

# Convex Discrete Optimization

Shmuel Onn

Lecture Series, Université de Montréal, June 2006

Technion - Israel Institute of Technology

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

Based on several papers joint with various authors among  
De Loera, Hemmecke, Rothblum, Weismantel

Research Supported in part by ISF - Israel Science Foundation

# Lecture Series Outline

# Lecture Series Outline

## Part 1. Convex Combinatorial Optimization:

Lecture 1. The Setup; Applications; Outline of the Approach.

Lecture 2. Proof of the Main Theorem: the Algorithm.

# Lecture Series Outline

## Part 1. Convex Combinatorial Optimization:

Lecture 1. The Setup; Applications; Outline of the Approach.

Lecture 2. Proof of the Main Theorem: the Algorithm.

## Part 2. Convex Integer Programming:

Lecture 3. The Problem; Applications; The Theorems.

Lecture 4. Proofs and Algorithms.

# Lecture Series Outline

## Part 1. Convex Combinatorial Optimization:

Lecture 1. The Setup; Applications; Outline of the Approach.

Lecture 2. Proof of the Main Theorem: the Algorithm.

## Part 2. Convex Integer Programming:

Lecture 3. The Problem; Applications; The Theorems.

Lecture 4. Proofs and Algorithms.

Lecture 5: Universality of Multiway Transportation.

# Part 1. Convex Combinatorial Optimization

Lecture 1. The Setup; Applications; Outline of the Approach.

# Linear Combinatorial Optimization (LCO)

LCO: Given family  $\mathcal{F} \subseteq 2^N$  of subsets of  $N = \{1, \dots, n\}$  and real weighting  $w: N \rightarrow \mathbb{R}$ , find  $F \in \mathcal{F}$  of maximum weight  $w(F) = \sum_{j \in F} w(j)$

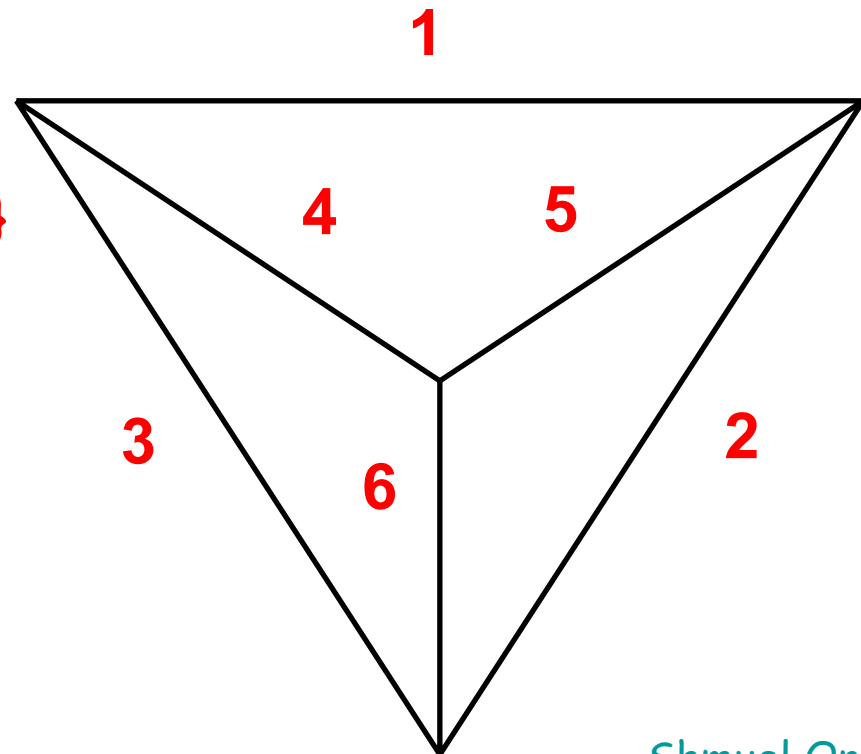
# Linear Combinatorial Optimization (LCO)

LCO: Given family  $F \subseteq 2^N$  of subsets of  $N = \{1, \dots, n\}$  and real weighting  $w: N \rightarrow \mathbb{R}$ , find  $F \in F$  of maximum weight  $w(F) = \sum_{j \in F} w(j)$

Example: Spanning Trees

e.g.  $n=6$ , graph  $G=K_4$  gives the family

$F = \{\{1,2,4\}, \{1,2,5\}, \dots, \{3,5,6\}, \{4,5,6\}\}$



# Linear Combinatorial Optimization (LCO)

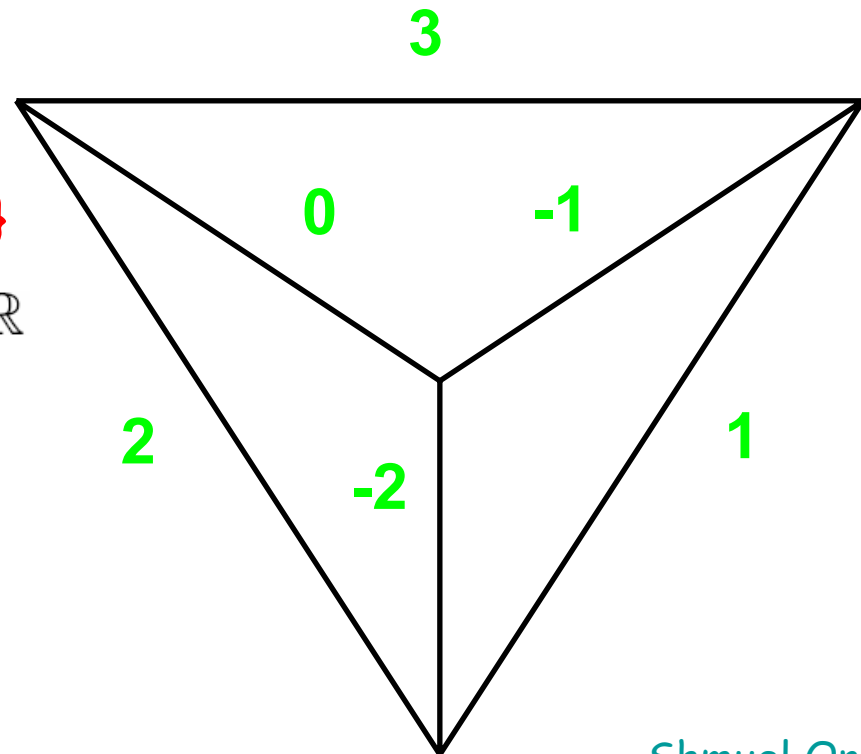
LCO: Given family  $F \subseteq 2^N$  of subsets of  $N = \{1, \dots, n\}$  and real weighting  $w: N \rightarrow \mathbb{R}$ , find  $F \in F$  of maximum weight  $w(F) = \sum_{j \in F} w(j)$

Example: Spanning Trees

e.g.  $n=6$ , graph  $G=K_4$  gives the family

$F = \{\{1,2,4\}, \{1,2,5\}, \dots, \{3,5,6\}, \{4,5,6\}\}$

Now consider weighting  $w: \{1, \dots, 6\} \rightarrow \mathbb{R}$



# Linear Combinatorial Optimization (LCO)

LCO: Given family  $F \subseteq 2^N$  of subsets of  $N = \{1, \dots, n\}$  and real weighting  $w: N \rightarrow \mathbb{R}$ , find  $F \in F$  of maximum weight  $w(F) = \sum_{j \in F} w(j)$

