Multiway Tables: Universality and Optimization

Shmuel Onn

Technion – Israel Institute of Technology

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

Based on several papers joint with various subsets of
{De Loera, Hemmecke, Rothblum, Weismantel}

Supported in part by ISF – Israel Science Foundation

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.

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 2×3:

0	1	2
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	

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	
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:

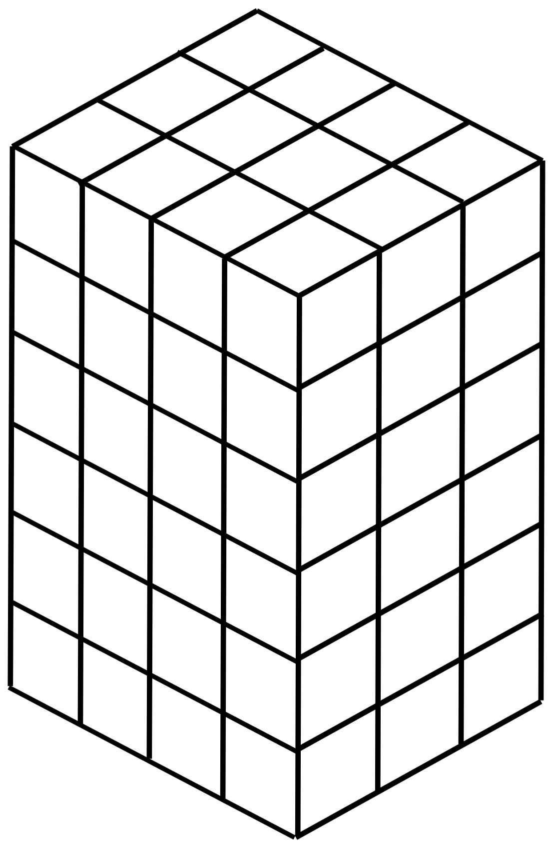
0	1	2	3
2	2	0	4
2	3	2	

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 $3 \times 4 \times 6$:



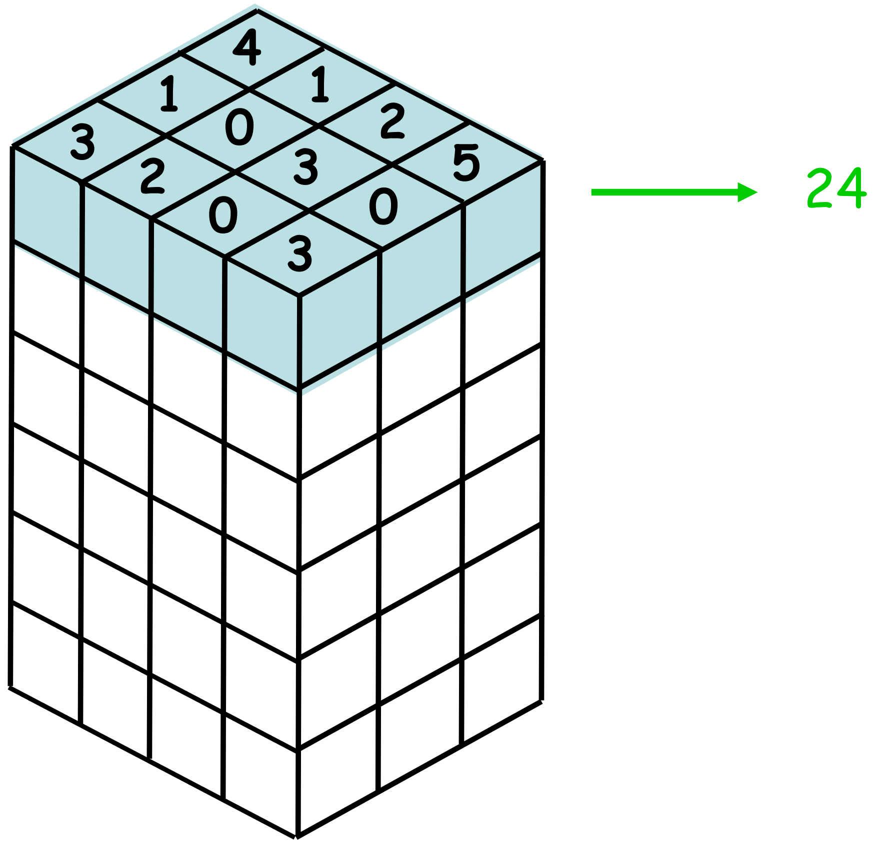
Shmuel Onn

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 $3 \times 4 \times 6$ with a plane-sum:



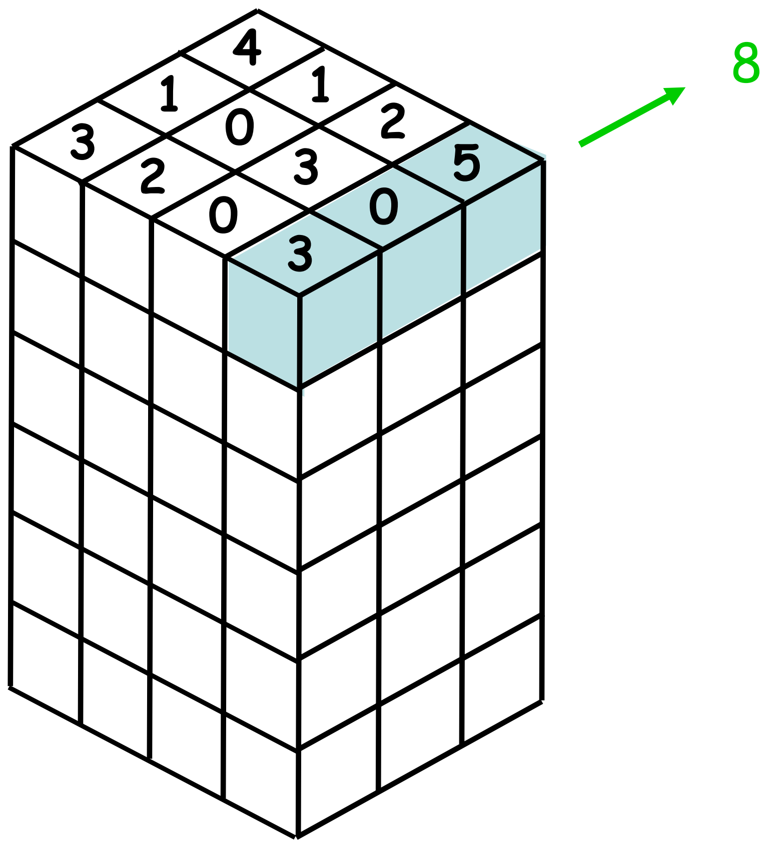
Shmuel Onn

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 $3 \times 4 \times 6$ with a line-sum:



Shmuel Onn

A multiway (transportation) polytope is the set of all nonnegative $m_1 \times \cdots \times m_k$ arrays with some margins fixed.

A multiway (transportation) polytope is the set of all nonnegative $m_1 \times \cdots \times m_k$ arrays with some margins fixed.

The $m_1 \times \cdots \times m_k$ tables with some margins fixed are the integer points in the corresponding multiway polytope.

A multiway (transportation) polytope is the set of all nonnegative $m_1 \times \cdots \times m_k$ arrays with some margins fixed.

The $m_1 \times \cdots \times m_k$ tables with some margins fixed are the integer points in the corresponding multiway polytope.

Two contrasting Statements:

A multiway (transportation) polytope is the set of all nonnegative $m_1 \times \cdots \times m_k$ arrays with some margins fixed.

The $m_1 \times \cdots \times m_k$ tables with some margins fixed are the integer points in the corresponding multiway polytope.

Two contrasting Statements:

Universality Theorem: Any rational polytope is an $r \times c \times 3$ line-sum polytope.

A multiway (transportation) polytope is the set of all nonnegative $m_1 \times \cdots \times m_k$ arrays with some margins fixed.

The $m_1 \times \cdots \times m_k$ tables with some margins fixed are the integer points in the corresponding multiway polytope.

Two contrasting Statements:

Universality Theorem: Any rational polytope is an $r \times c \times 3$ line-sum polytope.

Optimization Theorem: (Convex) Integer Programming over $m_1 \times \cdots \times m_k \times n$ polytopes is solvable in polynomial time.

Shmuel Onn

Some Formalism: Hierarchical Margins

More formally, a k-way polytope is the set of all $m_1 \times \cdots \times m_k$ nonnegative arrays $x = (x_{i_1,\ldots,i_k})$ such that the sums of the entries over some of their lower dimensional sub-arrays (margins) are specified. More precisely, for any tuple $(i_1, \ldots, i_k)$ with $i_j \in \{1, \ldots, m_j\} \cup \{+\}$, the corresponding *margin* $x_{i_1,\ldots,i_k}$ is the sum of entries of x over all coordinates j with $i_j = +$. The *support* of $(i_1, \ldots, i_k)$ and of $x_{i_1,\ldots,i_k}$ is the set $\mathrm{supp}(i_1, \ldots, i_k) := \{j : i_j \neq +\}$ of non-summed coordinates. For instance, if x is a $4 \times 5 \times 3 \times 2$ array then it has 12 margins with support $F = \{1, 3\}$ such as $x_{3,+,2,+} = \sum_{i_2=1}^{5} \sum_{i_4=1}^{2} x_{3,i_2,2,i_4}$. A collection of margins is *hierarchical* if, for some family $\mathcal{F}$ of subsets of $\{1, \ldots, k\}$, it consists of all margins $u_{i_1,\ldots,i_k}$ with support in $\mathcal{F}$. In particular, for any $0 \leq h \leq k$, the collection of all h-margins of k-tables is hierarchical with $\mathcal{F}$ the family of all h-subsets of $\{1, \ldots, k\}$. Given a hierarchical collection of margins $u_{i_1,\ldots,i_k}$ supported on a family $\mathcal{F}$ of subsets of $\{1, \ldots, k\}$, the corresponding *k-way polytope* is the set of nonnegative arrays with these margins,

$$T_{\mathcal{F}} = \left\{ x \in \mathbb{R}_+^{m_1 \times \cdots \times m_k} \ : \ x_{i_1,\ldots,i_k} = u_{i_1,\ldots,i_k} , \ \ \mathrm{supp}(i_1, \ldots, i_k) \in \mathcal{F} \right\} .$$

The integer points in this polytope are precisely the k-way tables with the specified margins.

Universality and its Consequences

Universality Theorem for Short 3-Way Polytopes

Theorem: Any rational polytope $P = \{y \in \mathbb{R}^m_+ : Ay = b\}$ is polytime representable as an $r \times c \times 3$ line-sum polytope

$$T = \left\{ x \in \mathbb{R}^{r\times c\times 3}_+ : \sum_i x_{i,j,k} = w_{j,k}\,,\ \sum_j x_{i,j,k} = v_{i,k}\,,\ \sum_k x_{i,j,k} = u_{i,j} \right\}$$

(there is a coordinate-erasing projection from $\mathbb{R}^{r\times c\times 3}$ to $\mathbb{R}^m$ giving a bijection between T and P and between their integer points).

Universality Theorem for Short 3-Way Polytopes

Theorem: Any rational polytope $P = \{y \in \mathbb{R}^m_+ \,:\, Ay = b\}$ is polytime representable as an $r \times c \times 3$ line-sum polytope

$$T = \left\{ x \in \mathbb{R}^{r\times c\times 3}_+ \,:\, \sum_i x_{i,j,k} = w_{j,k}\,,\ \sum_j x_{i,j,k} = v_{i,k}\,,\ \sum_k x_{i,j,k} = u_{i,j} \right\}$$

(there is a coordinate-erasing projection from $\mathbb{R}^{r\times c\times 3}$ to $\mathbb{R}^m$ giving a bijection between T and P and between their integer points).

⟶ Any linear/integer program is polytime representable as an r x c x 3 multiway program.

Universality Theorem for Short 3-Way Polytopes

Theorem: Any rational polytope $P = \{y \in \mathbb{R}^m_+ : Ay = b\}$ is polytime representable as an $r \times c \times 3$ line-sum polytope

$$T = \left\{ x \in \mathbb{R}^{r \times c \times 3}_+ : \sum_i x_{i,j,k} = w_{j,k}\,, \sum_j x_{i,j,k} = v_{i,k}\,, \sum_k x_{i,j,k} = u_{i,j} \right\}$$

(there is a coordinate-erasing projection from $\mathbb{R}^{r \times c \times 3}$ to $\mathbb{R}^m$ giving a bijection between T and P and between their integer points).

