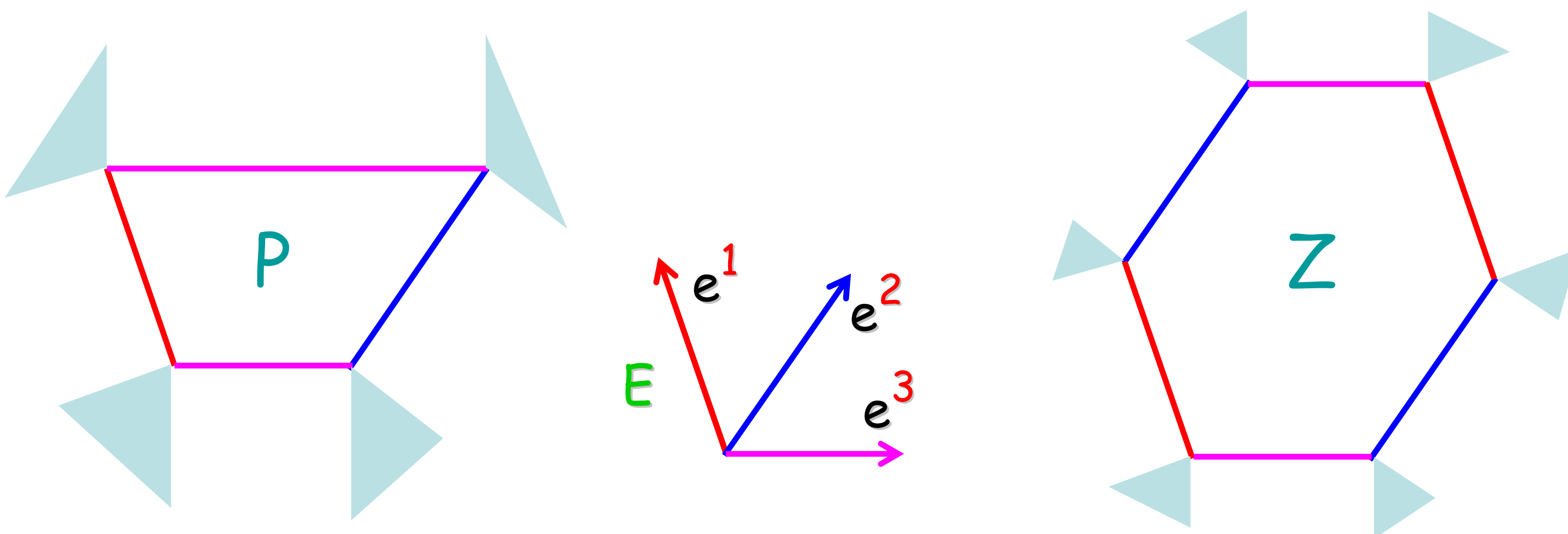
Lemma: If $E = \{e^1, \ldots, e^m\}$ covers all edge-directions of a polytope P then the zonotope $Z = [-1, 1] e^1 + \ldots + [-1, 1] e^m$ is a refinement of P.



Shmuel Onn

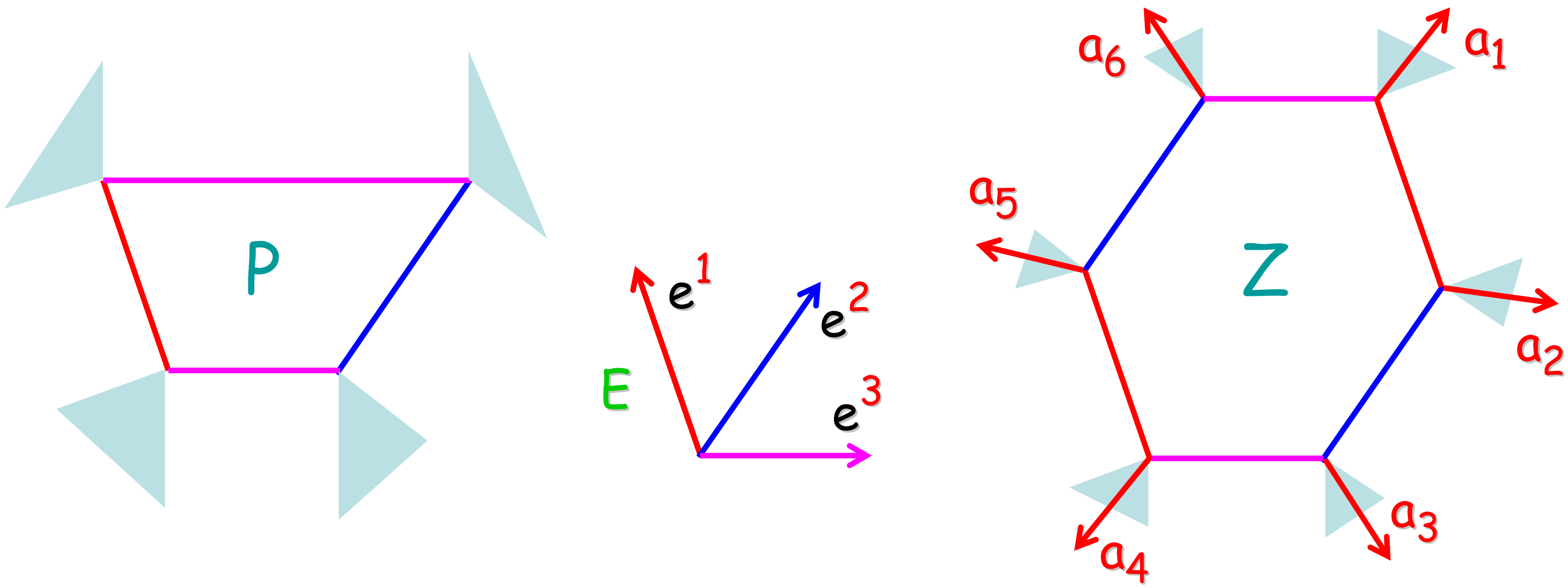
Proof: preliminaries on zonotopes

Lemma: If $E = \{e^1, \ldots, e^m\}$ covers all edge-directions of a polytope P then the zonotope $Z = [-1, 1]\, e^1 + \ldots + [-1, 1]\, e^m$ is a refinement of P.



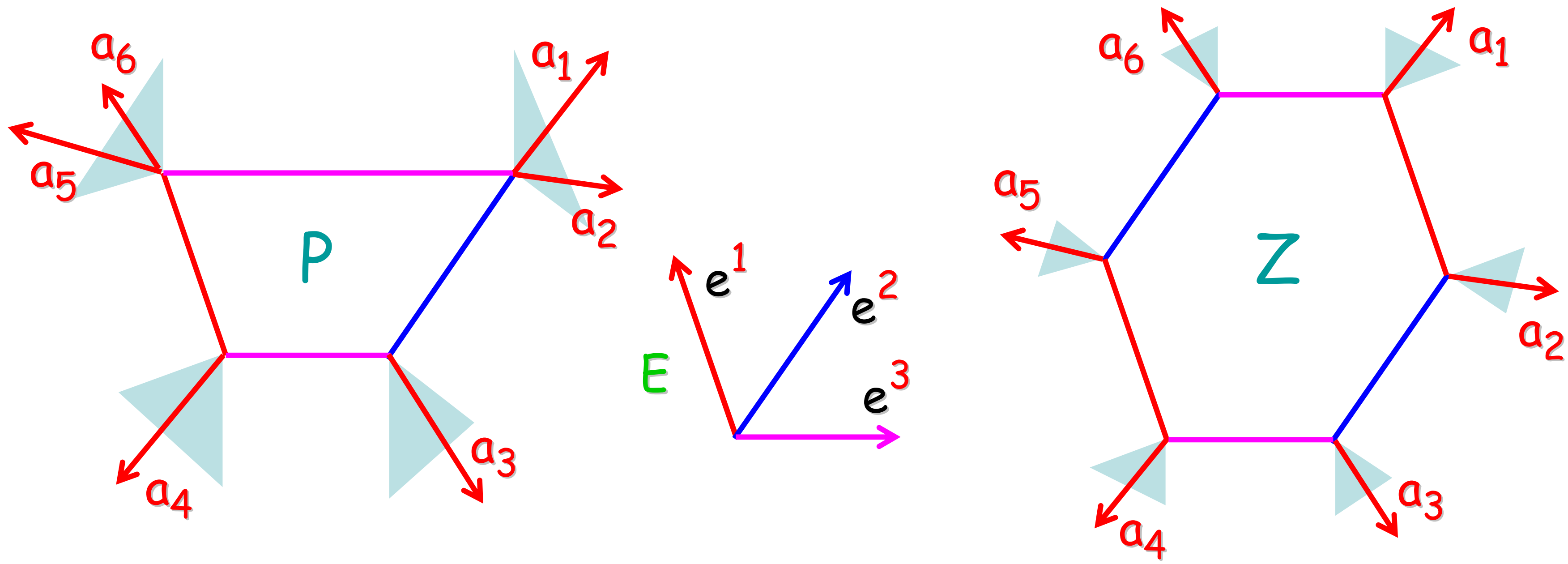
Proof: preliminaries on zonotopes

Lemma: If $E = \{e^1, \ldots, e^m\}$ covers all edge-directions of a polytope P then the zonotope $Z = [-1, 1]\, e^1 + \ldots + [-1, 1]\, e^m$ is a refinement of P.