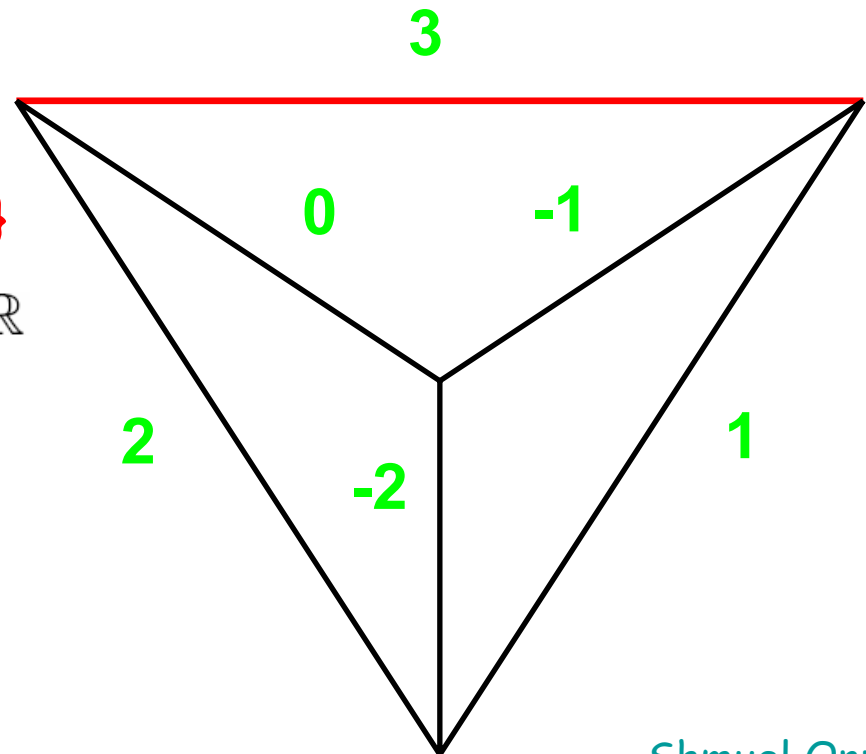
## Example: Spanning Trees

e.g.  $n=6$ , graph  $G=K_4$  gives the family

$F = \{\{1,2,4\}, \{1,2,5\}, \dots, \{3,5,6\}, \{4,5,6\}\}$

Now consider weighting  $w: \{1, \dots, 6\} \rightarrow \mathbb{R}$

The maximum weight tree is easily obtained by the greedy algorithm



# Linear Combinatorial Optimization (LCO)

**LCO:** Given family  $F \subseteq 2^N$  of subsets of  $N = \{1, \dots, n\}$  and real weighting  $w: N \rightarrow \mathbb{R}$ , find  $F \in F$  of maximum weight  $w(F) = \sum_{j \in F} w(j)$

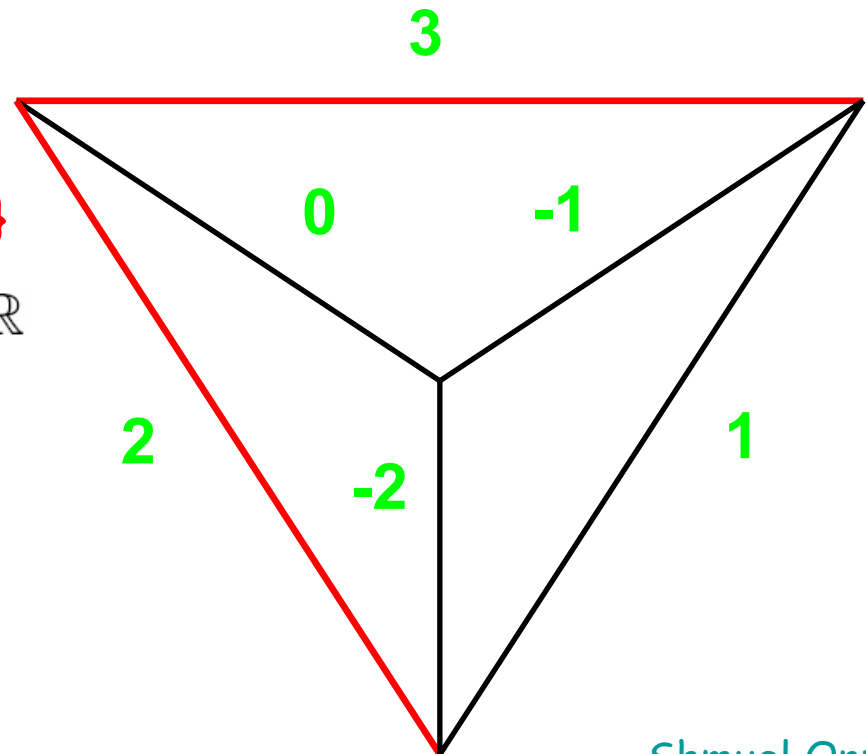
## Example: Spanning Trees

e.g.  $n=6$ , graph  $G=K_4$  gives the family

$F = \{\{1,2,4\}, \{1,2,5\}, \dots, \{3,5,6\}, \{4,5,6\}\}$

Now consider weighting  $w: \{1, \dots, 6\} \rightarrow \mathbb{R}$

The maximum weight tree is easily obtained by the greedy algorithm



# Linear Combinatorial Optimization (LCO)

**LCO:** Given family  $F \subseteq 2^N$  of subsets of  $N = \{1, \dots, n\}$  and real weighting  $w: N \rightarrow \mathbb{R}$ , find  $F \in F$  of maximum weight  $w(F) = \sum_{j \in F} w(j)$

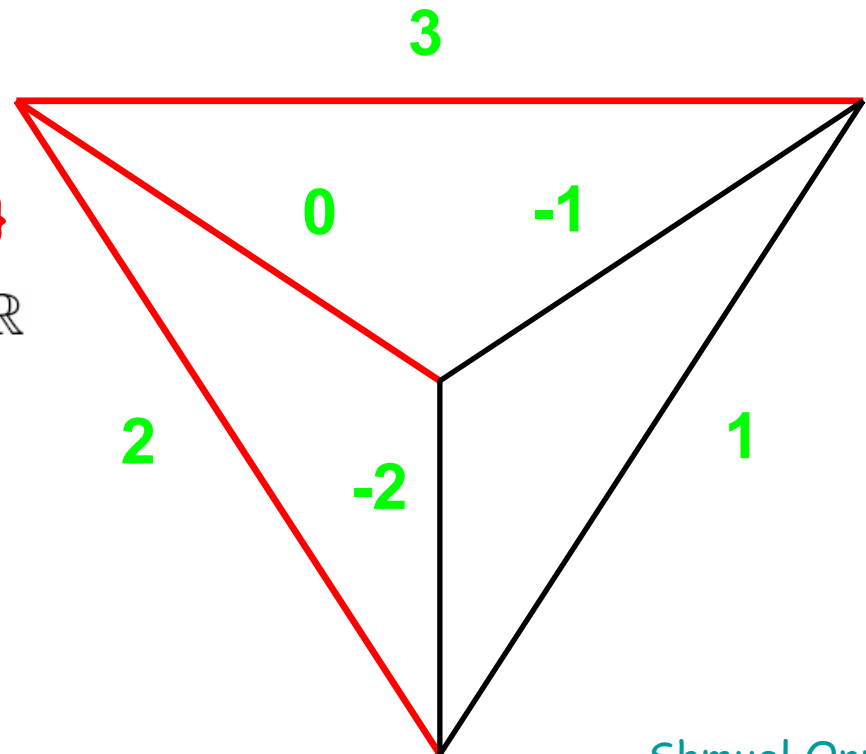
## Example: Spanning Trees

e.g.  $n=6$ , graph  $G=K_4$  gives the family

$F = \{\{1,2,4\}, \{1,2,5\}, \dots, \{3,5,6\}, \{4,5,6\}\}$

Now consider weighting  $w: \{1, \dots, 6\} \rightarrow \mathbb{R}$

The maximum weight tree is easily obtained by the greedy algorithm



# Convex Combinatorial Optimization (CCO)

CCO: Given family  $\mathcal{F} \subseteq 2^{\mathcal{N}}$ , vectorial weighting  $w: \mathcal{N} \rightarrow \mathbb{R}^d$  and convex functional  $c: \mathbb{R}^d \rightarrow \mathbb{R}$ , find  $F \in \mathcal{F}$  of maximum value  $c(w(F))$ .