→ Any linear/integer program is polytime representable as an r x c x 3 multiway program.

→ Optimization over r x c x 3 tables is NP-hard.

Universality Theorem for Short 3-Way Polytopes

Theorem: Any rational polytope $P = \{y \in \mathbb{R}^m_+ : Ay = b\}$ is polytime representable as an $r \times c \times 3$ line-sum polytope

$$T = \left\{ x \in \mathbb{R}^{r\times c\times 3}_+ \ : \ \sum_i x_{i,j,k} = w_{j,k}\,,\ \sum_j x_{i,j,k} = v_{i,k}\,,\ \sum_k x_{i,j,k} = u_{i,j} \right\}$$

(there is a coordinate-erasing projection from $\mathbb{R}^{r\times c\times 3}$ to $\mathbb{R}^m$ giving a bijection between T and P and between their integer points).

⟶ Any linear/integer program is polytime representable as an r x c x 3 multiway program.

⟶ Optimization over r x c x 3 tables is NP-hard.

⟶ Implications on the existence of a strongly polynomial time algorithm for linear programming ?

Universality Theorem for Short 3-Way Polytopes

Theorem: Any rational polytope $P = \{y \in \mathbb{R}^m_+ : Ay = b\}$ is polytime representable as an $r \times c \times 3$ line-sum polytope

$$T = \left\{ x \in \mathbb{R}^{r \times c \times 3}_+ : \sum_i x_{i,j,k} = w_{j,k} \, , \, \sum_j x_{i,j,k} = v_{i,k} \, , \, \sum_k x_{i,j,k} = u_{i,j} \right\}$$

(there is a coordinate-erasing projection from $\mathbb{R}^{r \times c \times 3}$ to $\mathbb{R}^m$ giving a bijection between T and P and between their integer points).

→ Any linear/integer program is polytime representable as an r x c x 3 multiway program.

→ Optimization over r x c x 3 tables is NP-hard.

→ Implications on the existence of a strongly polynomial time algorithm for linear programming ?

→ Implications on the rational version of Hilbert's 10th problem on the decidability of the realization problem for polytopes ?

Shmuel Onn

Table Security (confidential data disclosure)

Agencies such as the census bureau and center for health statistics allow public web-access to information on their data bases, but are concerned about confidentiality of individuals.

Table Security (confidential data disclosure)

Agencies such as the census bureau and center for health statistics
allow public web-access to information on their data bases,
but are concerned about confidentiality of individuals.

Common strategy: release margins but not table entries.

Table Security (confidential data disclosure)

Agencies such as the census bureau and center for health statistics
allow public web-access to information on their data bases,
but are concerned about confidentiality of individuals.

Common strategy: release margins but not table entries.

Question: how does the set of values that can occur in a
specific entry in all tables with the released margins look like ?

Shmuel Onn

Fact: for k-way tables with fixed hyperplane-sums, the set of values in an entry is always an interval.

Example: the values 0, 2 occur in an entry:

0	1	2	3
2	2	0	4
2	3	2	

2	1	0	3
0	2	2	4
2	3	2	

Shmuel Onn

Fact: for k-way tables with fixed hyperplane-sums, the set of values in an entry is always an interval.

Example: the values 0, 2 occur in an entry:

0	1	2	3
2	2	0	4
2	3	2	

2	1	0	3
0	2	2	4
2	3	2	

Therefore, also the value 1 occurs in that entry:

1	1	1	3
1	2	1	4
2	3	2	

In contrast we have the following universality:

Theorem: For every finite set S of nonnegative integers, there are r, c and line-sums for $r \times c \times 3$ tables such that the set of values occurring in a fixed entry in all possible tables with these line-sums is precisely S.

In contrast we have the following universality:

Theorem: For every finite set **S** of nonnegative integers, there are r, c and line-sums for r x c x 3 tables such that the set of values occurring in a fixed entry in all possible tables with these line-sums is precisely **S**.

Proof: Given $S = \{s_1, \ldots, s_m\}$, let

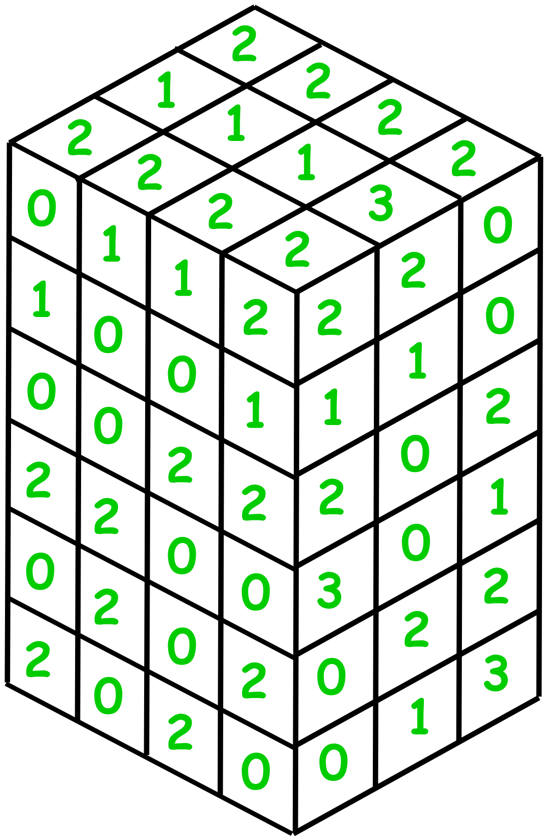
$$P := \{y \in \mathbb{R}^{m+1}_{+} \ : \ y_0 - \sum_{i=1}^{m} s_i y_i = 0 \, , \ \sum_{i=1}^{m} y_i = 1\}.$$

Lift P using the universality theorem to $r \times c \times 3$ line-sum polytope T.

Example: set of entry values with a gap

Example: set of entry values with a gap

Consider the following line-sums for **6** x **4** x **3** tables:

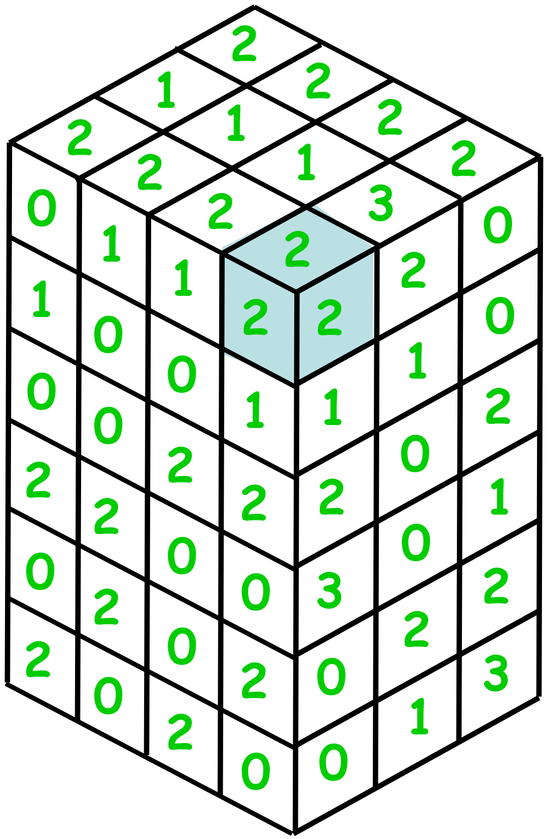


Shmuel Onn

Example: set of entry values with a gap

Consider the following line-sums for **6** x **4** x **3** tables:

Consider the designated entry:

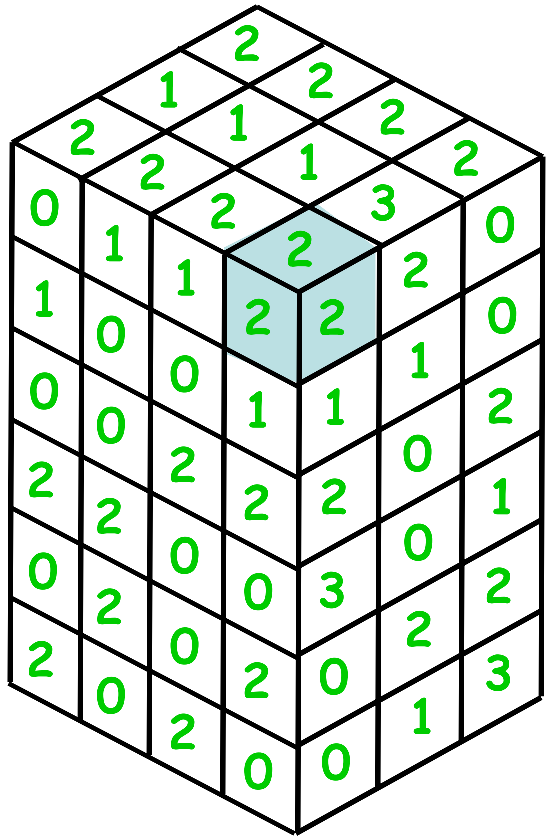


Shmuel Onn

Example: set of entry values with a gap

Consider the following line-sums for **6** x **4** x **3** tables:

Consider the designated entry:



The **only values** occurring in that entry in all possible tables with these line-sums are **0**, **2**

Shmuel Onn

Certain perception: if the set of values that can occur in a specific entry in all tables with the released margins contains many values then the entry is secure; otherwise it is vulnerable.

So common practice is to compute by linear programming lower bound **L** and upper bound **U** on the possible values of an entry and use the gap **U-L** as a measure of its security.

LP-Relaxation is Arbitrarily Bad

Since integer programming problems are generally intractable, a common practice by disclosing agencies is to compute a lower bound $\hat{l}$ and an upper bound $\hat{u}$ on the entry $x_{i_1,\ldots,i_k}$ in all tables with these margins, by solving the *linear programming relaxations* of the corresponding multiway programs,

$$\hat{l} \ := \ \min\{x_{i_1,\ldots,i_k} \ : \ x \in \mathbb{R}_+^{m_1\times\cdots\times m_k}, \quad x_{i_1,\ldots,i_k} = u_{i_1,\ldots,i_k}, \ \mathrm{supp}(i_1,\ldots,i_k) \in \mathcal{F}\}$$

$$\hat{u} \ := \ \max\{x_{i_1,\ldots,i_k} \ : \ x \in \mathbb{R}_+^{m_1\times\cdots\times m_k}, \quad x_{i_1,\ldots,i_k} = u_{i_1,\ldots,i_k}, \ \mathrm{supp}(i_1,\ldots,i_k) \in \mathcal{F}\}$$

that is, where the variables are nonnegative real numbers without integrality constraints. While this can be done efficiently for tables of any size, it is only an approximation on the true smallest value l and largest value u of that entry in (integer) tables, and can be far from the truth; it is easy to design examples (using again the Universality Theorem) of line-sums for $r \times c \times 3$ table where there is a unique integer entry $x_{1,1,1}$, while the linear programming bounds are arbitrarily far apart, that is,

$$\hat{l} \ << \ l \ = \ x_{1,1,1} \ = \ u \ << \ \hat{u} \ ,$$

which may lead to erroneously declaring insecure margin disclosure as secure. Indeed, let u be any large positive integer. Consider the triangle $P_u := \{y \in \mathbb{R}_+^2 : 2y_1+(2u+1)y_2 = 4u+1\}$. It has just one integer point $y = (u,1)$, with $y_1 = u$, while $\hat{l} := \min\{y_1 : y \in P_u\} = 0$ and $\hat{u} := \max\{y_1 : y \in P_u\} = 2u + \frac{1}{2}$. Lifting P_u to a suitable $r \times c \times 3$ line-sum polytope T_u with the coordinate y_1 embedded in the entry $x_{1,1,1}$ using Universality, we find that T_u has just one integer table, where the entry $x_{1,1,1}$ attains the unique value $l = x_{1,1,1} = u$, while the linear programming bounds are $\hat{l} = 0 << u << 2u + \frac{1}{2} = \hat{u}$.

As a simple consequence of our Convex Integer Programming Theorem we get, for the first time, a polynomial time algorithm allowing to compute the true smallest value l and largest value u over long d-way tables, enabling exact solution of the entry uniqueness problem and taking accurate decisions.