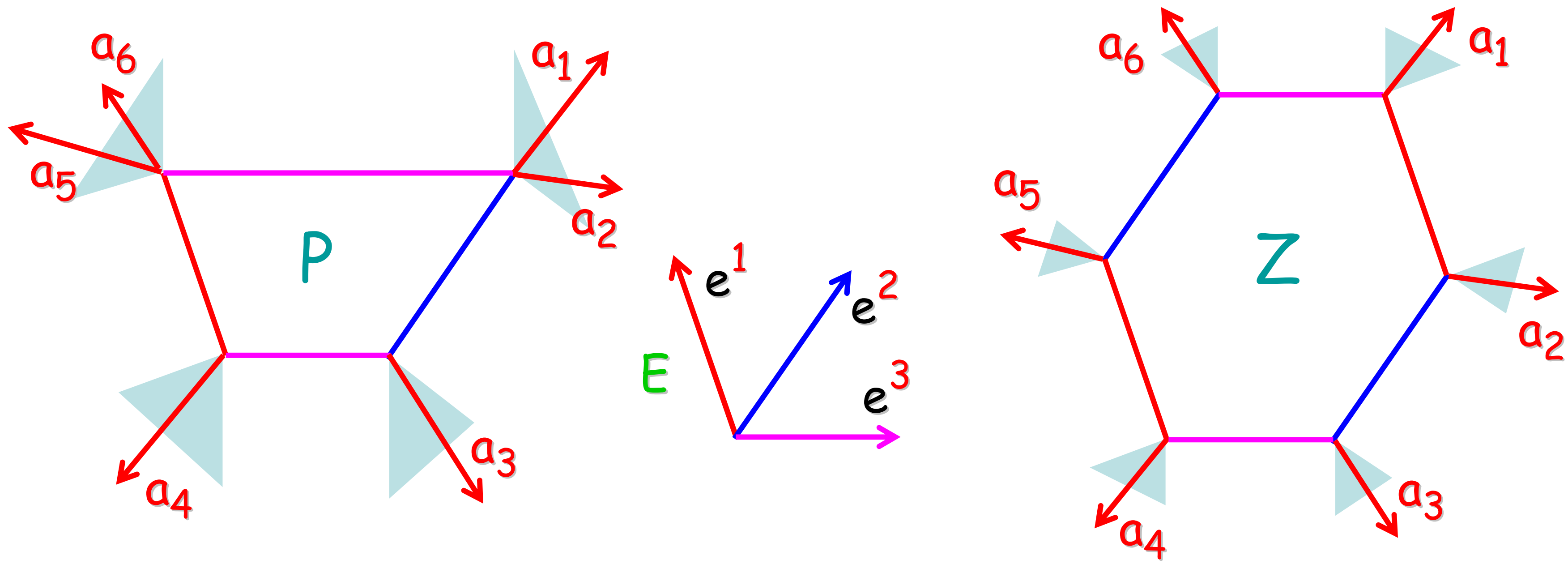
Proof: preliminaries on zonotopes

Lemma: If $E = \{e^1, \ldots, e^m\}$ covers all edge-directions of a polytope P then the zonotope $Z = [-1, 1]\, e^1 + \ldots + [-1, 1]\, e^m$ is a refinement of P.



Proof: preliminaries on zonotopes

Lemma: If $E = \{e^1, \ldots, e^m\}$ covers all edge-directions of a polytope P then the zonotope $Z = [-1, 1]\, e^1 + \ldots + [-1, 1]\, e^m$ is a refinement of P.



Lemma: In R^d, the zonotope Z can be constructed from $E = \{e^1, \ldots, e^m\}$ along with a vector a_i in the cone of every vertex in $O(m^{d-1})$ operations.

(Edelsbrunner, Gritzmann, Orourk, Seidel, Sharir, Sturmfels, …)

Shmuel Onn

Proof: the algorithm

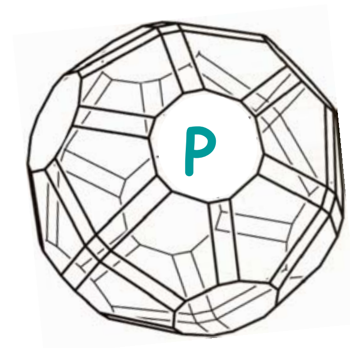
Input: S in Z^n given by linear optimization oracle, set E of edge-directions of P=conv{S}, d x n matrix w, and convex f on R^d given by comparison oracle

Shmuel Onn

Proof: the algorithm

Input: S in Z^n given by linear optimization oracle, set E of edge-directions of P=conv{S}, $d \times n$ matrix w, and convex f on R^d given by comparison oracle

E

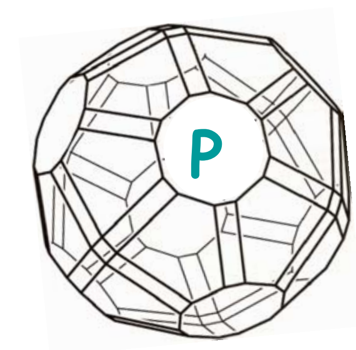


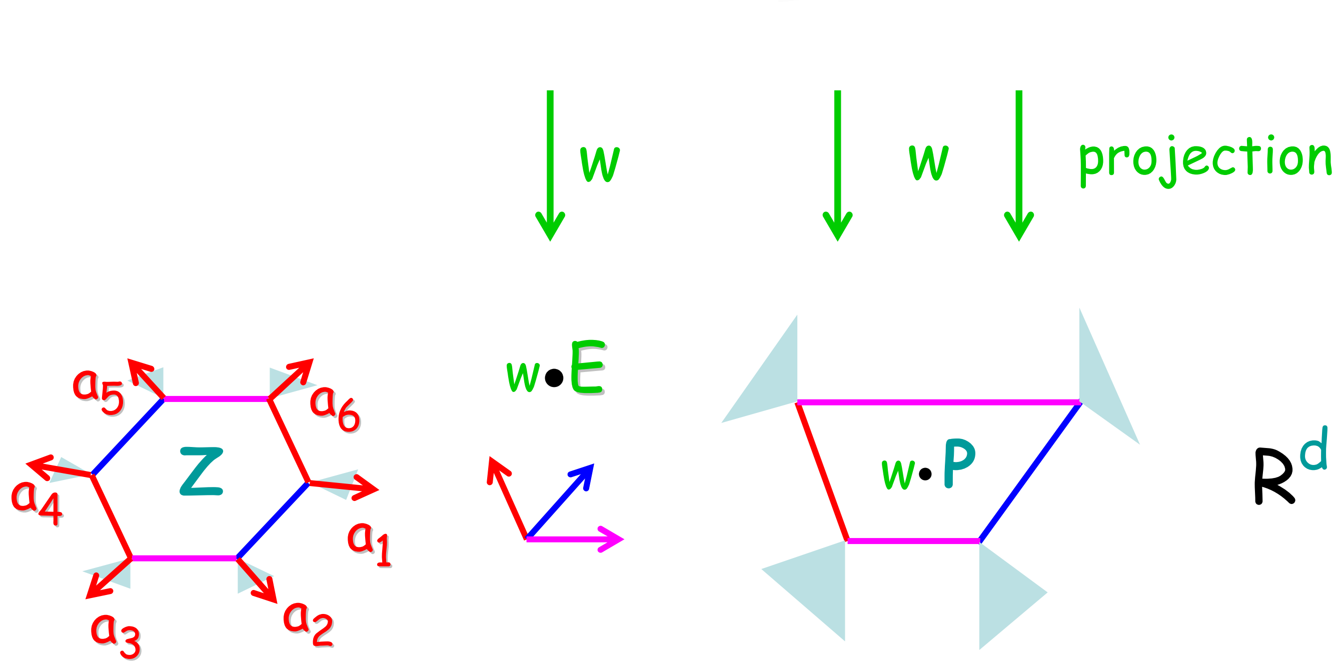
R^n

Proof: the algorithm

Input: S in Z^n given by linear optimization oracle, set E of edge-directions of $P=conv\{S\}$, $d \times n$ matrix w, and convex f on R^d given by comparison oracle