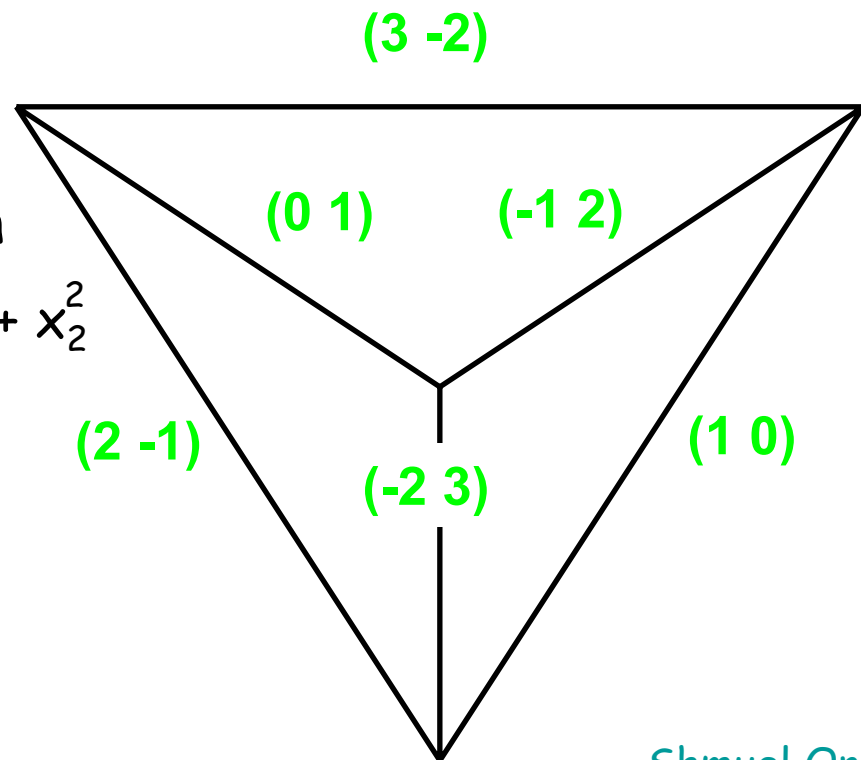
# Convex Combinatorial Optimization (CCO)

CCO: Given family  $\mathcal{F} \subseteq 2^{\mathcal{N}}$ , vectorial weighting  $w: \mathcal{N} \rightarrow \mathbb{R}^d$  and convex functional  $c: \mathbb{R}^d \rightarrow \mathbb{R}$ , find  $F \in \mathcal{F}$  of maximum value  $c(w(F))$ .

## Example: Spanning Trees

Consider again  $n=6$ , the graph  $G=K_4$ ,  $d=2$ , weighting  $w$  and convex function

$c: \mathbb{R}^2 \rightarrow \mathbb{R}$  defined by  $c(x) = |x|^2 = x_1^2 + x_2^2$



# Convex Combinatorial Optimization (CCO)

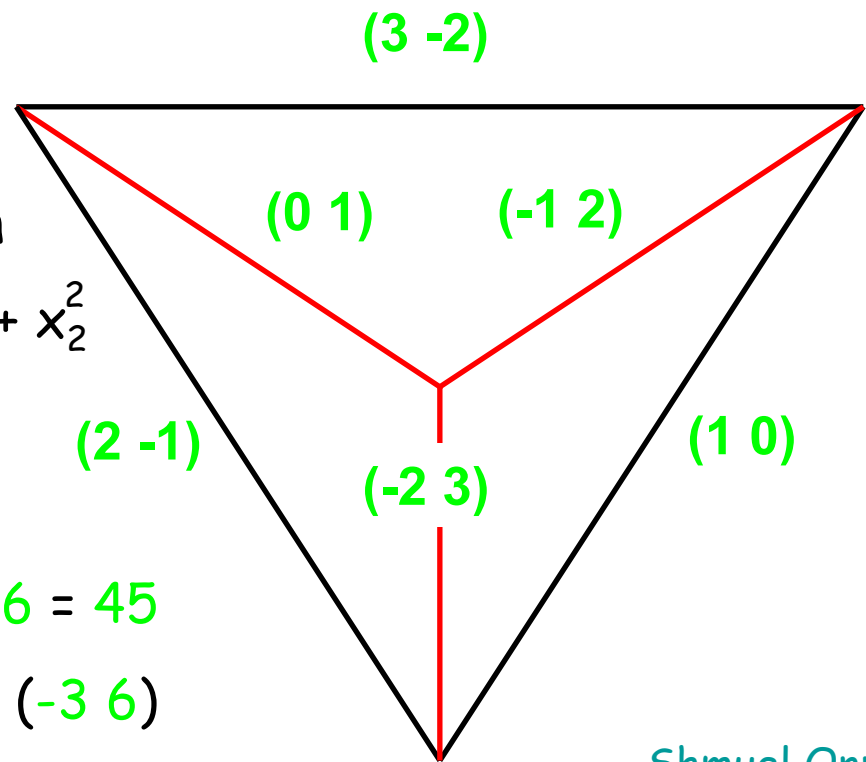
CCO: Given family  $\mathcal{F} \subseteq 2^{\mathcal{N}}$ , vectorial weighting  $w: \mathcal{N} \rightarrow \mathbb{R}^d$  and convex functional  $c: \mathbb{R}^d \rightarrow \mathbb{R}$ , find  $F \in \mathcal{F}$  of maximum value  $c(w(F))$ .

## Example: Spanning Trees

Consider again  $n=6$ , the graph  $G=K_4$ ,  $d=2$ , weighting  $w$  and convex function

$c: \mathbb{R}^2 \rightarrow \mathbb{R}$  defined by  $c(x) = |x|^2 = x_1^2 + x_2^2$

The objective value of the optimal tree  $F^*$  is  $c(w(F^*)) = c((-3 \ 6)) = 9 + 36 = 45$  where  $w(F^*) = (0 \ 1) + (-1 \ 2) + (-2 \ 3) = (-3 \ 6)$



## Some bad news, good news, and questions

**LCO:** Given family  $\mathcal{F} \subseteq 2^N$  of subsets of  $N = \{1, \dots, n\}$  and real weighting  $w: N \rightarrow \mathbb{R}$ , find  $F \in \mathcal{F}$  of maximum weight  $w(F) = \sum_{j \in F} w(j)$

**CCO:** Given family  $\mathcal{F} \subseteq 2^N$ , vectorial weighting  $w: N \rightarrow \mathbb{R}^d$  and convex functional  $c: \mathbb{R}^d \rightarrow \mathbb{R}$ , find  $F \in \mathcal{F}$  of maximum value  $c(w(F))$ .

## Some bad news, good news, and questions

**LCO:** Given family  $\mathcal{F} \subseteq 2^N$  of subsets of  $N = \{1, \dots, n\}$  and real weighting  $w: N \rightarrow \mathbb{R}$ , find  $F \in \mathcal{F}$  of maximum weight  $w(F) = \sum_{j \in F} w(j)$

**CCO:** Given family  $\mathcal{F} \subseteq 2^N$ , vectorial weighting  $w: N \rightarrow \mathbb{R}^d$  and convex functional  $c: \mathbb{R}^d \rightarrow \mathbb{R}$ , find  $F \in \mathcal{F}$  of maximum value  $c(w(F))$ .

☺ Very broad expressive power

## Some bad news, good news, and questions

**LCO:** Given family  $\mathcal{F} \subseteq 2^N$  of subsets of  $N = \{1, \dots, n\}$  and real weighting  $w: N \rightarrow \mathbb{R}$ , find  $F \in \mathcal{F}$  of maximum weight  $w(F) = \sum_{j \in F} w(j)$

**CCO:** Given family  $\mathcal{F} \subseteq 2^N$ , vectorial weighting  $w: N \rightarrow \mathbb{R}^d$  and convex functional  $c: \mathbb{R}^d \rightarrow \mathbb{R}$ , find  $F \in \mathcal{F}$  of maximum value  $c(w(F))$ .

- 😊 Very broad expressive power
- 😞 Generally intractable even for  $d=1$

## Some bad news, good news, and questions

**LCO:** Given family  $\mathcal{F} \subseteq 2^N$  of subsets of  $N = \{1, \dots, n\}$  and real weighting  $w: N \rightarrow \mathbb{R}$ , find  $F \in \mathcal{F}$  of maximum weight  $w(F) = \sum_{j \in F} w(j)$

**CCO:** Given family  $\mathcal{F} \subseteq 2^N$ , vectorial weighting  $w: N \rightarrow \mathbb{R}^d$  and convex functional  $c: \mathbb{R}^d \rightarrow \mathbb{R}$ , find  $F \in \mathcal{F}$  of maximum value  $c(w(F))$ .