Hardness of Entry Uniqueness

Corollary *It is coNP-complete to decide, given r, c and consistent 2-margins (line-sums) for 3-way tables of size $r \times c \times 3$, if the value of the entry $x_{1,1,1}$ is the same in all tables with these margins.*

Proof. From the complement of *subset-sum*: given positive integers $a_0, a_1, \ldots, a_m$, need to decide if there is no $I \subseteq \{1, \ldots, m\}$ with $a_0 = \sum_{i \in I} a_i$. Consider the polytope in variables $y_0, y_1 \ldots, y_m, z_0, z_1, \ldots, z_m$,

$$P \quad := \quad \{(y, z) \in \mathbb{R}_+^{2(m+1)} \ : \ a_0 y_0 - \sum_{i=1}^{m} a_i y_i = 0\,, \ y_i + z_i = 1\,, \ i = 0, 1 \ldots, m\} \quad .$$

First, note that it always has one integer point with $y_0 = 0$, given by $y_i = 0$ and $z_i = 1$ for all i. Second, note that it has an integer point with $y_0 \neq 0$ if and only if there is an $I \subseteq \{1, \ldots, m\}$ with $a_0 = \sum_{i \in I} a_i$, given by $y_0 = 1$, $y_i = 1$ for $i \in I$, $y_i = 0$ for $i \in \{1, \ldots, m\} \setminus I$, and $z_i = 1 - y_i$ for all i. Lifting P to a suitable $r \times c \times 3$ line-sum polytope T with the coordinate y_0 embedded in the entry $x_{1,1,1}$ using Universality, we find that T has a table with $x_{1,1,1} = 0$, and this value is unique among the tables in T if and only if there is *no* solution to the subset sum problem with $a_0, a_1, \ldots, a_m$. $\square$

More Universality Consequences

Universality Theorem for Toric Ideals: Every toric ideal is embeddable in a toric ideal of $r \times c \times 3$ tables with fixed line-sums.

More Universality Consequences

Universality Theorem for Toric Ideals: Every toric ideal is embeddable in a toric ideal of $r \times c \times 3$ tables with fixed line-sums.

Solution of the Vlach Problems: Many problems of the corner stone paper by M. Vlach on transportation polytopes resolved.

More Universality Consequences

Universality Theorem for Toric Ideals: Every toric ideal is embeddable in a toric ideal of r x c x 3 tables with fixed line-sums.

Solution of the Vlach Problems: Many problems of the corner stone paper by M. Vlach on transportation polytopes resolved.

Universality Theorem for Bitransportation Polytopes:

Theorem: Any rational polytope $P = \{y \in \mathbb{R}^m_+ : Ay = b\}$ is polytime representable as an $n \times n$ bitransportation polytope

$$B = \left\{ (x^1, x^2) \in \oplus_2 \mathbb{R}^{n \times n}_+ \;:\; \sum_j x^k_{i,j} = r^k_i \,,\; \sum_i x^k_{i,j} = c^k_j \,,\; x^1_{i,j} + x^2_{i,j} \le u_{i,j} \right\}$$

Example 1. Vlach's rational-nonempty integer-empty transportation: using our construction, we automatically recover the smallest known example, first discovered by Vlach [21], of a rational-nonempty integer-empty transportation polytope, as follows. We start with the polytope $P = \{y \geq 0 : 2y = 1\}$ in one variable, containing a (single) rational point but no integer point. Our construction represents it as a transportation polytope T of $(6, 4, 3)$-arrays with line-sums given by the three matrices below; by Theorem 1, T is integer equivalent to P and hence also contains a (single) rational point but no integer point.

$$(u_{i,j}) = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}, \quad (v_{i,k}) = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \quad (w_{j,k}) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Example 2. Bipartite biflows with arbitrarily large denominator: Fix any positive integer q. Start with the polytope $P = \{y \geq 0 : qy = 1\}$ in one variable containing the single point $y = \frac{1}{q}$. Our construction represents it as a bipartite biflow polytope F with integer supplies, demands and capacities, where y is embedded as the flow $x^1_{1,1}$ of the first commodity from vertex $1 \in R$ to $1 \in C$. By Corollary 2, F contains a single biflow with $x^1_{1,1} = y = \frac{1}{q}$. For $q = 3$, the data for the biflow problem is below, resulting in a unique, $\{0, \frac{1}{3}, \frac{2}{3}\}$-valued, biflow.

$$(u_{i,j}) = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}, \quad (s^1_i) = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad (s^2_i) = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix},$$

$$(d^1_j) = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}, \quad (d^2_j) = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 2 & 1 \end{pmatrix}.$$

Table Sampling: Markov Bases

Shmuel Onn

Table Sampling: Markov Bases

A **Markov basis** is a set of arrays that enables a walk between any two tables with the same margins while staying nonnegative.

Shmuel Onn

Table Sampling: Markov Bases

A **Markov basis** is a set of arrays that enables a walk between any two tables with the same margins while staying nonnegative.

It enables sampling the (huge) set of tables with fixed margins.

Table Sampling: Markov Bases

A **Markov basis** is a set of arrays that enables a walk between any two tables with the same margins while staying nonnegative.

It enables sampling the (huge) set of tables with fixed margins.

Example: Markov bases of rxc tables with fixed line-sums are 2x2 minors:

1	-1	0
-1	1	0
0	0	0

1	0	-1
-1	0	1
0	0	0

. . .

0	0	0
1	0	-1
-1	0	1

0	0	0
0	1	-1
0	-1	1

Table Sampling: Markov Bases

A **Markov basis** is a set of arrays that enables a walk between any two tables with the same margins while staying nonnegative.

It enables sampling the (huge) set of tables with fixed margins.

Example: Markov bases of rxc tables with fixed line-sums are 2x2 minors:

1	-1	0
-1	1	0
0	0	0

1	0	-1
-1	0	1
0	0	0

. . .

0	0	0
1	0	-1
-1	0	1

0	0	0
0	1	-1
0	-1	1

So Markov bases of rxc tables with fixed line-sums are simple:
they have constant support 4 and constant degree 1 regardless of r,c.

Shmuel Onn

Table Sampling: Markov Bases

A **Markov basis** is a set of arrays that enables a walk between any two tables with the same margins while staying nonnegative.

It enables sampling the (huge) set of tables with fixed margins.

Example: Markov bases of rxc tables with fixed line-sums are 2x2 minors:

1	-1	0
-1	1	0
0	0	0

1	0	-1
-1	0	1
0	0	0

. . .

0	0	0
1	0	-1
-1	0	1

0	0	0
0	1	-1
0	-1	1

So Markov bases of rxc tables with fixed line-sums are simple:
they have constant support 4 and constant degree 1 regardless of r,c.

Same holds for d-tables with fixed hyperplane-sums.

Shmuel Onn

Universality Theorem for Markov Bases

Shmuel Onn

Universality Theorem for Markov Bases

Markov bases of 3-tables with fixed line-sums are much more complicated.

Universality Theorem for Markov Bases

Markov bases of 3-tables with fixed line-sums are much more complicated.

Nice result (Aoki-Takemura, Santos-Sturmfels): for tables of size rxcxn, with two sides r,c fixed and one side n variable, there is an upper bound u(r,c) on degree and support of Markov base elements, regardless of n.

Universality Theorem for Markov Bases

Markov bases of 3-tables with fixed line-sums are much more complicated.

Nice result (Aoki-Takemura, Santos-Sturmfels): for tables of size rxcxn, with two sides r,c fixed and one side n variable, there is an upper bound u(r,c) on degree and support of Markov base elements, regardless of n.

In contrast, we show the following universality of tables of size rxcx3, with one side 3 fixed and smallest possible and two sides r,c variable.

Universality Theorem for Markov Bases

Markov bases of 3-tables with fixed line-sums are much more complicated.

Nice result (Aoki-Takemura, Santos-Sturmfels): for tables of size rxcxn, with two sides r,c fixed and one side n variable, there is an upper bound u(r,c) on degree and support of Markov base elements, regardless of n.

In contrast, we show the following universality of tables of size rxcx3, with one side 3 fixed and smallest possible and two sides r,c variable.

Theorem: For every finite set V of integer vectors, there are r, c such that any Markov basis for r x c x 3 tables with fixed line-sums, restricted to some entries, contains V. So these Markov bases have unbounded degree and support.

Proof: Write $V = \{v^1, \ldots, v^k\}$ with $v^i = (v^i_1, \ldots, v^i_d)$. For each i let $u^i = (v^i)^+$ and $w^i = (v^i)^-$ be the positive and negative parts of v^i respectively, so that $v^i = u^i - w^i$.

Proof: Write $V=\{v^1,\ldots,v^k\}$ with $v^i=(v^i_1,\ldots,v^i_d)$. For each i let $u^i=(v^i)^+$ and $w^i=(v^i)^-$ be the positive and negative parts of v^i respectively, so that $v^i=u^i-w^i$.

Let P be the polytope in nonnegative variables $s_1,t_1,\ldots,s_k,t_k$, $x_1,\ldots,x_d$, satisfying the following equations, with parameter b:

Proof: Write $V = \{v^1, \ldots, v^k\}$ with $v^i = (v^i_1, \ldots, v^i_d)$. For each i let $u^i = (v^i)^+$ and $w^i = (v^i)^-$ be the positive and negative parts of v^i respectively, so that $v^i = u^i - w^i$.

Let P be the polytope in nonnegative variables $s_1, t_1, \ldots, s_k, t_k, x_1, \ldots, x_d$, satisfying the following equations, with parameter b:

$$\sum_{i=1}^{k} (s_i + t_i) \;=\; 1\,, \quad \sum_{i=1}^{k} i\,(s_i + t_i) \;=\; b\,,$$

$$x_j - \sum_{i=1}^{k} (u^i_j\, s_i + w^i_j\, t_i) \;=\; 0\,, \quad j = 1, \ldots, d\,.$$

Proof: Write $V = \{v^1, \ldots, v^k\}$ with $v^i = (v^i_1, \ldots, v^i_d)$. For each i let $u^i = (v^i)^+$ and $w^i = (v^i)^-$ be the positive and negative parts of v^i respectively, so that $v^i = u^i - w^i$.

Let P be the polytope in nonnegative variables $s_1, t_1, \ldots, s_k, t_k$, $x_1, \ldots, x_d$, satisfying the following equations, with parameter b:

$$\sum_{i=1}^{k} (s_i + t_i) \;=\; 1\,, \qquad \sum_{i=1}^{k} i\,(s_i + t_i) \;=\; b\,,$$

$$x_j - \sum_{i=1}^{k} (u^i_j\, s_i + w^i_j\, t_i) \;=\; 0\,, \qquad j = 1, \ldots, d\,.$$

Now, consider any $1 \leq i \leq k$ and set $b = i$. Then P has only two integer points: one with $s_i = 1$ and $x = u^i$, and the other with $t_i = 1$ and $x = w^i$. To connect these two points, any Markov basis must contain their difference which, restricted to the x variables, is precisely $v^i = u^i - w^i$. This holds for $v^1, \ldots, v^k$.

Proof: Write $V = \{v^1, \ldots, v^k\}$ with $v^i = (v^i_1, \ldots, v^i_d)$. For each i let $u^i = (v^i)^+$ and $w^i = (v^i)^-$ be the positive and negative parts of v^i respectively, so that $v^i = u^i - w^i$.

Let P be the polytope in nonnegative variables $s_1, t_1, \ldots, s_k, t_k$, $x_1, \ldots, x_d$, satisfying the following equations, with parameter b:

$$\sum_{i=1}^{k} (s_i + t_i) \ = \ 1 \,, \qquad \sum_{i=1}^{k} i\,(s_i + t_i) \ = \ b \,,$$

$$x_j - \sum_{i=1}^{k} (u^i_j\, s_i + w^i_j\, t_i) \ = \ 0 \,, \qquad j = 1, \ldots, d \,.$$

Now, consider any $1 \leq i \leq k$ and set $b = i$. Then P has only two integer points: one with $s_i = 1$ and $x = u^i$, and the other with $t_i = 1$ and $x = w^i$. To connect these two points, any Markov basis must contain their difference which, restricted to the x variables, is precisely $v^i = u^i - w^i$. This holds for $v^1, \ldots, v^k$.