1. Construct the zonotope Z generated by the projection $w \bullet E$, and find a_i in each normal cone





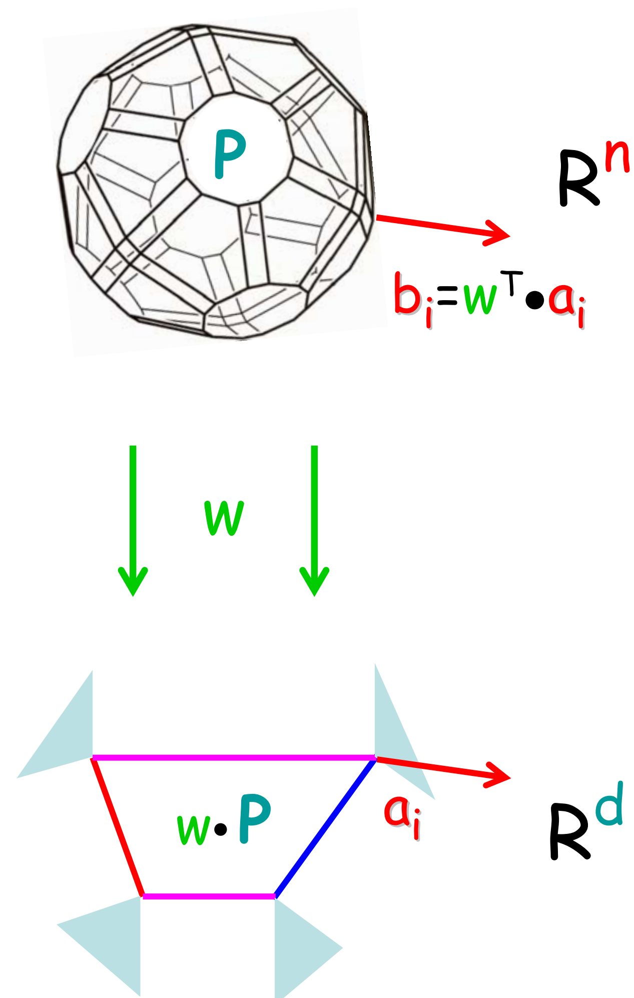
Shmuel Onn

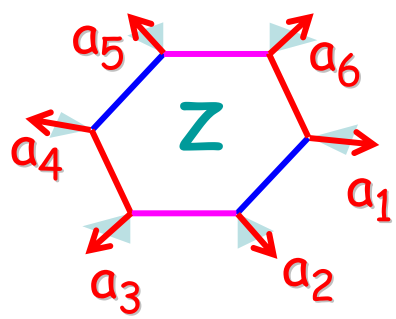
Proof: the algorithm

Input: S in Z^n given by linear optimization oracle, set E of edge-directions of P=conv{S}, d x n matrix w, and convex f on R^d given by comparison oracle

1. Construct the zonotope Z generated by the projection $w \bullet E$, and find a_i in each normal cone

2. Lift each a_i in R^d to $b_i = w^T \bullet a_i$ in R^n and solve linear optimization with objective b_i over S





Shmuel Onn

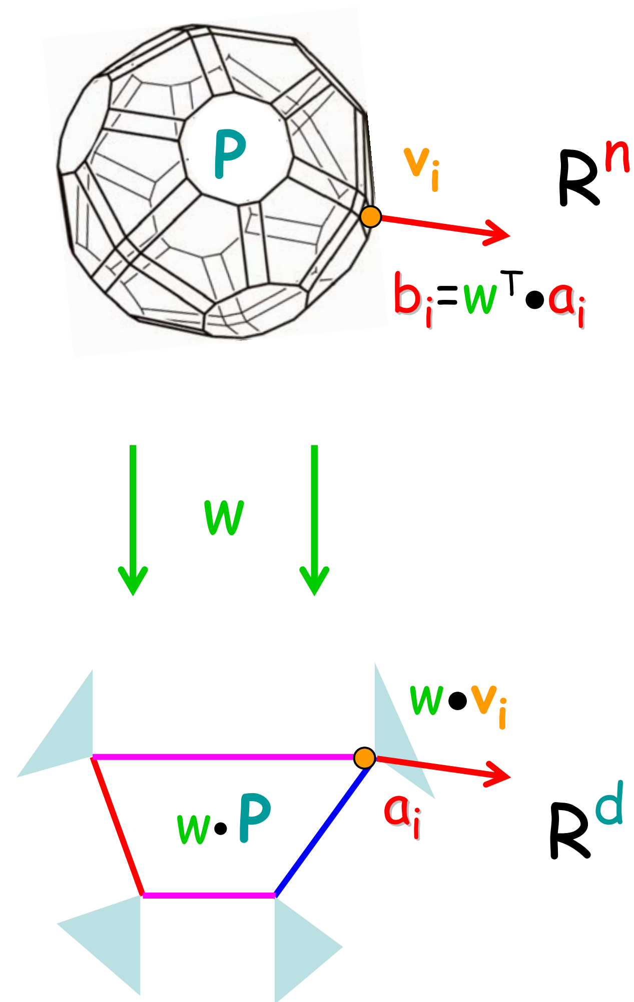
Proof: the algorithm

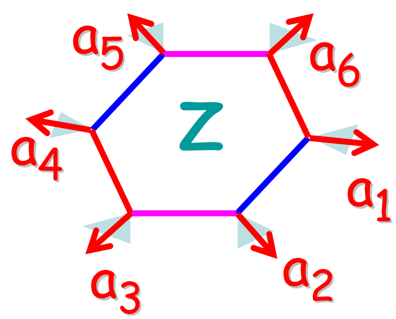
Input: S in Z^n given by linear optimization oracle, set E of edge-directions of P=conv{S}, d x n matrix w, and convex f on R^d given by comparison oracle

1. Construct the zonotope Z generated by the projection $w \bullet E$, and find a_i in each normal cone

2. Lift each a_i in R^d to $b_i = w^T \bullet a_i$ in R^n and solve linear optimization with objective b_i over S

3. Obtain the vertex v_i of P and the vertex $w \bullet v_i$ of $w \bullet P$





Shmuel Onn

Proof: the algorithm

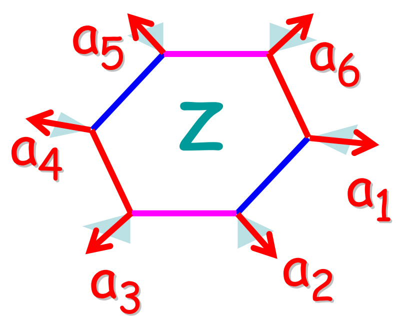
Input: S in Z^n given by linear optimization oracle, set E of edge-directions of $P=\text{conv}\{S\}$, $d \times n$ matrix w, and convex f on R^d given by comparison oracle

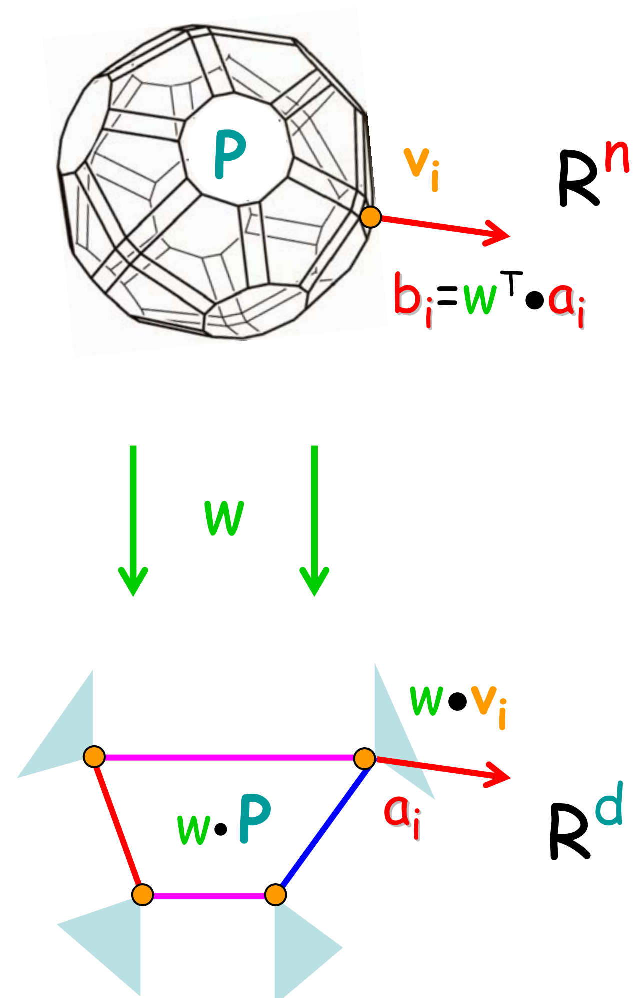
1. Construct the zonotope Z generated by the projection $w{\bullet}E$, and find a_i in each normal cone