- 😊 Very broad expressive power
- 😞 Generally intractable even for  $d=1$
- 😞 For variable  $d$ , intractable even for  $\mathcal{F} = 2^N$

## Some bad news, good news, and questions

**LCO:** Given family  $\mathcal{F} \subseteq 2^N$  of subsets of  $N = \{1, \dots, n\}$  and real weighting  $w: N \rightarrow \mathbb{R}$ , find  $F \in \mathcal{F}$  of maximum weight  $w(F) = \sum_{j \in F} w(j)$

**CCO:** Given family  $\mathcal{F} \subseteq 2^N$ , vectorial weighting  $w: N \rightarrow \mathbb{R}^d$  and convex functional  $c: \mathbb{R}^d \rightarrow \mathbb{R}$ , find  $F \in \mathcal{F}$  of maximum value  $c(w(F))$ .

- 😊 Very broad expressive power
- 😞 Generally intractable even for  $d=1$
- 😞 For variable  $d$ , intractable even for  $\mathcal{F}=2^N$
- 😊 We provide broad polynomial time solvable setup

## Some bad news, good news, and questions

**LCO:** Given family  $\mathcal{F} \subseteq 2^N$  of subsets of  $N = \{1, \dots, n\}$  and real weighting  $w: N \rightarrow \mathbb{R}$ , find  $F \in \mathcal{F}$  of maximum weight  $w(F) = \sum_{j \in F} w(j)$

**CCO:** Given family  $\mathcal{F} \subseteq 2^N$ , vectorial weighting  $w: N \rightarrow \mathbb{R}^d$  and convex functional  $c: \mathbb{R}^d \rightarrow \mathbb{R}$ , find  $F \in \mathcal{F}$  of maximum value  $c(w(F))$ .

- 😊 Very broad expressive power
- 😞 Generally intractable even for  $d=1$
- 😞 For variable  $d$ , intractable even for  $\mathcal{F}=2^N$
- 😊 We provide broad polynomial time solvable setup
- ? Approximation algorithms

## Some bad news, good news, and questions

**LCO:** Given family  $\mathcal{F} \subseteq 2^N$  of subsets of  $N = \{1, \dots, n\}$  and real weighting  $w: N \rightarrow \mathbb{R}$ , find  $F \in \mathcal{F}$  of maximum weight  $w(F) = \sum_{j \in F} w(j)$

**CCO:** Given family  $\mathcal{F} \subseteq 2^N$ , vectorial weighting  $w: N \rightarrow \mathbb{R}^d$  and convex functional  $c: \mathbb{R}^d \rightarrow \mathbb{R}$ , find  $F \in \mathcal{F}$  of maximum value  $c(w(F))$ .

- 😊 Very broad expressive power
- 😞 Generally intractable even for  $d=1$
- 😞 For variable  $d$ , intractable even for  $\mathcal{F}=2^N$
- 😊 We provide broad polynomial time solvable setup
- ? Approximation algorithms
- ? Classification of "edge-well-behaved" families  $\mathcal{F}$

## Some bad news, good news, and questions

**LCO:** Given family  $\mathcal{F} \subseteq 2^N$  of subsets of  $N = \{1, \dots, n\}$  and real weighting  $w: N \rightarrow \mathbb{R}$ , find  $F \in \mathcal{F}$  of maximum weight  $w(F) = \sum_{j \in F} w(j)$

**CCO:** Given family  $\mathcal{F} \subseteq 2^N$ , vectorial weighting  $w: N \rightarrow \mathbb{R}^d$  and convex functional  $c: \mathbb{R}^d \rightarrow \mathbb{R}$ , find  $F \in \mathcal{F}$  of maximum value  $c(w(F))$ .

- 😊 Very broad expressive power
- 😞 Generally intractable even for  $d=1$
- 😞 For variable  $d$ , intractable even for  $\mathcal{F}=2^N$
- 😊 We provide broad polynomial time solvable setup
- ? Approximation algorithms
- ? Classification of "edge-well-behaved" families  $\mathcal{F}$
- ? Convex minimization

# Application 1: Matroids

# Application 1: Matroids

Here  $F$  is the family of **independent sets** or **bases** of a matroid over  $N$  presented by a **membership oracle**.

# Application 1: Matroids

Here  $F$  is the family of **independent sets** or **bases** of a matroid over  $N$  presented by a **membership oracle**.

In particular, includes the **spanning trees** of before.

# Application 1: Matroids

Here  $F$  is the family of **independent sets** or **bases** of a matroid over  $N$  presented by a **membership oracle**.

In particular, includes the **spanning trees** of before.

**Corollary 1:** **CCO** over **matroids** is **polynomially solvable**

(studied by Hassin and Tamir, Onn)

## Application 2: Positive Semidefinite Quadratic Assignment

Quadratic Assignment: Given  $n \times n$  matrix  $M$ ,  
find vector  $x \in \{0,1\}^n$  maximizing  $x^T M x$

## Application 2: Positive Semidefinite Quadratic Assignment

**Quadratic Assignment:** Given  $n \times n$  matrix  $M$ ,  
find vector  $x \in \{0,1\}^n$  maximizing  $x^T M x$

**Special Case (yet NP-hard):**  $M$  positive semidefinite of rank  $d$ ,  
so  $M = W^T W$  with  $W$  given  $d \times n$  matrix

## Application 2: Positive Semidefinite Quadratic Assignment

**Quadratic Assignment:** Given  $n \times n$  matrix  $M$ ,  
find vector  $x \in \{0,1\}^n$  maximizing  $x^T M x$

**Special Case (yet NP-hard):**  $M$  positive semidefinite of rank  $d$ ,  
so  $M = W^T W$  with  $W$  given  $d \times n$  matrix

### Formulation as CCO:

$\mathcal{F} = 2^N$  is the family of all subsets

$w(j) = W^j$  is the  $j$ -th column of  $W$

$c: \mathbb{R}^d \rightarrow \mathbb{R}$  is defined by  $c(x) = |x|^2 = x_1^2 + \dots + x_d^2$

## Application 2: Positive Semidefinite Quadratic Assignment

**Quadratic Assignment:** Given  $n \times n$  matrix  $M$ ,  
find vector  $x \in \{0,1\}^n$  maximizing  $x^T M x$

**Special Case (yet NP-hard):**  $M$  positive semidefinite of rank  $d$ ,  
so  $M = W^T W$  with  $W$  given  $d \times n$  matrix

### Formulation as CCO:

$\mathcal{F} = 2^N$  is the family of all subsets

$w(j) = W^j$  is the  $j$ -th column of  $W$

$c: \mathbb{R}^d \rightarrow \mathbb{R}$  is defined by  $c(x) = |x|^2 = x_1^2 + \dots + x_d^2$

**Corollary 2:** PSD Quadratic Assignment is polynomially solvable

(studied by Allemand, Fukuda, Liebling and Steiner)

# Application 3: Partitioning Problems

## Application 3: Partitioning Problems

Partition  $m$  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  $m$  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  $m=6$  items,  $k=2$  criteria,  $p=3$  players







# Application 3: Partitioning Problems

Partition  $m$  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  $m=6$  items,  $k=2$  criteria,  $p=3$  players

The criteria-item matrix is:

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

Each player should get 2 items

The convex functional on  $k \times p$  matrices is  $c(X) = \sum X_{ij}^3$

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

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



All 90 partitions  $\Pi$   
of items  $\{1, \dots, 6\}$  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 ]

All 90 partitions  $\pi$   
of items  $\{1, \dots, 6\}$  To  
3 players where each  
player gets 2 items

The optimal partition is:  
 $\pi = (34, 56, 12)$

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

All 90 partitions  $\pi$  of items  $\{1, \dots, 6\}$  To 3 players where each player gets 2 items

The optimal partition is:  
 $\pi = (34, 56, 12)$

with optimal utility:

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

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

[ 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 ]