Now lift P using the universality theorem to a suitable $r \times c \times 3$ line-sum polytope T with lines-sums depending on b. $\square$

Toric ideals and Tables

Shmuel Onn

Toric ideals and Tables

Each table $v = (v_{i_1,\ldots,i_d})$ of size $n_1 \times \cdots \times n_d$ lifts to monomial in variables $x = (x_{i_1,\ldots,i_d})$ indexed by table entries:

$$x^v \quad = \quad \prod_{i_1=1}^{n_1} \cdots \prod_{i_d=1}^{n_d} x_{i_1,\ldots,i_d}^{v_{i_1,\ldots,i_d}}$$

Toric ideals and Tables

Each table $v = (v_{i_1,\ldots,i_d})$ of size $n_1 \times \cdots \times n_d$ lifts to monomial in variables $x = (x_{i_1,\ldots,i_d})$ indexed by table entries:

$$x^v \quad = \quad \prod_{i_1=1}^{n_1} \cdots \prod_{i_d=1}^{n_d} x_{i_1,\ldots,i_d}^{v_{i_1,\ldots,i_d}}$$

For example, $v = $

2	1	0
0	5	4

lifts to $x^v = x_{1,1}^2 x_{1,2} x_{2,2}^5 x_{2,3}^4$

Toric ideals and Tables

Each table $v = (v_{i_1,\ldots,i_d})$ of size $n_1 \times \cdots \times n_d$ lifts to monomial in variables $x = (x_{i_1,\ldots,i_d})$ indexed by table entries:

$$x^v \quad = \quad \prod_{i_1=1}^{n_1} \cdots \prod_{i_d=1}^{n_d} x_{i_1,\ldots,i_d}^{v_{i_1,\ldots,i_d}}$$

The equations forcing the same margins on tables, such as line-sums, plane-sums, and so on, lift to a corresponding toric ideal generated by all binomials coming from pairs of tables with the same margins:

$$\mathrm{I} \;=\; \langle\, x^u - x^v \;:\; u, v \ \text{tables with same margins} \,\rangle \ .$$

Toric ideals and Tables

Each table $v = (v_{i_1,\ldots,i_d})$ of size $n_1 \times \cdots \times n_d$ lifts to monomial in variables $x = (x_{i_1,\ldots,i_d})$ indexed by table entries:

$$x^v \quad = \quad \prod_{i_1=1}^{n_1} \cdots \prod_{i_d=1}^{n_d} x_{i_1,\ldots,i_d}^{v_{i_1,\ldots,i_d}}$$

The equations forcing the same margins on tables, such as line-sums, plane-sums, and so on, lift to a corresponding toric ideal generated by all binomials coming from pairs of tables with the same margins:

$$\mathrm{I} \;=\; \langle\, x^u - x^v \;\;:\;\; u, v \text{ tables with same margins} \,\rangle \;.$$

Fundamental result (Diaconis-Sturmfels): the binomials x^u-x^v generate a toric ideal if and only if the corresponding arrays u-v form a Markov basis.

Toric ideals and Tables

Each table $v = (v_{i_1,\ldots,i_d})$ of size $n_1 \times \cdots \times n_d$ lifts to monomial in variables $x = (x_{i_1,\ldots,i_d})$ indexed by table entries:

$$x^v \quad = \quad \prod_{i_1=1}^{n_1} \cdots \prod_{i_d=1}^{n_d} x_{i_1,\ldots,i_d}^{v_{i_1,\ldots,i_d}}$$

The equations forcing the same margins on tables, such as line-sums, plane-sums, and so on, lift to a corresponding toric ideal generated by all binomials coming from pairs of tables with the same margins:

$$\mathrm{I} \; = \; \langle x^u - x^v \; : \; u, v \text{ tables with same margins} \rangle \; .$$

Fundamental result (Diaconis-Sturmfels): the binomials x^u-x^v generate a toric ideal if and only if the corresponding arrays u-v form a Markov basis.

We have the following universality theorem for toric ideals.

Toric ideals and Tables

Each table $v = (v_{i_1,\dots,i_d})$ of size $n_1 \times \cdots \times n_d$ lifts to monomial in variables $x = (x_{i_1,\dots,i_d})$ indexed by table entries:

$$x^v = \prod_{i_1=1}^{n_1} \cdots \prod_{i_d=1}^{n_d} x_{i_1,\dots,i_d}^{v_{i_1,\dots,i_d}}$$

The equations forcing the same margins on tables, such as line-sums, plane-sums, and so on, lift to a corresponding toric ideal generated by all binomials coming from pairs of tables with the same margins:

$$I = \langle x^u - x^v \ : \ u, v \text{ tables with same margins} \rangle .$$

Fundamental result (Diaconis-Sturmfels): the binomials $x^u - x^v$ generate a toric ideal if and only if the corresponding arrays u-v form a Markov basis.

We have the following universality theorem for toric ideals.

Theorem 3: For every toric ideal I, there are r, c such that any generating set of the ideal of r x c x 3 tables with fixed line-sums, restricted to some variables, contains a generating set of I.

Shmuel Onn

A glimpse at step 3 of the proof of the Universality Theorem:

Shmuel Onn

$$
(u_{I,J}) = \begin{array}{c} \\ 11 \\ 12 \\ \vdots \\ 1m \\ \\ 21 \\ 22 \\ \vdots \\ 2m \\ \\ \vdots \\ \\ l1 \\ l2 \\ \vdots \\ lm \end{array}
\begin{array}{c}
\begin{array}{cccccccccccc} 11 & 12 & \cdots & 1n & 21 & 22 & \cdots & 2l & 31 & 32 & \cdots & 3m \end{array} \\
\left(\begin{array}{cccccccccccc}
e_{1,1,1} & e_{1,1,2} & \cdots & e_{1,1,n} & U & 0 & \cdots & 0 & U & 0 & \cdots & 0 \\
e_{1,2,1} & e_{1,2,2} & \cdots & e_{1,2,n} & U & 0 & \cdots & 0 & 0 & U & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
e_{1,m,1} & e_{1,m,2} & \cdots & e_{1,m,n} & U & 0 & \cdots & 0 & 0 & 0 & \cdots & U \\
\\
e_{2,1,1} & e_{2,1,2} & \cdots & e_{2,1,n} & 0 & U & \cdots & 0 & U & 0 & \cdots & 0 \\
e_{2,2,1} & e_{2,2,2} & \cdots & e_{2,2,n} & 0 & U & \cdots & 0 & 0 & U & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
e_{2,m,1} & e_{2,m,2} & \cdots & e_{2,m,n} & 0 & U & \cdots & 0 & 0 & 0 & \cdots & U \\
\\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\\
e_{l,1,1} & e_{l,1,2} & \cdots & e_{l,1,n} & 0 & 0 & \cdots & U & U & 0 & \cdots & 0 \\
e_{l,2,1} & e_{l,2,2} & \cdots & e_{l,2,n} & 0 & 0 & \cdots & U & 0 & U & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
e_{l,m,1} & e_{l,m,2} & \cdots & e_{l,m,n} & 0 & 0 & \cdots & U & 0 & 0 & \cdots & U
\end{array}\right)
\end{array}
$$

$$
(v_{I,K}) = \begin{array}{c} \\ 11 \\ 12 \\ \vdots \\ 1m \\ 21 \\ 22 \\ \vdots \\ 2m \\ \vdots \\ l1 \\ l2 \\ \vdots \\ lm \end{array}
\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\
\begin{pmatrix}
U & e_{1,1,+} & U \\
U & e_{1,2,+} & U \\
\vdots & \vdots & \vdots \\
U & e_{1,m,+} & U \\
U & e_{2,1,+} & U \\
U & e_{2,2,+} & U \\
\vdots & \vdots & \vdots \\
U & e_{2,m,+} & U \\
\vdots & \vdots & \vdots \\
U & e_{l,1,+} & U \\
U & e_{l,2,+} & U \\
\vdots & \vdots & \vdots \\
U & e_{l,m,+} & U
\end{pmatrix} \end{array}
\quad
(w_{J,K}) = \begin{array}{c} \\ 11 \\ 12 \\ \vdots \\ 1n \\ 21 \\ 22 \\ \vdots \\ 2l \\ 31 \\ 32 \\ \vdots \\ 3m \end{array}
\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\
\begin{pmatrix}
c_1 & e_{+,+,1} - c_1 & 0 \\
c_2 & e_{+,+,2} - c_2 & 0 \\
\vdots & \vdots & \vdots \\
c_n & e_{+,+,n} - c_n & 0 \\
m \cdot U - a_1 & 0 & a_1 \\
m \cdot U - a_2 & 0 & a_2 \\
\vdots & \vdots & \vdots \\
m \cdot U - a_l & 0 & a_l \\
0 & b_1 & l \cdot U - b_1 \\
0 & b_2 & l \cdot U - b_2 \\
\vdots & \vdots & \vdots \\
0 & b_m & l \cdot U - b_m
\end{pmatrix} \end{array}
$$

Convex Integer Programming

The Convex Integer Programming Problem

Shmuel Onn

The Convex Integer Programming Problem

We consider the following convex integer programming problem:

$$\max \{c(w_1 x, \ldots, w_d x) : x \geq 0,\ Ax = b,\ x \text{ integer}\}$$

where $w_1, \ldots, w_d$ are linear forms and c is a convex functional on R^d.

The Convex Integer Programming Problem

We consider the following convex integer programming problem:

$$\max \{c(w_1 x, \ldots, w_d x) : x \geq 0,\ Ax = b,\ x \text{ integer}\}$$

where $w_1, \ldots, w_d$ are linear forms and c is a convex functional on R^d.

The problem can be interpreted as multiobjective integer programming: given d linear criteria, the goal is to maximize their "convex balancing".

The Convex Integer Programming Problem

We consider the following convex integer programming problem:

$$\max \{c(w_1 x, \ldots, w_d x) : x \geq 0,\ Ax = b,\ x \text{ integer}\}$$

where $w_1, \ldots, w_d$ are linear forms and c is a convex functional on R^d.

The problem can be interpreted as multiobjective integer programming: given d linear criteria, the goal is to maximize their "convex balancing".

It is generally intractable even for fixed d=1, since standard linear integer programming is the special case with c the identity on R.

The Convex Integer Programming Problem

We consider the following convex integer programming problem:

$$\max \{c(w_1 x, \ldots, w_d x) : x \geq 0,\ Ax = b,\ x \text{ integer}\}$$

where $w_1, \ldots, w_d$ are linear forms and c is a convex functional on R^d.

The problem can be interpreted as multiobjective integer programming: given d linear criteria, the goal is to maximize their "convex balancing".

It is generally intractable even for fixed $d=1$, since standard linear integer programming is the special case with c the identity on R.

Nonetheless, as a consequence of our more general theorem below, we obtain the following Optimization Theorem for long multiway polytopes:

The Convex Integer Programming Problem

We consider the following convex integer programming problem:

$$\max \{c(w_1 x, \ldots, w_d x) : x \geq 0,\ Ax = b,\ x \text{ integer}\}$$

where $w_1, \ldots, w_d$ are linear forms and c is a convex functional on R^d.

The problem can be interpreted as multiobjective integer programming: given d linear criteria, the goal is to maximize their "convex balancing".

It is generally intractable even for fixed $d=1$, since standard linear integer programming is the special case with c the identity on R.

Nonetheless, as a consequence of our more general theorem below, we obtain the following Optimization Theorem for long multiway polytopes:

Theorem: Fix $d, m_1, \ldots, m_k$. Then convex integer programming over any $m_1 \times \cdots \times m_k \times n$ multiway polytope is solvable in polynomial oracle-time for any margins, $w_1, \ldots, w_d$, and convex c presented by comparison oracle.

Shmuel Onn

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 establish the following theorem.

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 establish the following theorem.

Theorem: For any fixed d and $(r+s) \times t$ matrix A, there is a polynomial oracle-time algorithm that, given n, b, $w_1, \ldots, w_d$, and convex c presented by comparison oracle, solves the convex integer programming problem

$$\max \{ c(w_1 x, \ldots, w_d x) \ : \ A^{(n)} x = b, \ x \text{ in } N^{nt} \}$$

Shmuel Onn

Efficient Treatment of Long Multiway Tables

The margin equations for any $m_1 \times \cdots \times m_k \times n$ polytope form an n-fold system defined by a suitable 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.