2. Lift each a_i in R^d to $b_i = w^T{\bullet}\, a_i$ in R^n and solve linear optimization with objective b_i over S

3. Obtain the vertex v_i of P and the vertex $w{\bullet}v_i$ of $w{\bullet}P$

4. Output any v_i attaining maximum value $f(w{\bullet}\, v_i)$ using comparison oracle





Shmuel Onn

Strongly Polynomial
Convex Combinatorial Maximization

Theorem: For any fixed d, there is a strongly polynomial time algorithm that, given any S in $\{0,1\}^n$ presented by **membership oracle** and endowed with set E covering all edge-directions of conv$\{S\}$, and convex f, solves

$$\max \{ f(w_1 x, \ldots, w_d x) \ : \ x \text{ in } S \}$$

Shmuel Onn

Strongly Polynomial Convex Combinatorial Maximization

Theorem: For any fixed d, there is a strongly polynomial time algorithm that, given any S in $\{0,1\}^n$ presented by **membership oracle** and endowed with set E covering all edge-directions of $\text{conv}\{S\}$, and convex f, solves

$$\max \{ f(w_1 x, \ldots, w_d x) \ :\ x \text{ in } S \}$$

- Convex combinatorial optimization (Disc. Comp. Geom.)

Strongly Polynomial Convex Combinatorial Maximization

Theorem: For any fixed d, there is a strongly polynomial time algorithm that, given any S in $\{0,1\}^n$ presented by **membership oracle** and endowed with set E covering all edge-directions of conv$\{S\}$, and convex f, solves

$$\max \{ f(w_1 x, \ldots, w_d x) \; : \; x \text{ in } S \}$$

Natural case with exponentially many edge-directions – permutation matrices and Birkhoff polytope - is treated in

- Nonlinear bipartite matching (Disc. Opt.)

Proof: membership → augmentation → linear optimization

Consider maximizing wx over S, with E all edge-directions of $\text{conv}\{S\}$:

Proof: membership → augmentation → linear optimization

Consider maximizing wx over S, with E all edge-directions of conv$\{S\}$:

Lemma: Membership → Augmentation

Proof: x in S can be improved if and only if there is an edge direction e in E such that $w \cdot e > 0$ and $x + e = y$ for some y in S.

Proof: membership → augmentation → linear optimization

Consider maximizing wx over S, with E all edge-directions of conv{S}:

Lemma: Membership → Augmentation

Proof: x in S can be improved if and only if there is an edge direction e in E such that $w \cdot e > 0$ and $x + e = y$ for some y in S.

Lemma: Augmentation → linear Optimization

Proof: Schulz-Weismantel-Ziegler and Grötschel–Lovász using scaling ideas going back to Edmonds-Karp.

Proof: membership → augmentation → linear optimization

Consider maximizing wx over S, with E all edge-directions of conv{S}:

Lemma: Membership → Augmentation

Proof: x in S can be improved if and only if there is an edge direction e in E such that w•e > 0 and x + e = y for some y in S.

Lemma: Augmentation → linear Optimization

Proof: Schulz-Weismantel-Ziegler and Grötschel–Lovász using scaling ideas going back to Edmonds-Karp.

Lemma: Polynomial time → Strongly polynomial time

Proof: Frank-Tárdos show that using Diophantine approximation can replace w by w' of bit size depending polynomially only on n.

Some Algebraic Methods:

Nonlinear Integer Programming

Nonlinear Integer Programming

We now concentrate on the nonlinear integer programming problem:

$$\min \text{ or } \max \{f(w_1 x, \ldots, w_d x) : x \geq 0,\ Bx = b,\ x \text{ integer}\}$$

with $w_1 x, \ldots, w_d x$ linear forms on R^n and f real valued function on R^d.

N-Fold Systems

Shmuel Onn

N-Fold Systems

Let A be $(r+s) \times t$ matrix with submatrices A_1, A_2 of first r and last s rows.

N-Fold Systems

Let A be $(r+s) \times t$ matrix with submatrices A_1, A_2 of first r and last s rows.

Define the n-fold product of A to be the following $(r+ns) \times nt$ matrix,

$$A^{(n)} = \underbrace{\begin{pmatrix} A_1 & A_1 & A_1 & \cdots & A_1 \\ A_2 & 0 & 0 & \cdots & 0 \\ 0 & A_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A_2 \end{pmatrix}}_{n} .$$

N-Fold Systems

Let A be $(r+s) \times t$ matrix with submatrices A_1, A_2 of first r and last s rows.

Define the n-fold product of A to be the following $(r+ns) \times nt$ matrix,

$$A^{(n)} = \underbrace{\begin{pmatrix} A_1 & A_1 & A_1 & \cdots & A_1 \\ A_2 & 0 & 0 & \cdots & 0 \\ 0 & A_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A_2 \end{pmatrix}}_{n} .$$

We have the following four theorems on n-fold integer programming:

N-Fold Systems

Let A be $(r+s) \times t$ matrix with submatrices A_1, A_2 of first r and last s rows.

Define the n-fold product of A to be the following $(r+ns) \times nt$ matrix,

$$A^{(n)} = \underbrace{\begin{pmatrix} A_1 & A_1 & A_1 & \cdots & A_1 \\ A_2 & 0 & 0 & \cdots & 0 \\ 0 & A_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A_2 \end{pmatrix}}_{n} .$$

We have the following four theorems on n-fold integer programming:

Theorem 1: For any fixed matrix A, **linear integer programming** over n-fold products of A can be done in polynomial time:

$$\max \{ wx : \ A^{(n)}x = a,\ x \geq 0,\ x \text{ integer} \}$$

N-Fold Systems

Let A be $(r+s) \times t$ matrix with submatrices A_1, A_2 of first r and last s rows.

Define the n-fold product of A to be the following $(r+ns) \times nt$ matrix,

$$A^{(n)} = \underbrace{\begin{pmatrix} A_1 & A_1 & A_1 & \cdots & A_1 \\ A_2 & 0 & 0 & \cdots & 0 \\ 0 & A_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A_2 \end{pmatrix}}_{n} .$$

We have the following four theorems on n-fold integer programming:

Theorem 2: For any fixed d and matrix A, **convex integer maximization** over n-fold products of A can be done in polynomial time:

$$\max \{ f(w_1 x, \ldots, w_d x) \ : \ A^{(n)} x = a,\ x \geq 0,\ x \text{ integer} \}$$

N-Fold Systems

Let A be $(r+s) \times t$ matrix with submatrices A_1, A_2 of first r and last s rows.

Define the **n-fold product** of A to be the following $(r+ns) \times nt$ matrix,

$$A^{(n)} = \underbrace{\begin{pmatrix} A_1 & A_1 & A_1 & \cdots & A_1 \\ A_2 & 0 & 0 & \cdots & 0 \\ 0 & A_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A_2 \end{pmatrix}}_{n} .$$

We have the following four theorems on n-fold integer programming:

Theorem 3: For fixed matrix A, **separable convex integer minimization** over n-fold products of A can be done in polynomial time:

$$\min \{ f_1(x_1) + \ldots + f_{nt}(x_{nt}) \ : \ A^{(n)}x = a,\ x \geq 0,\ x \text{ integer} \}$$

Shmuel Onn

N-Fold Systems

Let A be $(r+s) \times t$ matrix with submatrices A_1, A_2 of first r and last s rows.