## Formulation as CCO:

Define  $n=mp$ ,  $d=kp$

The ground set is  $N = \{ (i,j) : i=1, \dots, m, j=1, \dots, p \}$

## Formulation as CCO:

Define  $n=mp$ ,  $d=kp$

The ground set is  $N = \{ (i,j) : i=1, \dots, m, j=1, \dots, p \}$

Each partition  $\pi = (\pi_1, \dots, \pi_p)$  is encoded as  $F_\pi = \{ (i,j) : i \in \pi_j \}$ ;  
 $\mathcal{F} \subseteq 2^N$  consists of all  $F_\pi$  of partitions  $\pi$  obeying **restrictions**

## Formulation as CCO:

Define  $n=mp$ ,  $d=kp$

The ground set is  $N = \{ (i,j) : i=1, \dots, m, j=1, \dots, p \}$

Each partition  $\pi = (\pi_1, \dots, \pi_p)$  is encoded as  $F_\pi = \{ (i,j) : i \in \pi_j \}$ ;

$\mathcal{F} \subseteq 2^N$  consists of all  $F_\pi$  of partitions  $\pi$  obeying restrictions

The weight function  $w:N \rightarrow \mathbb{R}^d \cong \mathbb{R}^{k \times p}$  is given by

$$w(i,j) = A^i \otimes 1_j = \underbrace{\left( \begin{array}{c|c|c} & \overset{j}{\text{---}} & \\ \hline 0 & A^i & 0 \\ \hline \end{array} \right)}_p \Bigg\}^k \quad i=1, \dots, m, \quad j=1, \dots, p$$

The convex functional  $c$  on  $k \times p$  matrices is as given.

## Formulation as CCO:

Define  $n=mp$ ,  $d=kp$

The ground set is  $N = \{ (i,j) : i=1, \dots, m, j=1, \dots, p \}$

Each partition  $\pi = (\pi_1, \dots, \pi_p)$  is encoded as  $F_\pi = \{ (i,j) : i \in \pi_j \}$ ;  
 $\mathcal{F} \subseteq 2^N$  consists of all  $F_\pi$  of partitions  $\pi$  obeying restrictions

The weight function  $w:N \rightarrow \mathbb{R}^d \cong \mathbb{R}^{k \times p}$  is given by

$$w(i,j) = A^i \otimes 1_j = \underbrace{\left( \begin{array}{c|c|c} & j & \\ \hline 0 & A^i & 0 \\ \hline & & \end{array} \right)}_p \Bigg\}^k \quad i=1, \dots, m, j=1, \dots, p$$

The convex functional  $c$  on  $k \times p$  matrices is as given.

**Corollary 3:** Partitioning problems are polynomially solvable

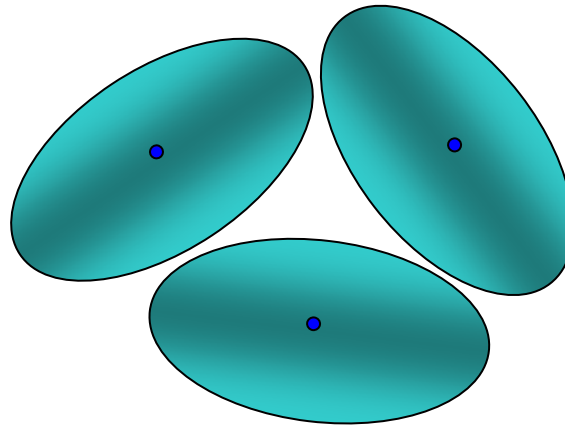
(studied by Aviran, Granot, Barnes, Hoffman, Hwang, Onn, Rothblum, and more)

## Application 4: Minimum Variance Clustering

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

## Application 4: Minimum Variance Clustering

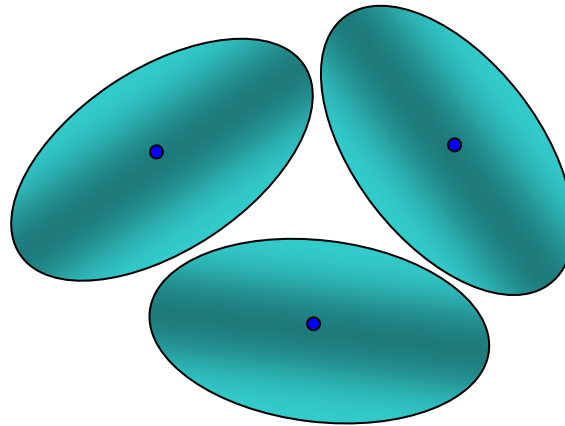
Given  $m$  points  $v^1, \dots, v^m$  in  $\mathbb{R}^k$ , group them into  $p$  (balanced) clusters so as to minimize the sum of cluster variances.



$P=3, k=3, m$  large

## Application 4: Minimum Variance Clustering

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



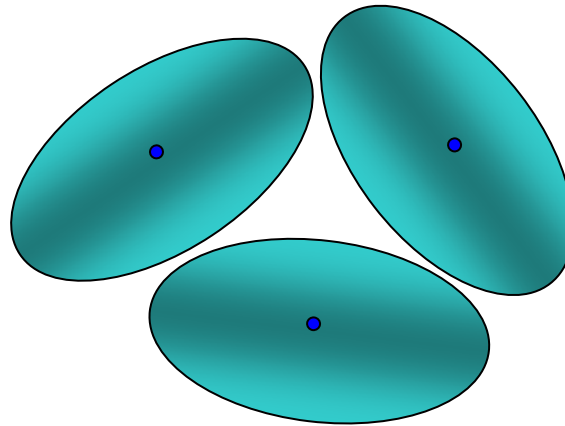
$P=3, k=3, m$  large

Formulation as CCO:

Special case of partitioning, hence CCO problem with  $n=mp$ ,  $d=kp$ , weighting  $w$  of elements by  $k \times p$  matrices, and convex function  $c$  assigning to each  $k \times p$  matrix  $X$  its Euclidean norm  $c(X) = \sum X_{ij}^2$

## Application 4: Minimum Variance Clustering

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



$P=3, k=3, m$  large

### Formulation as CCO:

Special case of partitioning, hence CCO problem with  $n=mp$ ,  $d=kp$ , weighting  $w$  of elements by  $k \times p$  matrices, and convex function  $c$  assigning to each  $k \times p$  matrix  $X$  its Euclidean norm  $c(X) = \sum X_{ij}^2$

**Corollary 4:** Minimum variance clustering is polynomially solvable

(numerous applications in the analysis of statistical data and other areas)

# The Geometric Approach to Combinatorial Optimization

# The Geometric Approach to Combinatorial Optimization

Denote by  $1_j$  the  $j$ -th unit vector in  $\mathbb{R}^n$ .

## The Geometric Approach to Combinatorial Optimization

Denote by  $1_j$  the  $j$ -th unit vector in  $\mathbb{R}^n$ .

For a subset  $F \subseteq \mathbb{N}$  denote by  $1_F = \sum_{j \in F} 1_j$  its indicator.

# The Geometric Approach to Combinatorial Optimization

Denote by  $1_j$  the  $j$ -th unit vector in  $\mathbb{R}^n$ .

For a subset  $F \subseteq \mathbb{N}$  denote by  $1_F = \sum_{j \in F} 1_j$  its indicator.

The **family polytope** of  $\mathcal{F} \subseteq 2^{\mathbb{N}}$  is the 0-1 polytope

$$P^{\mathcal{F}} = \text{conv} \{ 1_F : F \in \mathcal{F} \}$$

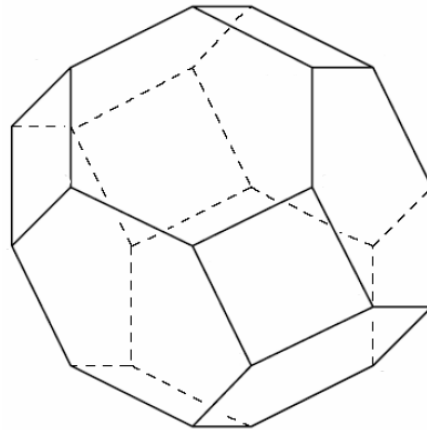
# The Geometric Approach to Combinatorial Optimization

Denote by  $1_j$  the  $j$ -th unit vector in  $\mathbb{R}^n$ .