Efficient Treatment of Long Multiway Tables

The margin equations for any $m_1 \times \cdots \times m_k \times n$ polytope form an n-fold system defined by a suitable 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:

Consider long 3-way tables of size $3 \times 3 \times n$ with all line-sums fixed, that is, with $k = 2$, $m_1 = m_2 = 3$, and the hierarchical collection of all 2-margins, supported on $\mathcal{F} = \{\{1,2\},\{1,3\},\{2,3\}\}$. Then $r = 9$, $s = 6$, $t = 9$, and writing $x^i = (x_{1,1,i}, x_{1,2,i}, x_{1,3,i}, x_{2,1,i}, x_{2,2,i}, x_{2,3,i}, x_{3,1,i}, x_{3,2,i}, x_{3,3,i})$ for $i = 1, \ldots, n$, the $(9+6) \times 9$ matrix A whose n-fold product $A^{(n)}$ defines the $3 \times 3 \times n$ multiway polytope has $A_1 = I_9$,

$$A_2 \quad = \quad \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \quad .$$

Already for this case, of $3 \times 3 \times n$ tables, the only polynomial time algorithm we are aware of for the corresponding integer programming problem is the one guaranteed by our theorem for n-fold systems.

Proof Ingredient 1: Edge-Directions

Proof Ingredient 1: Edge-Directions

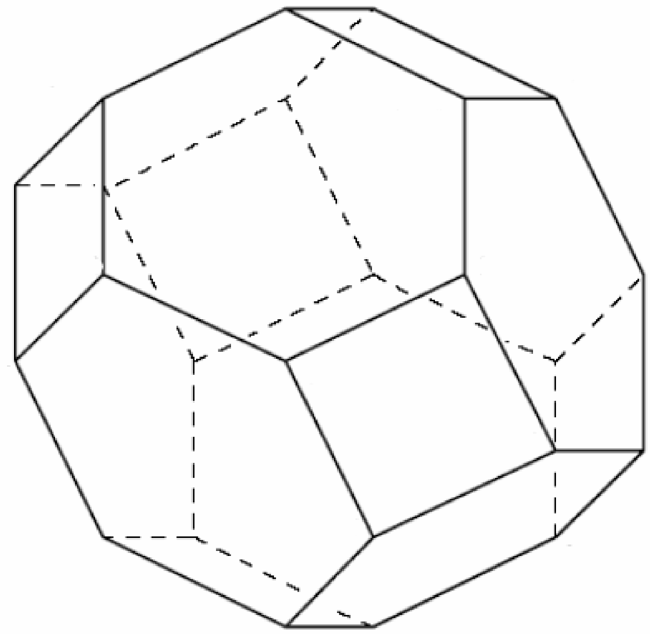
Exploit edge symmetry of the integer hull

$$P = \mathrm{conv}\{x : x \geq 0,\ Ax = b,\ x \text{ integer}\} \subseteq R^n$$

Proof Ingredient 1: Edge-Directions

Exploit edge symmetry of the integer hull

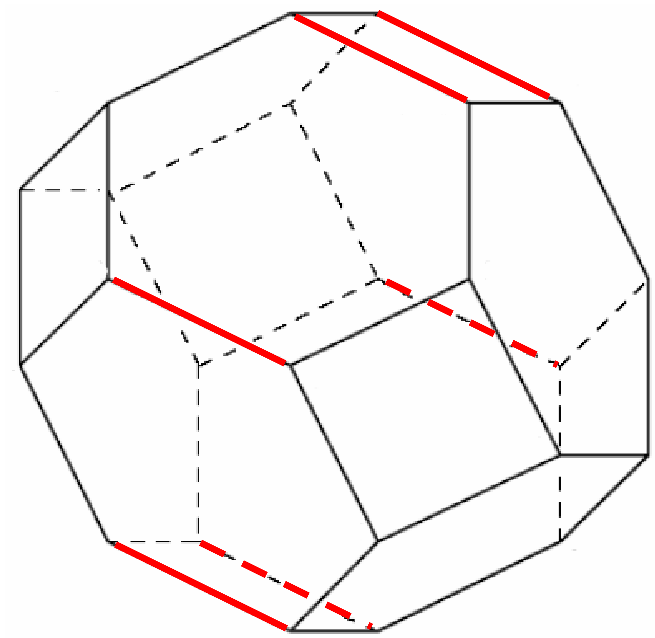
$$P = \text{conv}\{x : x \geq 0,\ Ax = b,\ x \text{ integer}\} \subseteq R^n$$



Proof Ingredient 1: Edge-Directions

Exploit edge symmetry of the integer hull

$$P = \text{conv}\{x : x \geq 0,\ Ax = b,\ x \text{ integer}\} \subseteq R^n$$

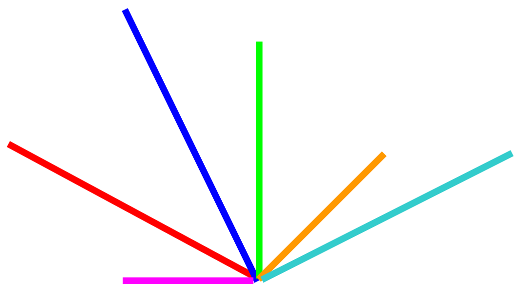


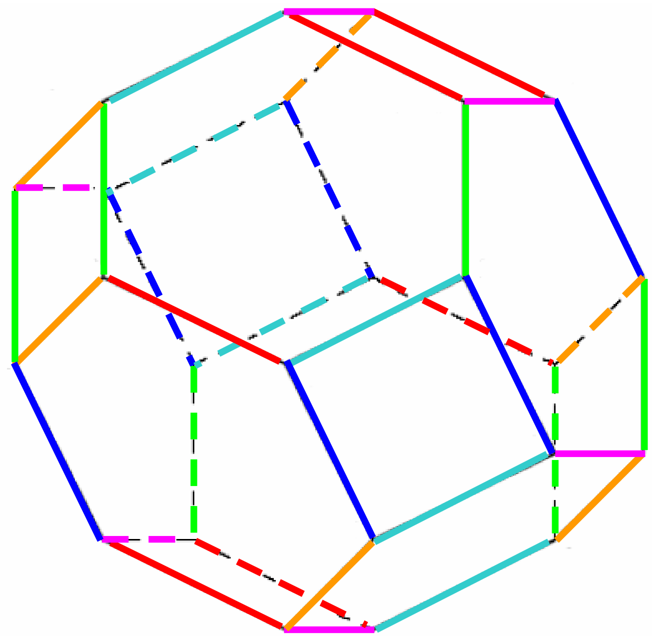
Shmuel Onn

Proof Ingredient 1: Edge-Directions

Exploit edge symmetry of the integer hull

$$P = \text{conv}\{x : x \geq 0,\ Ax = b,\ x \text{ integer}\} \subseteq R^n$$

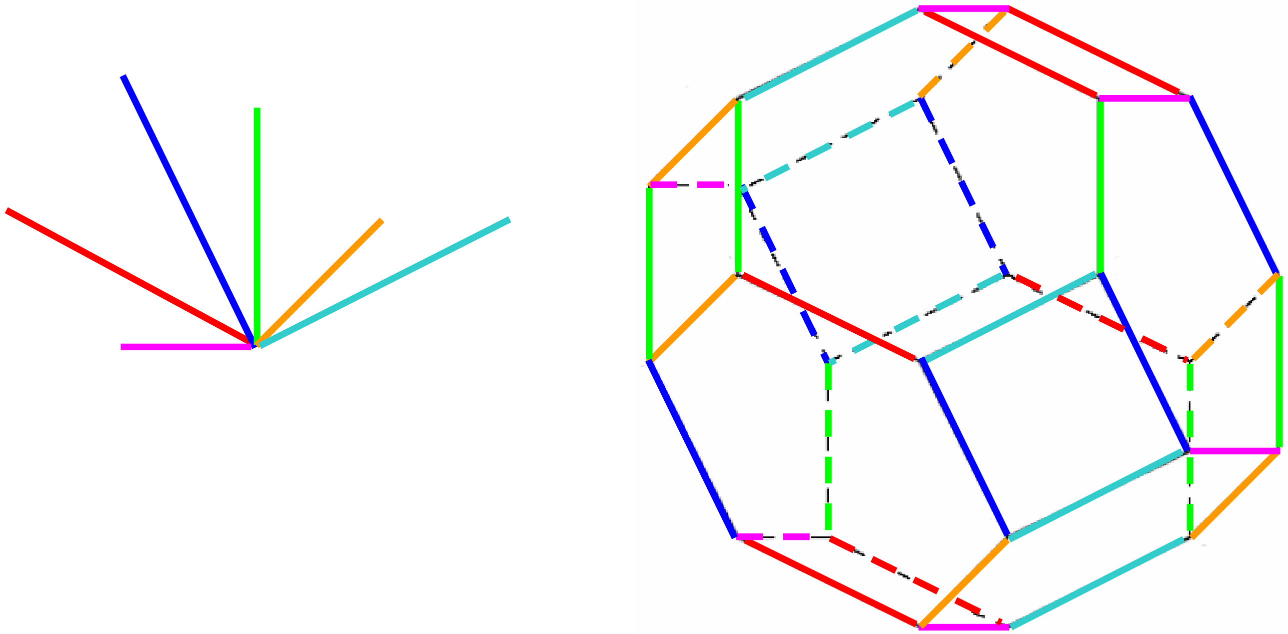




Proof Ingredient 1: Edge-Directions

Exploit edge symmetry of the integer hull

$$P = \mathrm{conv}\{x : x \geq 0,\ Ax = b,\ x \text{ integer}\} \subseteq R^n$$



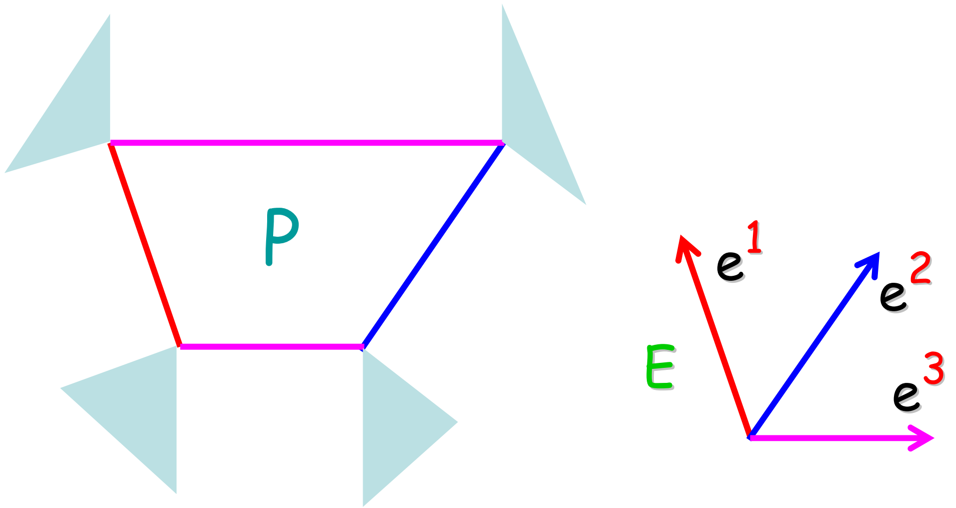
Lemma 1: Fix d. Then, given a set E covering all edge-directions of P, the convex integer programming problem over P is reducible to solving polynomially many linear integer programming counterparts over P.

Zonotope Refinement and Construction

Prop. 1: 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.

Zonotope Refinement and Construction

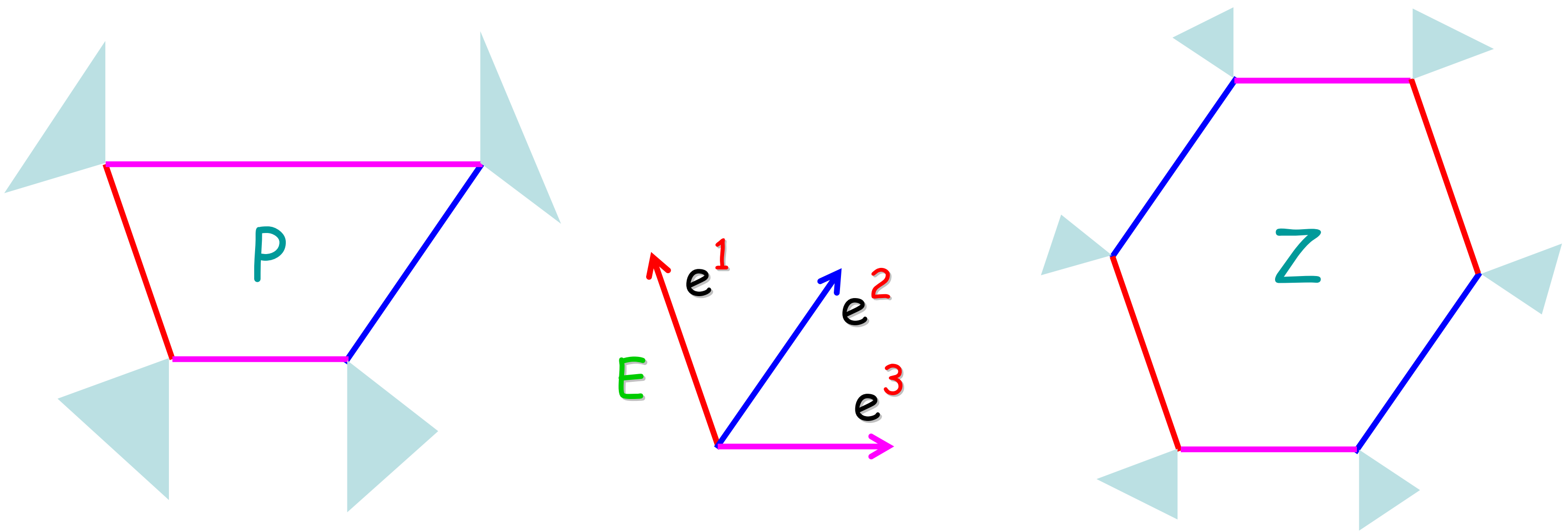
Prop. 1: If $E = \{e^1, \ldots, e^m\}$ covers all edge-directions of a polytope P then the zonotope $Z = [-1, 1]\, e^1 + \ldots + [-1, 1]\, e^m$ is a refinement of P.