Define the n-fold product of A to be the following $(r+ns) \times nt$ matrix,

$$A^{(n)} = \underbrace{\begin{pmatrix} A_1 & A_1 & A_1 & \cdots & A_1 \\ A_2 & 0 & 0 & \cdots & 0 \\ 0 & A_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A_2 \end{pmatrix}}_{n} .$$

We have the following four theorems on n-fold integer programming:

Theorem 4: For fixed matrix A, **integer point l_p-nearest to a given x** over n-fold products of A can be determined in polynomial time:

$$\min \{ \, |x - \mathbf{x}|_p \; : \; A^{(n)}x = a, \; x \geq 0, \; x \text{ integer} \}$$

Shmuel Onn

N-Fold Systems

Let A be $(r+s) \times t$ matrix with submatrices A_1, A_2 of first r and last s rows.

Define the n-fold product of A to be the following $(r+ns) \times nt$ matrix,

$$A^{(n)} = \underbrace{\begin{pmatrix} A_1 & A_1 & A_1 & \cdots & A_1 \\ A_2 & 0 & 0 & \cdots & 0 \\ 0 & A_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A_2 \end{pmatrix}}_{n} .$$

- N-fold integer programming (Disc. Opt. in memory of Dantzig)
- Convex integer maximization via Graver bases (J. Pure App. Algebra)
- Convex integer minimization (submitted)

Universality of N-Fold Systems

Shmuel Onn

Universality of N-Fold Systems

Define the n-product of A as the following variant of the n-fold operator:

$$A^{[n]} = \underbrace{\begin{pmatrix} I & I & I & \cdots & I \\ A & 0 & 0 & \cdots & 0 \\ 0 & A & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A \end{pmatrix}}_{n} \quad .$$

Shmuel Onn

Universality of N-Fold Systems

Define the n-product of A as the following variant of the n-fold operator:

$$A^{[n]} = \underbrace{\begin{pmatrix} I & I & I & \cdots & I \\ A & 0 & 0 & \cdots & 0 \\ 0 & A & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A \end{pmatrix}}_{n} .$$

Consider m-products of the 1×3 matrix $(1 \;\; 1 \;\; 1)$. For instance,

Universality of N-Fold Systems

Define the n-product of A as the following variant of the n-fold operator:

$$A^{[n]} = \underbrace{\begin{pmatrix} I & I & I & \cdots & I \\ A & 0 & 0 & \cdots & 0 \\ 0 & A & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A \end{pmatrix}}_{n} .$$

Consider m-products of the 1×3 matrix $(1\ \ 1\ \ 1)$. For instance,

$$(1\ \ 1\ \ 1)^{[3]} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} .$$

Universality of N-Fold Systems

Define the n-product of A as the following variant of the n-fold operator:

$$A^{[n]} = \begin{pmatrix} I & I & I & \cdots & I \\ A & 0 & 0 & \cdots & 0 \\ 0 & A & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A \end{pmatrix} .$$

Universality Theorem: Any bounded set $\{x \text{ integer} : x \geq 0,\ Bx = b\}$ is in polynomial-time-computable coordinate-embedding-bijection with some

$$\{x \text{ integer} : x \geq 0,\ (1\ \ 1\ \ 1)^{[m][n]} x = a\}$$

Universality of N-Fold Systems

Define the n-product of A as the following variant of the n-fold operator:

$$A^{[n]} = \begin{pmatrix} I & I & I & \cdots & I \\ A & 0 & 0 & \cdots & 0 \\ 0 & A & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A \end{pmatrix} .$$

Universality Theorem: Any bounded set $\{x \text{ integer} : x \geq 0,\ Bx = b\}$ is in polynomial-time-computable coordinate-embedding-bijection with some

$$\{x \text{ integer} : x \geq 0,\ (1\ \ 1\ \ 1)^{[m][n]} x = a\}$$

- All linear and integer programs are slim 3-way programs (SIAM Opt.)

Shmuel Onn

Universality of N-Fold Systems

Define the n-product of A as the following variant of the n-fold operator:

$$A^{[n]} = \begin{pmatrix} I & I & I & \cdots & I \\ A & 0 & 0 & \cdots & 0 \\ 0 & A & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A \end{pmatrix} .$$

Universality Theorem: Any bounded set $\{x \text{ integer} : x \geq 0,\ Bx = b\}$ is in polynomial-time-computable coordinate-embedding-bijection with some

$$\{x \text{ integer} : x \geq 0,\ (1\ \ 1\ \ 1)^{[m][n]}x = a\}$$

Scheme for Nonlinear Integer Programming:

any program: $\max \{f(w_1 x, \ldots, w_d x) : x \geq 0,\ Bx = b,\ x \text{ integer}\}$

can be lifted to

n-fold program: $\max \{f(w'_1 x, \ldots, w'_d x) : x \geq 0,\ (1\ 1\ 1)^{[m][n]}x = a,\ x \text{ integer}\}$

Shmuel Onn

The Computational Complexity of Nonlinear Integer Programming:

The Graver Complexity of $K_{3,m}$

Graver Bases

Shmuel Onn

Graver Bases

Vector x conforms to y if lie in same orthant and $|x_i| \leq |y_i|$ and for all i.

Shmuel Onn

Graver Bases

Vector x conforms to y if lie in same orthant and $|x_i| \leq |y_i|$ and for all i.

The Graver basis of an integer matrix A is the finite set $G(A)$ of conformal-minimal nonzero integer vectors x satisfying $Ax = 0$.

Graver Bases

Vector x conforms to y if lie in same orthant and $|x_i| \le |y_i|$ and for all i.

The Graver basis of an integer matrix A is the finite set $G(A)$ of conformal-minimal nonzero integer vectors x satisfying $Ax = 0$.

Example: Consider $A=(1\ 2\ 1)$.

Graver Bases

Vector x conforms to y if lie in same orthant and $|x_i| \leq |y_i|$ and for all i.

The Graver basis of an integer matrix A is the finite set $G(A)$ of conformal-minimal nonzero integer vectors x satisfying $Ax = 0$.

Example: Consider $A=(1\ 2\ 1)$. Then $G(A)$ consists of:

Graver Bases

Vector x conforms to y if lie in same orthant and $|x_i| \leq |y_i|$ and for all i.

The Graver basis of an integer matrix A is the finite set $G(A)$ of conformal-minimal nonzero integer vectors x satisfying $Ax = 0$.

Example: Consider $A=(1\ 2\ 1)$. Then $G(A)$ consists of:

circuits: $\pm(2\ -1\ 0)$

Graver Bases

Vector x conforms to y if lie in same orthant and $|x_i| \le |y_i|$ and for all i.

The Graver basis of an integer matrix A is the finite set $G(A)$ of conformal-minimal nonzero integer vectors x satisfying $Ax = 0$.

Example: Consider $A=(1\ 2\ 1)$. Then $G(A)$ consists of:

circuits: $\pm(2\ {-1}\ 0)$, $\pm(1\ 0\ {-1})$

Shmuel Onn

Graver Bases

Vector x conforms to y if lie in same orthant and $|x_i| \leq |y_i|$ and for all i.

The Graver basis of an integer matrix A is the finite set $G(A)$ of conformal-minimal nonzero integer vectors x satisfying $Ax = 0$.

Example: Consider A=(1 2 1). Then $G(A)$ consists of:

circuits: ±(2 -1 0), ±(1 0 -1), ±(0 1 -2)

Shmuel Onn

Graver Bases

Vector x conforms to y if lie in same orthant and $|x_i| \leq |y_i|$ and for all i.

The Graver basis of an integer matrix A is the finite set $G(A)$ of conformal-minimal nonzero integer vectors x satisfying $Ax = 0$.

Example: Consider A=(1 2 1). Then $G(A)$ consists of:

circuits: ±(2 -1 0), ±(1 0 -1), ±(0 1 -2)

non-circuits: ±(1 -1 1)

Shmuel Onn

Graver Bases of N-Products

Consider n-products of A:

$$A^{[n]} = \begin{pmatrix} I & I & I & \cdots & I \\ A & 0 & 0 & \cdots & 0 \\ 0 & A & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A \end{pmatrix} .$$

Graver Bases of N-Products

Consider n-products of A:

$$A^{[n]} = \begin{pmatrix} I & I & I & \cdots & I \\ A & 0 & 0 & \cdots & 0 \\ 0 & A & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A \end{pmatrix} .$$

The type of $x = (x^1, \ldots, x^n)$ is the number of its nonzero blocks.