For a subset  $F \subseteq \mathbb{N}$  denote by  $1_F = \sum_{j \in F} 1_j$  its indicator.

The **family polytope** of  $\mathcal{F} \subseteq 2^{\mathbb{N}}$  is the 0-1 polytope

$$P^{\mathcal{F}} = \text{conv} \{ 1_F : F \in \mathcal{F} \}$$



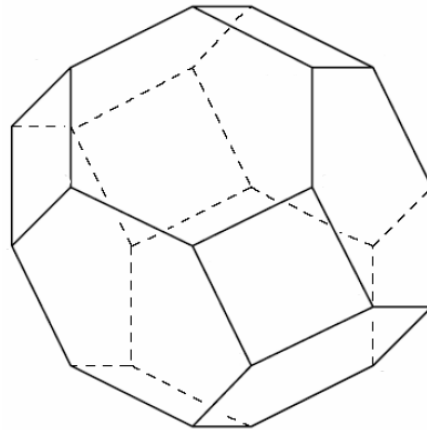
# The Geometric Approach to Combinatorial Optimization

Denote by  $1_j$  the  $j$ -th unit vector in  $\mathbb{R}^n$ .

For a subset  $F \subseteq \mathbb{N}$  denote by  $1_F = \sum_{j \in F} 1_j$  its indicator.

The **family polytope** of  $\mathcal{F} \subseteq 2^{\mathbb{N}}$  is the 0-1 polytope

$$P^{\mathcal{F}} = \text{conv} \{ 1_F : F \in \mathcal{F} \}$$



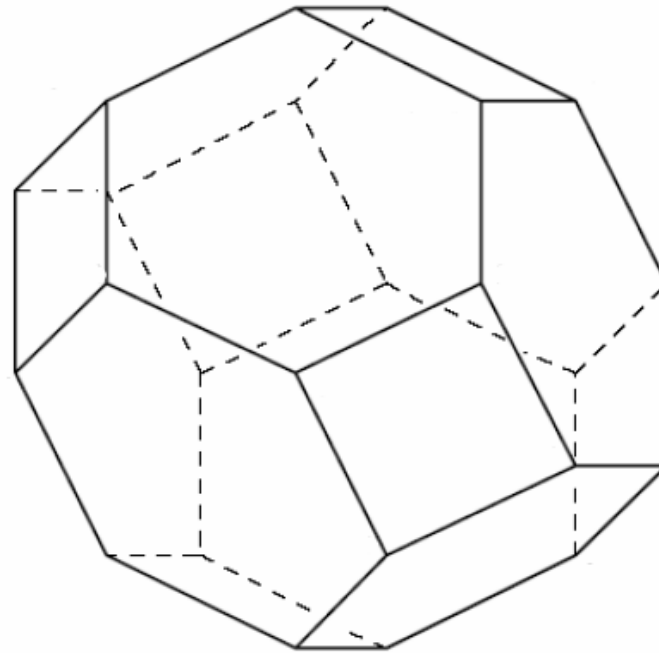
The **linear and convex combinatorial optimization** problems now ask for the **best vertex** of  $P^{\mathcal{F}}$

# The Grötschel - Lovász - Schrijver Theory

If  $P^F$  is facet-well-behaved then linear combinatorial optimization (LCO) over the family  $F$  is solvable in polynomial time.

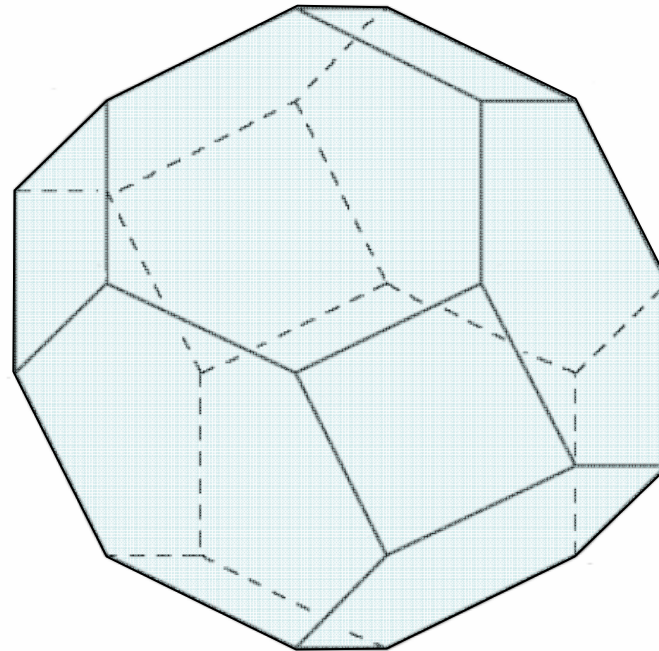
# The Grötschel - Lovász - Schrijver Theory

If  $P^F$  is facet-well-behaved then linear combinatorial optimization (LCO) over the family  $F$  is solvable in polynomial time.



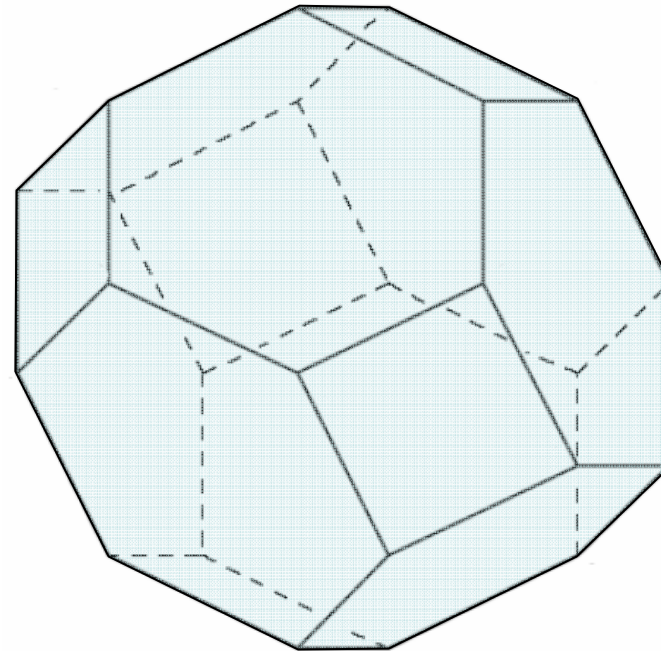
# The Grötschel - Lovász - Schrijver Theory

If  $P^F$  is facet-well-behaved then linear combinatorial optimization (LCO) over the family  $F$  is solvable in polynomial time.



# The Grötschel - Lovász - Schrijver Theory

If  $P^F$  is **facet-well-behaved** then **linear combinatorial optimization (LCO)** over the family  $F$  is solvable in **polynomial time**.



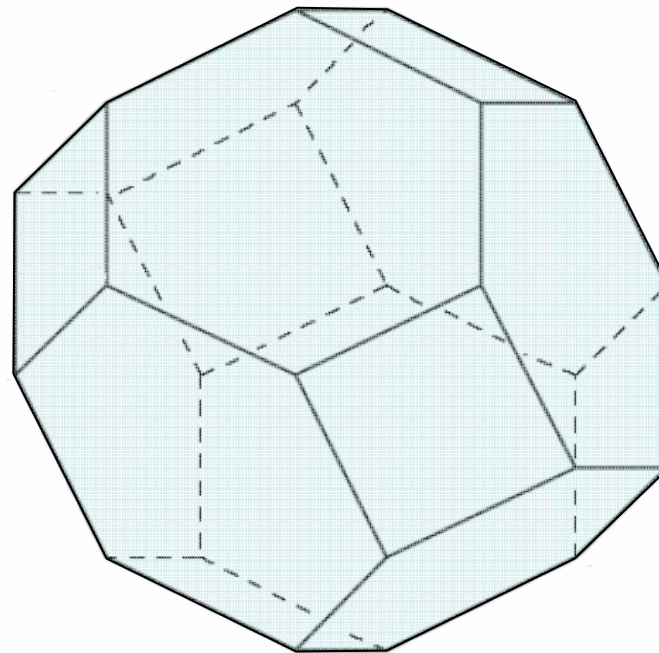
The proof makes use of:

- Ellipsoid method  
(equivalence of **separation** and **optimization**)

# The Grötschel - Lovász - Schrijver Theory

If  $P^F$  is **facet-well-behaved** then **linear combinatorial optimization (LCO)** over the family  $F$  is solvable in **polynomial time**.