Shmuel Onn

Zonotope Refinement and Construction

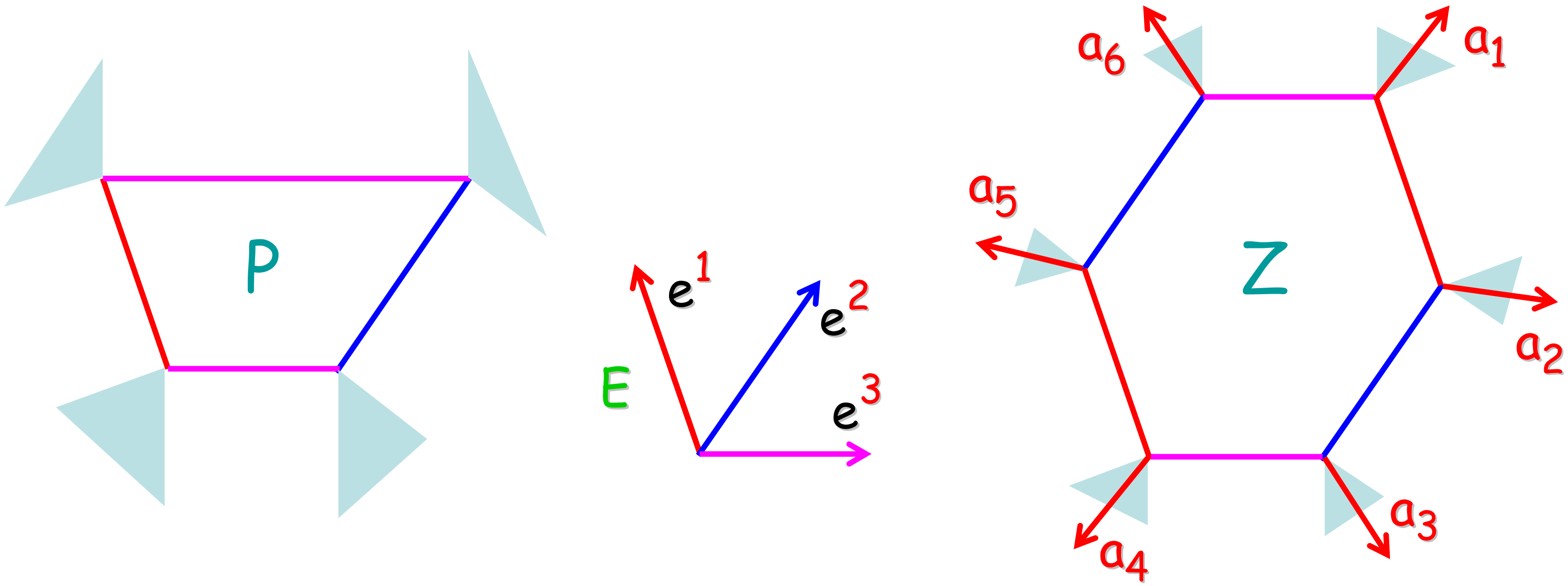
Prop. 1: If $E = \{e^1, \ldots, e^m\}$ covers all edge-directions of a polytope P then the zonotope $Z = [-1, 1]\, e^1 + \ldots + [-1, 1]\, e^m$ is a refinement of P.



Shmuel Onn

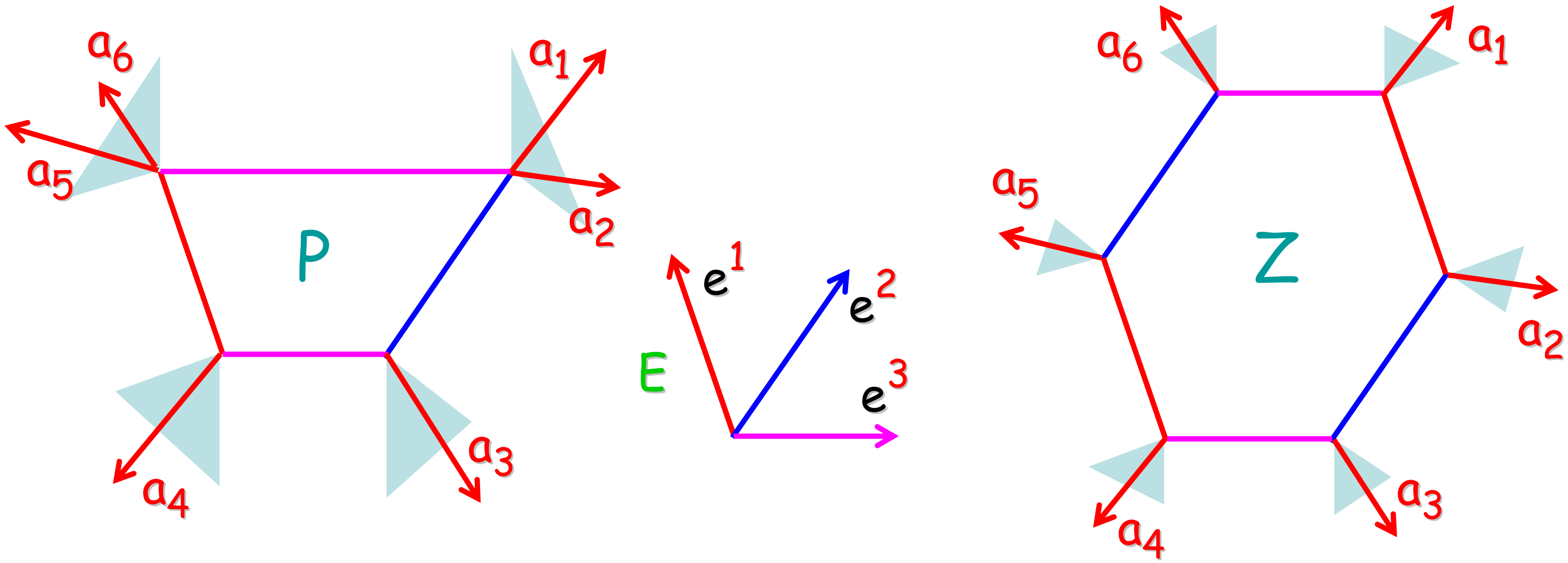
Zonotope Refinement and Construction

Prop. 1: 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.



Zonotope Refinement and Construction

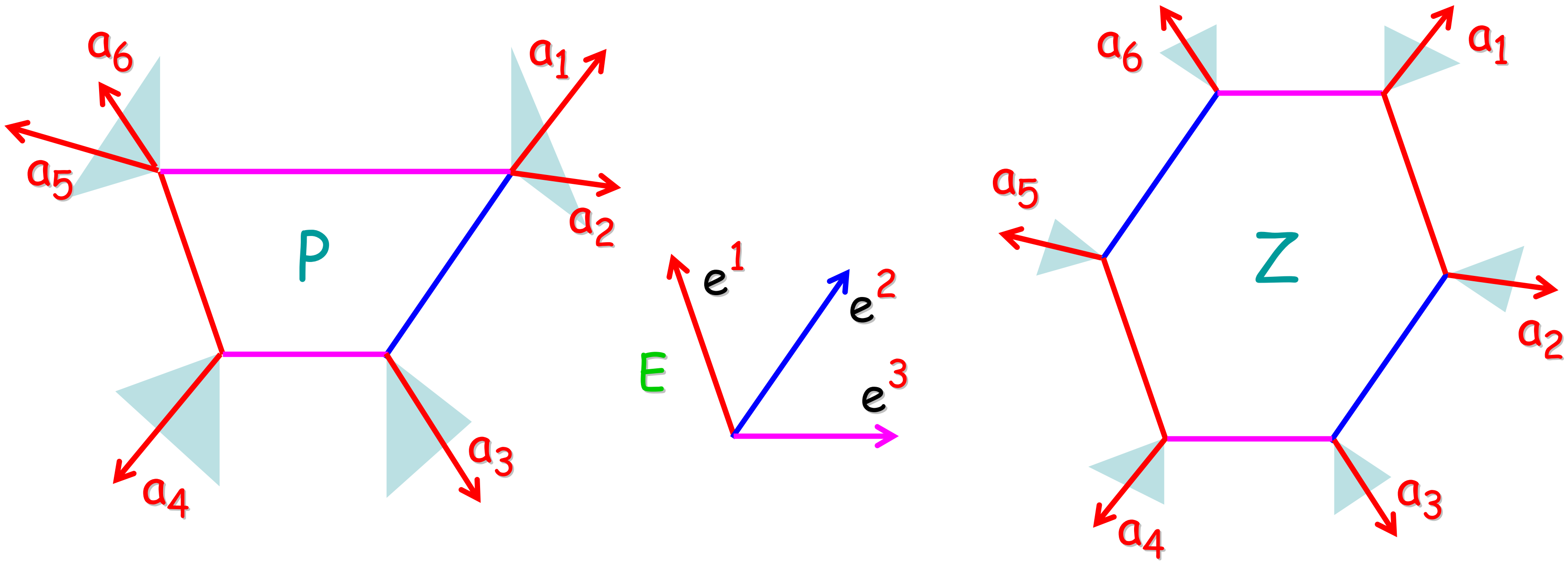
Prop. 1: If $E = \{e^1, \ldots, e^m\}$ covers all edge-directions of a polytope P then the zonotope $Z = [-1, 1]\, e^1 + \ldots + [-1, 1]\, e^m$ is a refinement of P.



Shmuel Onn

Zonotope Refinement and Construction

Prop. 1: 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.



Prop. 2: 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.

Shmuel Onn

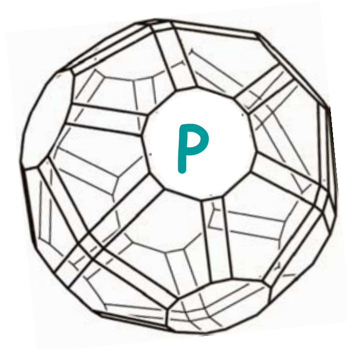
The Algorithm Establishing Lemma 1

Input: Polytope P in R^n given via A,b, set E covering its edge-directions, d x n matrix w, and convex functional c on R^d given by comparison oracle.

The Algorithm Establishing Lemma 1

Input: Polytope P in R^n given via A,b, set E covering its edge-directions, $d \times n$ matrix w, and convex functional c on R^d given by comparison oracle.