Graver Bases of N-Products

Consider n-products of A:

$$A^{[n]} = \begin{pmatrix} I & I & I & \cdots & I \\ A & 0 & 0 & \cdots & 0 \\ 0 & A & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A \end{pmatrix} .$$

The type of $x = (x^1, \ldots, x^n)$ is the number of its nonzero blocks.

Lemma: Every integer matrix A has a finite Graver complexity $g(A)$ such that any element in the Graver basis $G(A^{[n]})$ for any n has type $\leq g(A)$.

(Aoki-Takemura, Santos-Sturmfels, Hosten-Sullivant)

Graver Bases of N-Products

Consider n-products of A:

$$A^{[n]} = \begin{pmatrix} I & I & I & \cdots & I \\ A & 0 & 0 & \cdots & 0 \\ 0 & A & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A \end{pmatrix} .$$

The type of $x = (x^1, \ldots, x^n)$ is the number of its nonzero blocks.

Lemma: Every integer matrix A has a finite Graver complexity $g(A)$ such that any element in the Graver basis $G(A^{[n]})$ for any n has type $\leq g(A)$.

Theorem: Graver bases of $A^{[n]}$ and $A^{(n)}$ are computable in polynomial time.

Graver Bases of N-Products

Consider n-products of A:

$$A^{[n]} = \begin{pmatrix} I & I & I & \cdots & I \\ A & 0 & 0 & \cdots & 0 \\ 0 & A & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A \end{pmatrix} .$$

The type of $x = (x^1, \ldots, x^n)$ is the number of its nonzero blocks.

Lemma: Every integer matrix A has a finite Graver complexity $g(A)$ such that any element in the Graver basis $G(A^{[n]})$ for any n has type $\leq g(A)$.

Theorem: Graver bases of $A^{[n]}$ and $A^{(n)}$ are computable in polynomial time.

Practicality: the complexity is high and dominated by $n^{g(A)}$, but promising scheme is to consider Graver elements of gradually increasing type.

The Graver Complexity of a Graph

Shmuel Onn

The Graver Complexity of a Graph

Definition: The Graver complexity of a graph or a digraph G is the Graver complexity $g(G):=g(A)$ of its vertex-edge incidence matrix A.

The Graver Complexity of a Graph

Definition: The Graver complexity of a graph or a digraph G is the Graver complexity $g(G):=g(A)$ of its vertex-edge incidence matrix A.

Let $g(m) := g(K_{3,m}) = g((1\ 1\ 1)^{[m]})$.

The Graver Complexity of a Graph

Definition: The Graver complexity of a graph or a digraph G is the Graver complexity $g(G) := g(A)$ of its vertex-edge incidence matrix A.

Let $g(m) := g(K_{3,m}) = g((1\ 1\ 1)^{[m]})$.

For instance,

$$g(3) = g\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} = 9$$

The Graver Complexity of a Graph

Recall: $g(m) := g(K_{3,m}) = g((1\ 1\ 1)^{[m]})$

Theorem: Consider the following **universal nonlinear integer program**:

$$\max \{f(w_1 x, \ldots, w_d x) : x \geq 0,\ (1\ 1\ 1)^{[m][n]} x = a,\ x \text{ integer}\}$$

The Graver Complexity of a Graph

Recall: $g(m) := g(K_{3,m}) = g((1\ 1\ 1)^{[m]})$

Theorem: Consider the following **universal nonlinear integer program**:

$$\max \{f(w_1 x, \ldots, w_d x) : x \geq 0,\ (1\ 1\ 1)^{[m][n]} x = a,\ x \text{ integer}\}$$

1. For any fixed m can solve it in polynomial time $n^{g(m)}$

The Graver Complexity of a Graph

Recall: $g(m) := g(K_{3,m}) = g((1\ 1\ 1)^{[m]})$

Theorem: Consider the following **universal nonlinear integer program**:

$$\max \{f(w_1 x, \ldots, w_d x) : x \geq 0,\ (1\ 1\ 1)^{[m][n]} x = a,\ x \text{ integer}\}$$

1. For any fixed m can solve it in polynomial time $n^{g(m)}$

2. For variable m it is universal and NP-hard

Shmuel Onn

The Graver Complexity of a Graph

Recall: $g(m) := g(K_{3,m}) = g((1\ 1\ 1)^{[m]})$

Theorem: Consider the following **universal nonlinear integer program**:

$$\max \{f(w_1 x, \ldots, w_d x) : x \geq 0,\ (1\ 1\ 1)^{[m][n]} x = a,\ x \text{ integer}\}$$

1. For any fixed m can solve it in polynomial time $n^{g(m)}$

2. For variable m it is universal and NP-hard

So if P≠NP then $g(m)$ must grow with m. How fast ?

Shmuel Onn

The Graver Complexity of a Graph

Recall: $g(m) := g(K_{3,m}) = g((1\ 1\ 1)^{[m]})$

Theorem: We have $\Omega(2^m) = g(m) = O(m^4 6^m)$.

Shmuel Onn

The Graver Complexity of a Graph

Recall: $g(m) := g(K_{3,m}) = g((1\ 1\ 1)^{[m]})$

Theorem: We have $\Omega(2^m) = g(m) = O(m^4 6^m)$.

Also: $g(2)=3$, $g(3)=9$, $g(4) \geq 27$, $g(5) \geq 61$.

The Graver Complexity of a Graph

Recall: $g(m) := g(K_{3,m}) = g((1\ 1\ 1)^{[m]})$

Theorem: We have $\Omega(\ 2^m\ ) = g(m) = O(\ m^4 6^m\ )$.

Also: $g(2)=3$, $g(3)=9$, $g(4) \geq 27$, $g(5) \geq 61$.

- Graver complexity of integer programming (Annals Combin.)

Proofs of Theorems 1-4

about

Nonlinear N-Fold Integer Programming

Proof of Theorem 1 - Linear Optimization

Shmuel Onn

Proof of Theorem 1 - Linear Optimization

- Compute the Graver basis of $A^{(n)}$ efficiently.

Proof of Theorem 1 - Linear Optimization

- Compute the Graver basis of $A^{(n)}$ efficiently.

- Use integer caratheodory theorem and convergence property to show that $G(A^{(n)})$ allows to augment feasible to optimal point efficiently.

Proof of Theorem 1 - Linear Optimization

- Compute the Graver basis of $A^{(n)}$ efficiently.

- Use integer caratheodory theorem and convergence property to show that $G(A^{(n)})$ allows to augment feasible to optimal point efficiently.

- Use an auxiliary n-fold program to find an initial feasible point.

Proof of Theorem 2 - Convex Maximization

Proof of Theorem 2 - Convex Maximization

- Compute the Graver basis of $A^{(n)}$ efficiently.

Proof of Theorem 2 – Convex Maximization

- Compute the Graver basis of $A^{(n)}$ efficiently.

- Simulate linear optimization oracle using Theorem 1.

Proof of Theorem 2 – Convex Maximization

- Compute the Graver basis of $A^{(n)}$ efficiently.

- Simulate linear optimization oracle using Theorem 1.

- Reduce convex to linear maximization using the Geometric Theorem with the Graver basis $G(A^{(n)})$ providing a set of edge-directions of

$$\text{conv}\{x : A^{(n)}x = a,\ x \geq 0,\ x \text{ integer}\}$$

Proof of Theorem 3 - Separable Convex Minimization

Proof of Theorem 3 – Separable Convex Minimization

- Compute Graver basis of n-fold product of auxiliary larger matrix.

Proof of Theorem 3 – Separable Convex Minimization

- Compute Graver basis of n-fold product of auxiliary larger matrix.

- Use integer caratheodory, convergence property, and superadditivity, and use the Graver basis to augment feasible to optimal point.

Proof of Theorem 3 – Separable Convex Minimization

- Compute Graver basis of n-fold product of auxiliary larger matrix.

- Use integer caratheodory, convergence property, and superadditivity, and use the Graver basis to augment feasible to optimal point.

- Use an auxiliary n-fold program to find an initial feasible point.

Proof of Theorem 4 - l_p-Nearest Point

Shmuel Onn

Proof of Theorem 4 - l_p-Nearest Point

- For positive integer p this is the special case of Theorem 3 with

$$(|x - \mathbf{x}|_p)^p = |x_1 - \mathbf{x}_1|^p + \ldots + |x_{nt} - \mathbf{x}_{nt}|^p$$

Proof of Theorem 4 - l_p-Nearest Point

- For positive integer p this is the special case of Theorem 3 with

$$(|x - \mathbf{x}|_p)^p = |x_1 - \mathbf{x}_1|^p + \ldots + |x_{nt} - \mathbf{x}_{nt}|^p$$

- For p infinity this reduces to the case of finite q for suitably large q.