Leonid Khachiyan  
1952 - 2005



The proof makes use of:

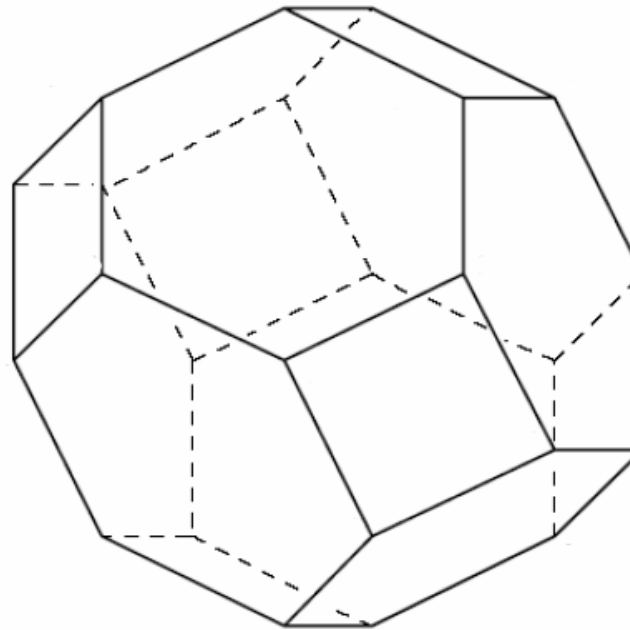
- Ellipsoid method  
(equivalence of **separation** and **optimization**)

## Our Work

If  $P^F$  is **edge-well-behaved** then **convex combinatorial optimization (CCO)** over the family  $F$  is solvable in **strongly polynomial time**.

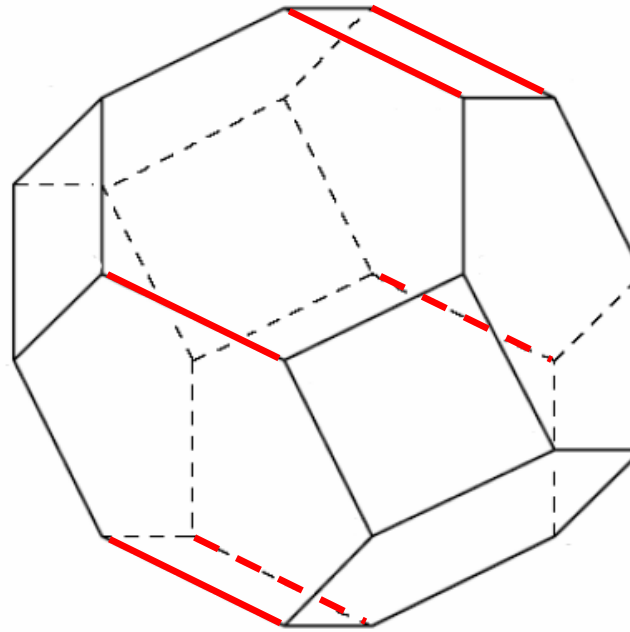
## Our Work

If  $P^F$  is **edge-well-behaved** then **convex combinatorial optimization (CCO)** over the family  $F$  is solvable in **strongly polynomial time**.



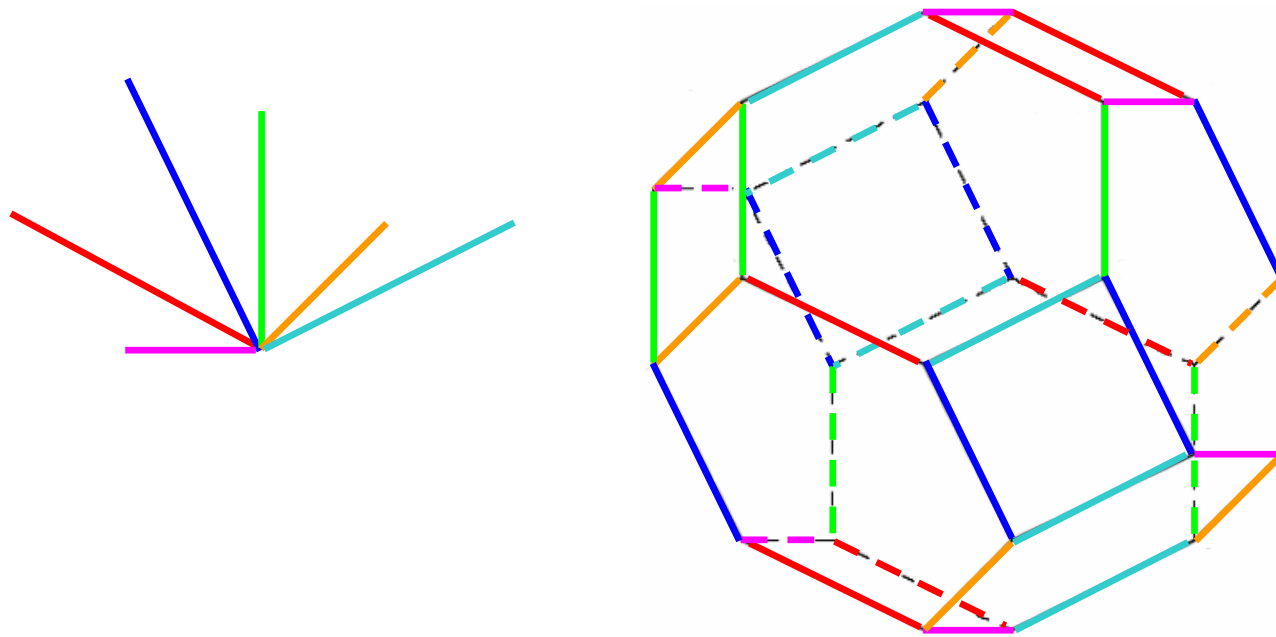
## Our Work

If  $P^F$  is **edge-well-behaved** then **convex combinatorial optimization (CCO)** over the family  $F$  is solvable in **strongly polynomial time**.



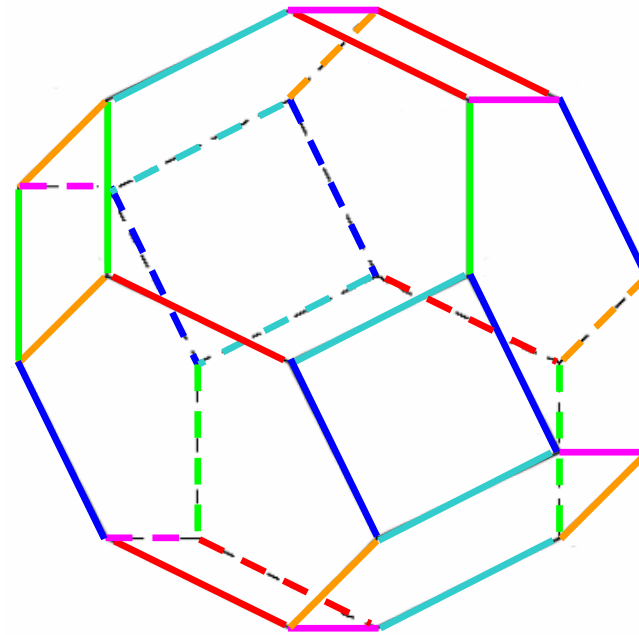
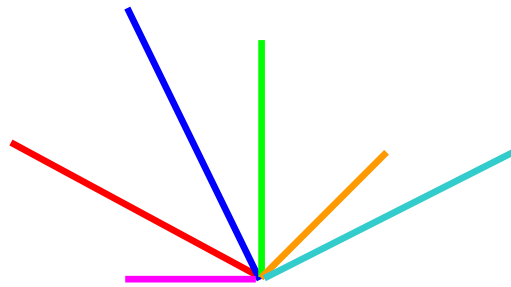
## Our Work

If  $P^F$  is **edge-well-behaved** then **convex combinatorial optimization (CCO)** over the family  $F$  is solvable in **strongly polynomial time**.