E



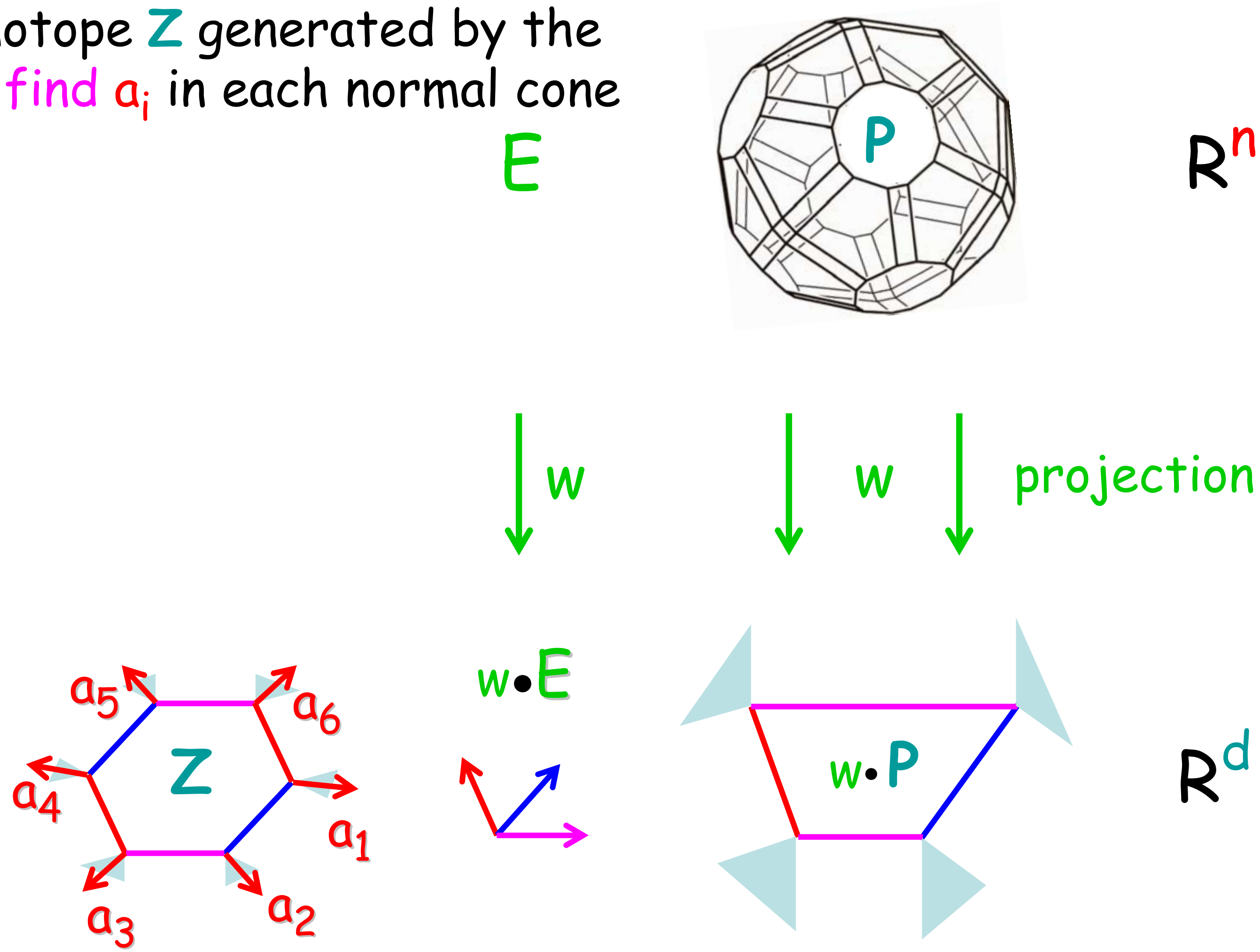
R^n

Shmuel Onn

The Algorithm Establishing Lemma 1

Input: Polytope P in R^n given via A,b, set E covering its edge-directions, d x n matrix w, and convex functional c on R^d given by comparison oracle.

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



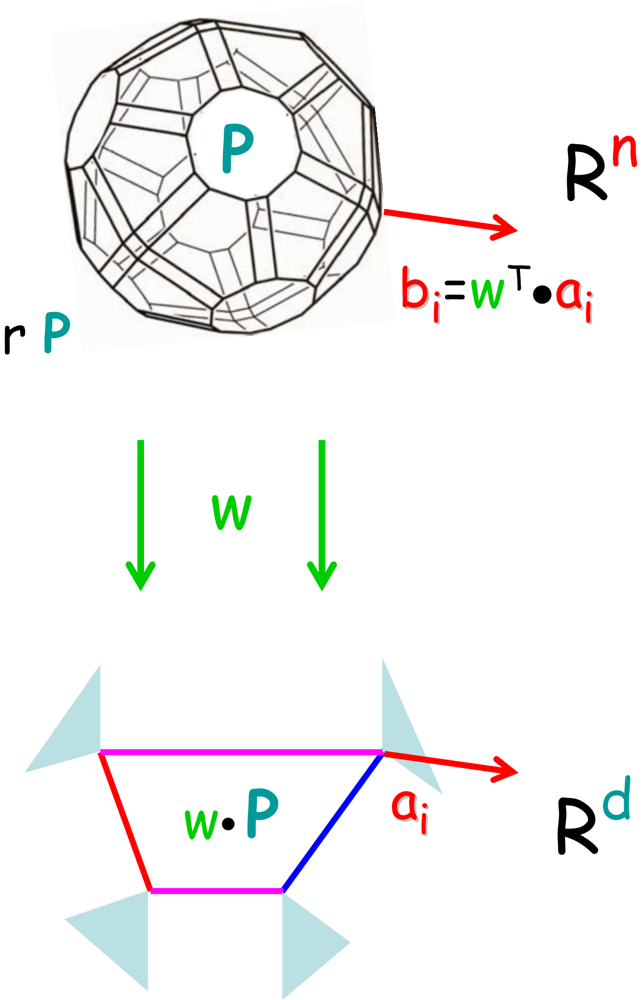
Shmuel Onn

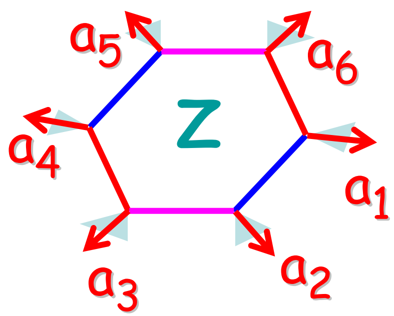
The Algorithm Establishing Lemma 1

Input: Polytope P in R^n given via A,b, set E covering its edge-directions, $d \times n$ matrix w, and convex functional c 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 integer programming with objective b_i over P





Shmuel Onn

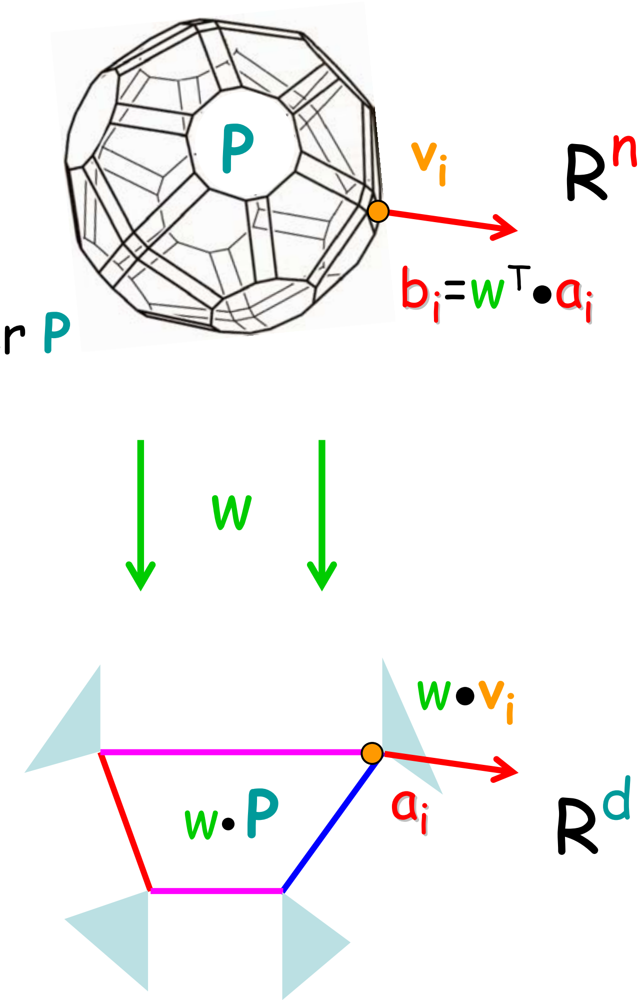
The Algorithm Establishing Lemma 1

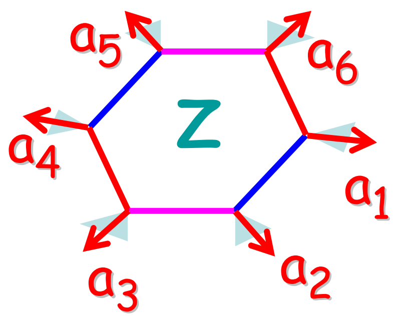
Input: Polytope P in R^n given via A,b, set E covering its edge-directions, $d \times n$ matrix w, and convex functional c on R^d given by comparison oracle.

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 integer programming with objective b_i over P

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





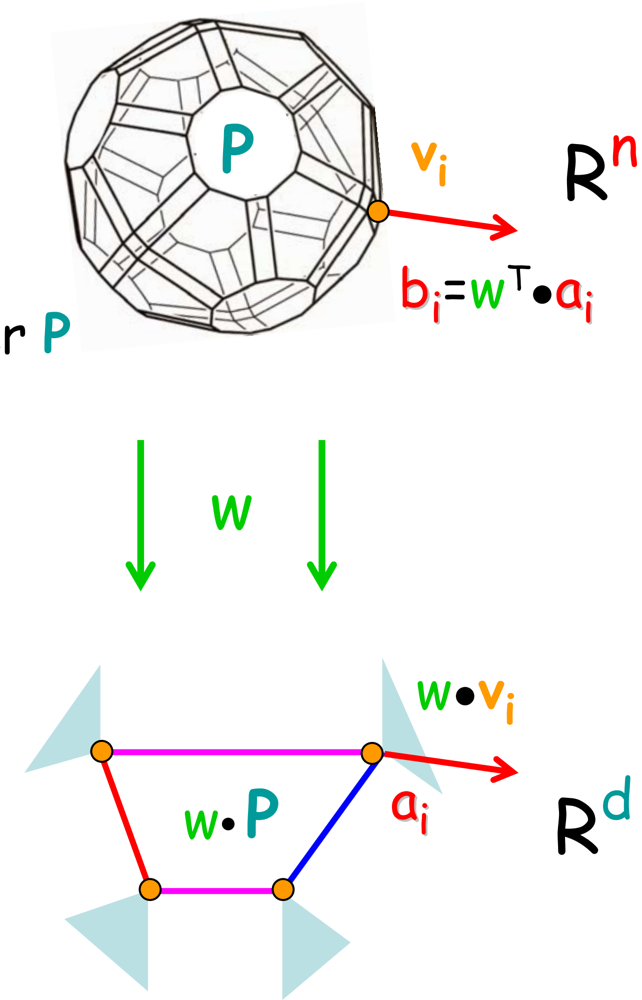
Shmuel Onn

The Algorithm Establishing Lemma 1

Input: Polytope P in R^n given via A,b, set E covering its edge-directions, $d \times n$ matrix w, and convex functional c on R^d given by comparison oracle.

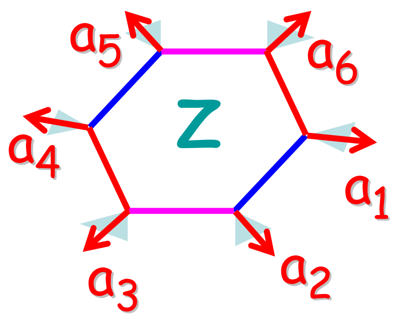
1. Construct the zonotope Z generated by the projection $w \bullet E$, and find a_i in each normal cone

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



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 $c(w \bullet v_i)$ using comparison oracle



Shmuel Onn

Proof Ingredient 2: Graver Bases

Proof Ingredient 2: Graver Bases

The Graver basis of an integer matrix A is the set of conformal-minimal nonzero integer dependencies on A, i.e. vectors with $Av = 0$. For instance, the Graver basis of $A = [1\ 2\ 1]$ is $\pm\{\ [2\ -1\ 0],\ [0\ -1\ 2],\ [1\ 0\ -1],\ [1\ -1\ 1]\ \}$.

Proof Ingredient 2: Graver Bases

The Graver basis of an integer matrix A is the set of conformal-minimal nonzero integer dependencies on A, i.e. vectors with $Av = 0$. For instance, the Graver basis of $A = [1\ 2\ 1]$ is $\pm\{[2\ {-1}\ 0], [0\ {-1}\ 2], [1\ 0\ {-1}], [1\ {-1}\ 1]\}$.

(A vector u is conformal to vector v if $|u_i| \leq |v_i|$ and $u_i v_i \geq 0$ for all i).