Shmuel Onn

Multiway Tables and Applications

Multiway Tables and Margins

Shmuel Onn

Multiway Tables and Margins

A k-way table is an $m_1 \times \cdots \times m_k$ array of nonnegative integers.

Shmuel Onn

Multiway Tables and Margins

A k-way table is an $m_1 \times \cdots \times m_k$ array of nonnegative integers.

A margin of a table is the sum of all entries in some flat of the table, so can be a line-sum, plane-sum, and so on.

Multiway Tables and Margins

A k-way table is an $m_1 \times \cdots \times m_k$ array of nonnegative integers.

A margin of a table is the sum of all entries in some flat of the table, so can be a line-sum, plane-sum, and so on.

Example: 2-way table of size 2 x 3:

0	1	2
2	2	0

Shmuel Onn

Multiway Tables and Margins

A k-way table is an $m_1 \times \cdots \times m_k$ array of nonnegative integers.
A margin of a table is the sum of all entries in some flat of the table, so can be a line-sum, plane-sum, and so on.

Example: 2-way table of size 2×3 with line-sums:

0	1	2	3
2	2	0	

Multiway Tables and Margins

A k-way table is an $m_1 \times \cdots \times m_k$ array of nonnegative integers.

A margin of a table is the sum of all entries in some flat of the table, so can be a line-sum, plane-sum, and so on.

Example: 2-way table of size 2×3 with line-sums:

0	1	2	3
2	2	0	
2			

Multiway Tables and Margins

A k-way table is an $m_1 \times \cdots \times m_k$ array of nonnegative integers.

A margin of a table is the sum of all entries in some flat of the table, so can be a line-sum, plane-sum, and so on.

Example: 2-way table of size 2×3 with line-sums:

0	1	2	3
2	2	0	4
2	3	2	

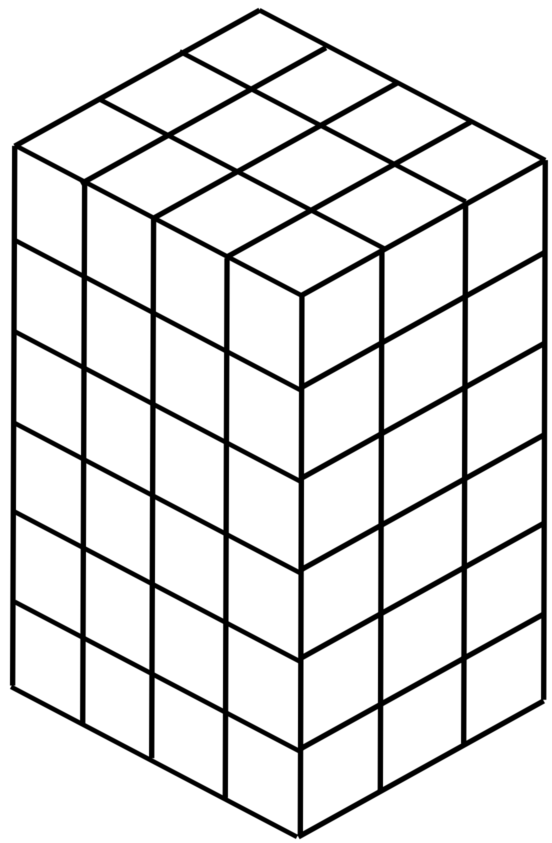
Shmuel Onn

Multiway Tables and Margins

A k-way table is an $m_1 \times \cdots \times m_k$ array of nonnegative integers.

A margin of a table is the sum of all entries in some flat of the table, so can be a line-sum, plane-sum, and so on.

Example: 3-way table of size $6 \times 4 \times 3$:

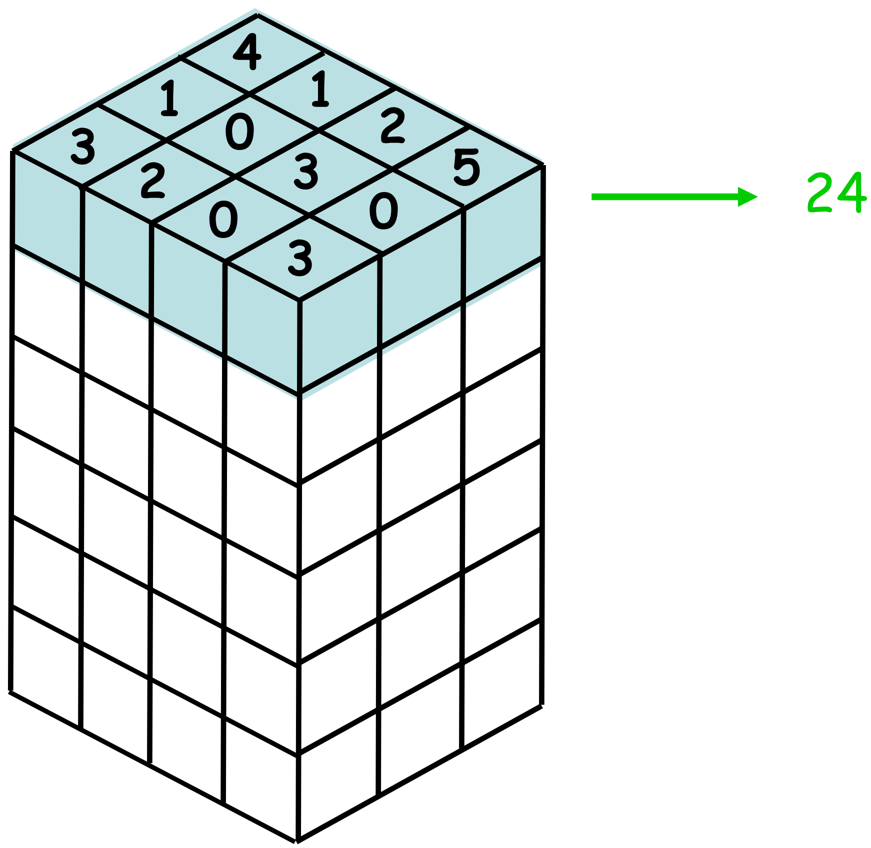


Shmuel Onn

Multiway Tables and Margins

A k-way table is an $m_1 \times \cdots \times m_k$ array of nonnegative integers.
A margin of a table is the sum of all entries in some flat of the table, so can be a line-sum, plane-sum, and so on.

Example: 3-way table of size $6 \times 4 \times 3$ with a plane-sum:



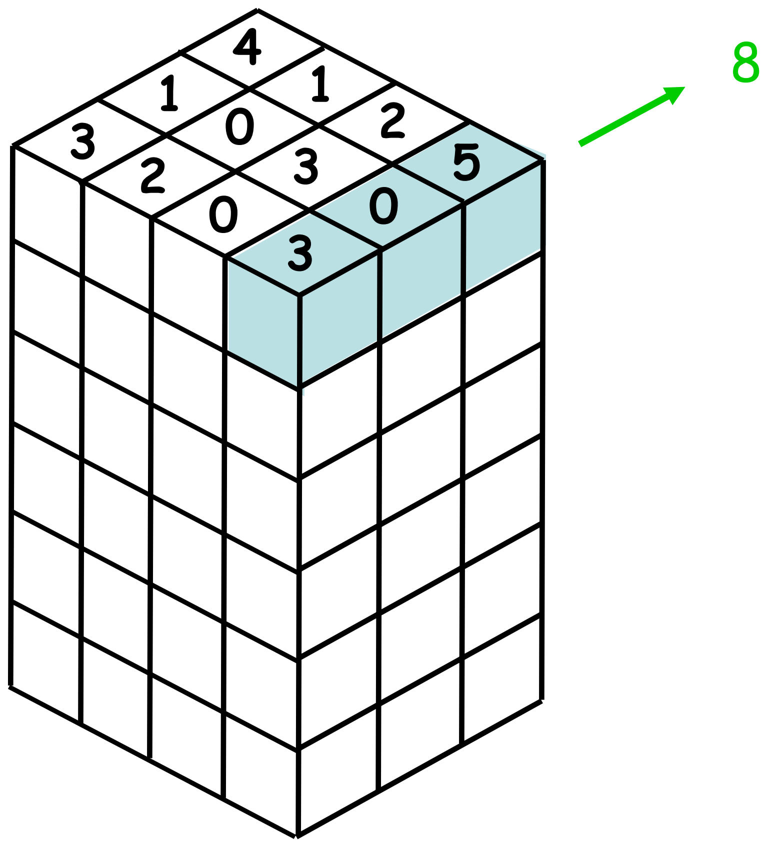
Shmuel Onn

Multiway Tables and Margins

A k-way table is an $m_1 \times \cdots \times m_k$ array of nonnegative integers.

A margin of a table is the sum of all entries in some flat of the table, so can be a line-sum, plane-sum, and so on.

Example: 3-way table of size $6 \times 4 \times 3$ with a line-sum:



Shmuel Onn

Casting "Long" Multiway Tables as N-fold Programs

Shmuel Onn

Casting "Long" Multiway Tables as N-fold Programs

The margin equations for $m_1 \times \cdots \times m_k \times n$ tables form an n-fold system defined by a suitable $(r+s) \times t$ matrix A, where A_1 controls the equations of margins involving summation over layers, whereas A_2 controls the equations of margins involving summation within a single layer at a time.

Casting "Long" Multiway Tables as N-fold Programs

The margin equations for $m_1 \times \cdots \times m_k \times n$ tables form an n-fold system defined by a suitable $(r+s) \times t$ matrix A, where A_1 controls the equations of margins involving summation over layers, whereas A_2 controls the equations of margins involving summation within a single layer at a time.

$$A^{(n)} = \underbrace{\begin{pmatrix} A_1 & A_1 & A_1 & \cdots & A_1 \\ A_2 & 0 & 0 & \cdots & 0 \\ 0 & A_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A_2 \end{pmatrix}}_{n} .$$

$$A^{(n)} x = a, \qquad x = (x^1, \ldots, x^n), \qquad a = (a^0, a^1, \ldots, a^n)$$

$$A_1(x^1 + \ldots + x^n) = a^0, \qquad A_2 x^k = a^k, \quad k = 1, \ldots n$$

Casting "Long" Multiway Tables as N-fold Programs

The margin equations for $m_1 \times \cdots \times m_k \times n$ tables form an n-fold system defined by a suitable $(r+s) \times t$ matrix A, where A_1 controls the equations of margins involving summation over layers, whereas A_2 controls the equations of margins involving summation within a single layer at a time.

Example:

The line-sum equations for $3 \times 3 \times n$ tables are defined by the $(9+6) \times 9$ matrix A where A_1 is the 9×9 identity matrix and

$$A_2 = (1 \;\; 1 \;\; 1)^{[3]} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} .$$

Long Tables are Efficiently Treatable

Corollary: nonlinear optimization over $m_1 \times \cdots \times m_k \times n$ tables with hierarchical margin constraints can be done in polynomial time.

Privacy and Confidentiality in Statistical Data Bases: Entry Uniqueness Problem

Shmuel Onn

Privacy and Confidentiality in Statistical Data Bases: Entry Uniqueness Problem

Agencies allow web-access to information on their data bases
but are concerned about confidentiality of individuals.

Shmuel Onn

Privacy and Confidentiality in Statistical Data Bases: Entry Uniqueness Problem

Agencies allow web-access to information on their data bases
but are concerned about confidentiality of individuals.

Common strategy: disclose margins but not table entries.

Shmuel Onn

Privacy and Confidentiality in Statistical Data Bases: Entry Uniqueness Problem

Agencies allow web-access to information on their data bases but are concerned about confidentiality of individuals.

Common strategy: disclose margins but not table entries.

If the value of an entry is the same in all tables with the disclosed margins then that entry is **vulnerable**; otherwise it is **secure**.

Shmuel Onn

Privacy and Confidentiality in Statistical Data Bases: Entry Uniqueness Problem

Agencies allow web-access to information on their data bases but are concerned about confidentiality of individuals.

Common strategy: disclose margins but not table entries.

If the value of an entry is the same in all tables with the disclosed margins then that entry is **vulnerable**; otherwise it is **secure**.

Corollary: For $m_1 \times \cdots \times m_k \times n$ tables can check entry uniqueness in polynomial time and refrain from margin disclosure if vulnerable.

Shmuel Onn

Congestion Avoiding Transportation

Shmuel Onn

Congestion Avoiding Transportation

We wish to find minimum cost transportation or routing subject to demand and supply constraints and channel capacity constraints.

Congestion Avoiding Transportation

We wish to find minimum cost transportation or routing subject to demand and supply constraints and channel capacity constraints.

The classical approach assumes fixed channel cost per unit flow. But due to **channel congestion** when subject to heavy traffic or communication load, the delay and cost are better approximated by a nonlinear function of the flow such as

$$f(x) = \sum_{ij} f_{ij}(x_{ij}) = \sum_{ij} c_{ij}|x_{ij}|^{a_{ij}}$$

Congestion Avoiding Transportation

We wish to find minimum cost transportation or routing subject to demand and supply constraints and channel capacity constraints.

The classical approach assumes fixed channel cost per unit flow. But due to **channel congestion** when subject to heavy traffic or communication load, the delay and cost are better approximated by a nonlinear function of the flow such as

$$f(x) = \sum_{ij} f_{ij}(x_{ij}) = \sum_{ij} c_{ij}|x_{ij}|^{a_{ij}}$$

In the high dimensional extension, wish to find an optimal multiway table satisfying margin constraints and upper bounds on the entries.

Congestion Avoiding Transportation

We wish to find minimum cost transportation or routing subject to demand and supply constraints and channel capacity constraints.

The classical approach assumes fixed channel cost per unit flow. But due to **channel congestion** when subject to heavy traffic or communication load, the delay and cost are better approximated by a nonlinear function of the flow such as

$$f(x) = \sum_{ij} f_{ij}(x_{ij}) = \sum_{ij} c_{ij}|x_{ij}|^{a_{ij}}$$

In the high dimensional extension, wish to find an optimal multiway table satisfying margin constraints and upper bounds on the entries.

Corollary: We can find congestion avoiding transportation over $m_1 \times \cdots \times m_k \times n$ tables in polynomial time.

Shmuel Onn

Error Correcting Codes

Shmuel Onn

Error Correcting Codes

We wish to transmit a message X on a noisy channel. The message is augmented with some "check-sums" to allow for error correction.

Shmuel Onn

Error Correcting Codes

We wish to transmit a message X on a noisy channel. The message is augmented with some "check-sums" to allow for error correction.

Multiway tables provide an appealing way of organizing the check sum protocol. The sender transmits the message as a multiway table augmented with slack entries summing up to pre-agreed margins.

Shmuel Onn

Error Correcting Codes

We wish to transmit a message X on a noisy channel. The message is augmented with some "check-sums" to allow for error correction.

Multiway tables provide an appealing way of organizing the check sum protocol. The sender transmits the message as a multiway table augmented with slack entries summing up to pre-agreed margins.

The receiver gets a distorted table X and reconstructs the message as that table X with the pre-agreed margins that is l_p-nearest to X.

Shmuel Onn

Error Correcting Codes

We wish to transmit a message X on a noisy channel. The message is augmented with some "check-sums" to allow for error correction.

Multiway tables provide an appealing way of organizing the check sum protocol. The sender transmits the message as a multiway table augmented with slack entries summing up to pre-agreed margins.

The receiver gets a distorted table X and reconstructs the message as that table X with the pre-agreed margins that is l_p-nearest to X.

Corollary: We can error-correct messages of format $m_1 \times \cdots \times m_k \times n$ in polynomial time.

Shmuel Onn

Bibliography

available online at http://ie.technion.ac.il/~onn

- Convex discrete optimization (Montréal Lecture Notes)
- Shaped partition problems (SIAM Opt.)
- Cutting corners (Adv. App. Math.)
- Partition problems with convex objectives (Math. OR)
- Convex matroid optimization (SIAM Disc. Math.)
- The Hilbert zonotope and universal Gröbner bases (Adv. App. Math.)
- The complexity of 3-way tables (SIAM Comp.)
- Convex combinatorial optimization (Disc. Comp. Geom.)
- Markov bases of 3-way tables (J. Symb. Comp.)
- All linear and integer programs are slim 3-way programs (SIAM Opt.)
- N-fold integer programming (Disc. Opt. in memory of Dantzig)
- Graver complexity of integer programming (Annals Combin.)
- Nonlinear bipartite matching (Disc. Opt.)
- Convex integer maximization via Graver bases (J. Pure App. Algebra)
- Nonlinear matroid optimization and experimental design (SIAM Disc. Math.)
- Convex integer minimization (submitted)
- Nonlinear optimization over a weighted independence system (submitted)