Proof Ingredient 2: Graver Bases

The Graver basis of an integer matrix A is the set of conformal-minimal nonzero integer dependencies on A, i.e. vectors with $Av = 0$. For instance, the Graver basis of $A = [1\ 2\ 1]$ is $\pm\{ [2\ {-1}\ 0], [0\ {-1}\ 2], [1\ 0\ {-1}], [1\ {-1}\ 1] \}$.

(A vector u is conformal to vector v if $|u_i| \leq |v_i|$ and $u_i v_i \geq 0$ for all i).

Lemma 2: The Graver basis of A allows to augment in polynomial time any feasible solution to an optimal solution of any linear integer program

$$\max \{ wx : x \geq 0,\ Ax = b,\ x \text{ integer} \}$$

Proof Ingredient 2: Graver Bases

The Graver basis of an integer matrix A is the set of conformal-minimal nonzero integer dependencies on A, i.e. vectors with $Av = 0$. For instance, the Graver basis of $A = [1\ 2\ 1]$ is $\pm\{ [2\ {-1}\ 0], [0\ {-1}\ 2], [1\ 0\ {-1}], [1\ {-1}\ 1] \}$.

(A vector u is conformal to vector v if $|u_i| \leq |v_i|$ and $u_i v_i \geq 0$ for all i).

Lemma 2: The Graver basis of A allows to augment in polynomial time any feasible solution to an optimal solution of any linear integer program

$$\max \{ wx : x \geq 0,\ Ax = b,\ x \text{ integer}\}$$

Proof: use equivalence of directed augmentation and optimization.

Proof Ingredient 2: Graver Bases

The Graver basis of an integer matrix A is the set of conformal-minimal nonzero integer dependencies on A, i.e. vectors with $Av = 0$. For instance, the Graver basis of $A = [1\ 2\ 1]$ is $\pm\{[2\ {-1}\ 0], [0\ {-1}\ 2], [1\ 0\ {-1}], [1\ {-1}\ 1]\}$.

(A vector u is conformal to vector v if $|u_i| \leq |v_i|$ and $u_i v_i \geq 0$ for all i).

Lemma 2: The Graver basis of A allows to augment in polynomial time any feasible solution to an optimal solution of any linear integer program

$$\max\{wx : x \geq 0,\ Ax = b,\ x \text{ integer}\}$$

Proof: use equivalence of directed augmentation and optimization.

Lemma 3: The Graver basis of A covers all edge-directions of any fiber

$$P = \text{conv}\{x : x \geq 0,\ Ax = b,\ x \text{ integer}\}$$

Proof Ingredient 2: Graver Bases

The Graver basis of an integer matrix A is the set of conformal-minimal nonzero integer dependencies on A, i.e. vectors with $Av = 0$. For instance, the Graver basis of $A = [1\ 2\ 1]$ is $\pm\{ [2\ {-1}\ 0], [0\ {-1}\ 2], [1\ 0\ {-1}], [1\ {-1}\ 1] \}$.

(A vector u is conformal to vector v if $|u_i| \le |v_i|$ and $u_i v_i \ge 0$ for all i).

Lemma 2: The Graver basis of A allows to augment in polynomial time any feasible solution to an optimal solution of any linear integer program

$$\max\{ wx : x \ge 0,\ Ax = b,\ x \text{ integer}\}$$

Proof: use equivalence of directed augmentation and optimization.

Lemma 3: The Graver basis of A covers all edge-directions of any fiber

$$P = \text{conv}\{x : x \ge 0,\ Ax = b,\ x \text{ integer}\}$$

Lemma 4: The Graver basis of the product $A^{(n)}$ is polytime computable.

Proof Ingredient 2: Graver Bases

The Graver basis of an integer matrix A is the set of conformal-minimal nonzero integer dependencies on A, i.e. vectors with $Av = 0$. For instance, the Graver basis of $A = [1\ 2\ 1]$ is $\pm\{\ [2\ {-1}\ 0], [0\ {-1}\ 2], [1\ 0\ {-1}], [1\ {-1}\ 1]\ \}$.

(A vector u is conformal to vector v if $|u_i| \leq |v_i|$ and $u_i v_i \geq 0$ for all i).

Lemma 2: The Graver basis of A allows to augment in polynomial time any feasible solution to an optimal solution of any linear integer program

$$\max\{\ wx : x \geq 0,\ Ax = b,\ x \text{ integer}\}$$

Proof: use equivalence of directed augmentation and optimization.

Lemma 3: The Graver basis of A covers all edge-directions of any fiber

$$P = \text{conv}\{x : x \geq 0,\ Ax = b,\ x \text{ integer}\}$$

Lemma 4: The Graver basis of the product $A^{(n)}$ is polytime computable.

Proof: use Graver basis stabilization.

Shmuel Onn

Example of Graver Complexity and Stabilization

Consider the $(2+1)\times 2$ matrix A with $A_1 = I_2$ and $A_2 = [1\ 1]$. The *Graver complexity* of A is $g(A) = 2$. The 2-fold matrix of A and its Graver basis, consisting of two antipodal vectors only, are

$$A^{(2)} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \qquad \mathcal{G}(A^{(2)}) = \pm \begin{pmatrix} 1 & -1 & -1 & 1 \end{pmatrix} .$$

Since $g(A) = 2$, the Graver basis of the 4-fold matrix $A^{(4)}$ can be computed by taking the union of the images of $\mathcal{G}(A^{(2)})$ under the $6 = \binom{4}{2}$ maps $\phi_{k_1,k_2} : \mathbb{Z}^{2\cdot 2} \longrightarrow \mathbb{Z}^{4\cdot 2}$ for $1 \leq k_1 < k_2 \leq 4$, and we obtain

$$A^{(4)} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \ \mathcal{G}(A^{(4)}) = \pm \begin{pmatrix} 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 \end{pmatrix} .$$

Combining Lemmas 1 – 4 plus some additional components, we obtain the aforementioned theorem on n-fold systems:

Combining Lemmas 1 – 4 plus some additional components, we obtain the aforementioned theorem on n-fold systems:

Theorem: For any fixed d and $(r+s) \times t$ matrix A, there is a polynomial oracle-time algorithm that, given n, b, $w_1, \ldots, w_d$, and convex c presented by comparison oracle, solves the convex integer programming problem

$$\max \{ c(w_1 x, \ldots, w_d x) \ : \ A^{(n)} x = b, \ x \text{ in } N^{nt} \}$$

Application 1: Multiway Tables

The margin equations for any $m_1 \times \cdots \times m_k \times n$ polytope form an n-fold system defined by a suitable 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.

Application 1: Multiway Tables

The margin equations for any $m_1 \times \cdots \times m_k \times n$ polytope form an n-fold system defined by a suitable 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.

We deduce the optimization theorem for long k-way polytopes:

Theorem: Fix $d, m_1, \ldots, m_k$. Then convex integer programming over any $m_1 \times \cdots \times m_k \times n$ multiway polytope is solvable in polynomial oracle-time for any margins, $w_1, \ldots, w_d$, and convex c presented by comparison oracle.

Application 1: Multiway Tables

The margin equations for any $m_1 \times \cdots \times m_k \times n$ polytope form an n-fold system defined by a suitable 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.

We deduce the optimization theorem for long k-way polytopes:

Theorem: Fix $d, m_1, \ldots, m_k$. Then convex integer programming over any $m_1 \times \cdots \times m_k \times n$ multiway polytope is solvable in polynomial oracle-time for any margins, $w_1, \ldots, w_d$, and convex c presented by comparison oracle.

Recall that in contrast, short 3-way polytopes are universal:

Theorem: Any rational polytope is an $r \times c \times 3$ line-sum 3-way polytope.

Application 2: Bin Packing

Shmuel Onn

Application 2: Bin Packing

Pack many items of several types into bins to maximize utility.
More precisely, there are t types of items, n_j items of type j of weight v_j each, and n bins with weight capacity u_k for bin k.

Application 2: Bin Packing

Pack many items of several types into bins to maximize utility.
More precisely, there are t types of items, n_j items of type j of weight v_j each, and n bins with weight capacity u_k for bin k.

In the linear problem, there is a utility matrix w with $w_{j,k}$ the utility of packing one item of type j in bin k. In the convex problem, there are d utility matrices and total utility is a suitable convex balancing.

Application 2: Bin Packing

Pack many items of several types into bins to maximize utility.
More precisely, there are t types of items, n_j items of type j of weight v_j each, and n bins with weight capacity u_k for bin k.

In the linear problem, there is a utility matrix w with $w_{j,k}$ the utility of packing one item of type j in bin k. In the convex problem, there are d utility matrices and total utility is a suitable convex balancing.

This can be shown to be an n-fold system defined by a $(t+1) \times t$ matrix A, where A_1 is the $t \times t$ identity matrix and $A_2 = (v_1, \ldots, v_t)$. **So we deduce**:

Application 2: Bin Packing

Pack many items of several types into bins to maximize utility.
More precisely, there are t types of items, n_j items of type j of weight v_j each, and n bins with weight capacity u_k for bin k.

In the linear problem, there is a utility matrix w with $w_{j,k}$ the utility of packing one item of type j in bin k. In the convex problem, there are d utility matrices and total utility is a suitable convex balancing.

This can be shown to be an n-fold system defined by a $(t+1) \times t$ matrix A, where A_1 is the $t \times t$ identity matrix and $A_2 = (v_1, \ldots, v_t)$. **So we deduce**:

Theorem: Fix d, t, $v_1, \ldots, v_t$. Then convex bin packing is polytime solvable.

Application 3: Partitioning Problems

Application 3: Partitioning Problems

Partition n items evaluated by k criteria to p players, to maximize social utility which is convex on the sums of values of items each player gets.

Application 3: Partitioning Problems

Partition n items evaluated by k criteria to p players, to maximize social utility which is convex on the sums of values of items each player gets.

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

Application 3: Partitioning Problems

Partition n items evaluated by k criteria to p players, to maximize social utility which is convex on the sums of values of items each player gets.

Example: Consider $n=6$ 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

Application 3: Partitioning Problems

Partition n items evaluated by k criteria to p players, to maximize social utility which is convex on the sums of values of items each player gets.

Example: Consider n=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

Application 3: Partitioning Problems

Partition n items evaluated by k criteria to p players, to maximize social utility which is convex on the sums of values of items each player gets.

Example: Consider n=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 convex functional on k x p matrices is $c(X) = \sum X_{ij}^3$

Application 3: Partitioning Problems

Partition n items evaluated by k criteria to p players, to maximize social utility which is convex on the sums of values of items each player gets.

Example: Consider n=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 convex functional on k x p matrices is $c(X) = \sum X_{ij}^3$

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

players

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

Application 3: Partitioning Problems

Partition n items evaluated by k criteria to p players, to maximize social utility which is convex on the sums of values of items each player gets.

Example: Consider n=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 convex functional on k x p matrices is $c(X) = \sum X_{ij}^3$

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

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

[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]

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)

[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]

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

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

[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]

This can be shown to be an n-fold system defined by a $(p+1) \times p$ matrix A, where A_1 is the $p \times p$ identity matrix and $A_2 = (1, \ldots, 1)$. **So we deduce**:

This can be shown to be an n-fold system defined by a $(p+1) \times p$ matrix A, where A_1 is the $p \times p$ identity matrix and $A_2 = (1, \ldots, 1)$. **So we deduce**:

Theorem: Partitioning problems with fixed p and k are polytime solvable.

Bibliography: most papers are available at

http://ie.technion.ac.il/~onn/Home-Page/selected-articles.html

Bibliography: most papers are available at

http://ie.technion.ac.il/~onn/Home-Page/selected-articles.html

Most relevant:

- Convex integer programming (in preparation)
- N-fold integer programming (submitted)
- All linear and integer programs are
  slim 3-way transportation programs (SIAM J. Opt., to appear)

Bibliography: most papers are available at

http://ie.technion.ac.il/~onn/Home-Page/selected-articles.html

Most relevant:

- Convex integer programming (in preparation)
- N-fold integer programming (submitted)
- All linear and integer programs are slim 3-way transportation programs (SIAM J. Opt., to appear)

Also related:

- Markov bases of three-way tables are arbitrarily complicated (J. Symb. Comp. 2006)
- Convex combinatorial optimization (Disc. Comp. Geom. 2004)
- The Hilbert zonotope and a polynomial time algorithm for universal Gröbner bases (Adv. App. Math. 2003)