## Our Work

If  $P^F$  is **edge-well-behaved** then **convex combinatorial optimization (CCO)** over the family  $F$  is solvable in **strongly polynomial time**.



The proof makes use of:

- Zonotope construction
- Equivalence of augmentation and optimization  
(via scaling and diophantine approximation)

## The Main CCO Theorem

A family  $F$  is **edge-guaranteed** if it comes with a set of vectors

$E = \{e^1, \dots, e^m\} \subseteq \mathbb{R}^n$  containing **every edge direction** of the polytope  $P^F$ .

## The Main CCO Theorem

A family  $F$  is **edge-guaranteed** if it comes with a set of vectors  $E = \{e^1, \dots, e^m\} \subseteq \mathbb{R}^n$  containing **every edge direction** of the polytope  $P^F$ .

Typically, have **edge-well-behaved classes**  $\mathcal{C} = (F_n)$  of families having **"uniform"** edge-direction sets  $E_n = \{e^1, \dots, e^{m(n)}\}$  with  $m(n)$  polynomial in  $n$ .

## The Main CCO Theorem

A family  $F$  is **edge-guaranteed** if it comes with a set of vectors  $E = \{e^1, \dots, e^m\} \subseteq \mathbb{R}^n$  containing **every edge direction** of the polytope  $P^F$ .

Typically, have **edge-well-behaved classes**  $\mathcal{C} = (F_n)$  of families having "uniform" edge-direction sets  $E_n = \{e^1, \dots, e^{m(n)}\}$  with  $m(n)$  polynomial in  $n$ .

Now, recall **CCO**: Given family  $F \subseteq 2^N$ , vectorial weighting  $w: N \rightarrow \mathbb{R}^d$ , and convex functional  $c: \mathbb{R}^d \rightarrow \mathbb{R}$ , **find**  $F \in F$  of maximum value  $c(w(F))$ .

## The Main CCO Theorem

A family  $F$  is **edge-guaranteed** if it comes with a set of vectors  $E = \{e^1, \dots, e^m\} \subseteq \mathbb{R}^n$  containing **every edge direction** of the polytope  $P^F$ .

Typically, have **edge-well-behaved classes**  $\mathcal{C} = (F_n)$  of families having "uniform" edge-direction sets  $E_n = \{e^1, \dots, e^{m(n)}\}$  with  $m(n)$  polynomial in  $n$ .

Now, recall **CCO**: Given family  $F \subseteq 2^N$ , vectorial weighting  $w: N \rightarrow \mathbb{R}^d$ , and convex functional  $c: \mathbb{R}^d \rightarrow \mathbb{R}$ , **find**  $F \in F$  of maximum value  $c(w(F))$ .

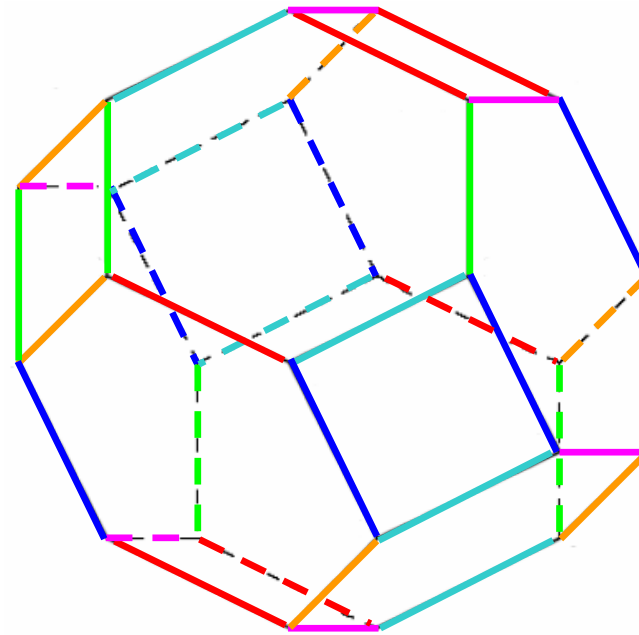
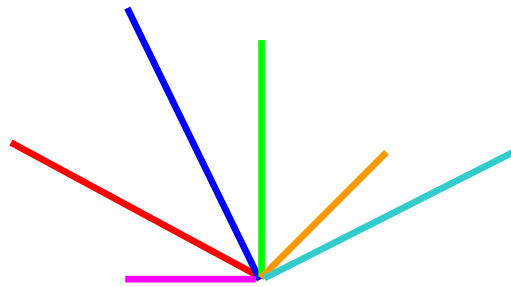
**Theorem:** Fix  $d$ . Then **convex combinatorial optimization (CCO)** is **strongly polynomially time solvable** over any **edge-guaranteed family**  $F$  given by a **membership oracle** and convex  $c$  given by an **evaluation oracle**.

## Part 1. Convex Combinatorial Optimization

Lecture 2. Proof of the Main Theorem: the Algorithm.

## Recall: Our Work

If  $P^F$  is **edge-well-behaved** then **convex combinatorial optimization (CCO)** over the family  $F$  is solvable in **strongly polynomial time**.



The proof makes use of:

- Zonotope construction
- Equivalence of augmentation and optimization  
(via scaling and diophantine approximation)

## Recall: The Main CCO Theorem

A family  $F$  is **edge-guaranteed** if it comes with a set of vectors  $E = \{e^1, \dots, e^m\} \subseteq \mathbb{R}^n$  containing **every edge direction** of the polytope  $P^F$ .

Typically, have **edge-well-behaved classes**  $\mathcal{C} = (F_n)$  of families having "uniform" edge-direction sets  $E_n = \{e^1, \dots, e^{m(n)}\}$  with  $m(n)$  polynomial in  $n$ .

Now, recall **CCO**: Given family  $F \subseteq 2^N$ , vectorial weighting  $w: N \rightarrow \mathbb{R}^d$ , and convex functional  $c: \mathbb{R}^d \rightarrow \mathbb{R}$ , **find**  $F \in F$  of maximum value  $c(w(F))$ .

**Theorem:** Fix  $d$ . Then **convex combinatorial optimization (CCO)** is **strongly polynomially time solvable** over any **edge-guaranteed family**  $F$  given by a **membership oracle** and convex  $c$  given by an **evaluation oracle**.

## Back to the Applications

1. **Matroids** over  $n$  elements: **edge-well-behaved** with

$$E_n = \{ 1_i - 1_j : 1 \leq i < j \leq n \} \text{ so } m(n) = O(n^2)$$

## Back to the Applications

1. Matroids over  $n$  elements: edge-well-behaved with

$$E_n = \{ 1_i - 1_j : 1 \leq i < j \leq n \} \text{ so } m(n) = O(n^2)$$

2. PSD quadratic assignment over  $n \times n$  matrices: edge-well-behaved

$$\text{with } E_n = \{ 1_i : 1 \leq i \leq n \} \text{ so } m(n) = n$$

## Back to the Applications

1. **Matroids** over  $n$  elements: **edge-well-behaved** with  $E_n = \{1_i - 1_j : 1 \leq i < j \leq n\}$  so  $m(n) = O(n^2)$
2. **PSD quadratic assignment** over  $n \times n$  matrices: **edge-well-behaved** with  $E_n = \{1_i : 1 \leq i \leq n\}$  so  $m(n) = n$
3. **Partitioning**  $n$  items to  $p$  players: **edge-well-behaved** with  $E_n$  the set of all  $n \times p$  **circuit** matrices so  $m(n) = O(n^p)$
4. **Minimum variance clustering**: same as **partitioning**,  $m(n) = O(n^p)$

## Back to the Applications

1. **Matroids** over  $n$  elements: **edge-well-behaved** with  $E_n = \{1_i - 1_j : 1 \leq i < j \leq n\}$  so  $m(n) = O(n^2)$
2. **PSD quadratic assignment** over  $n \times n$  matrices: **edge-well-behaved** with  $E_n = \{1_i : 1 \leq i \leq n\}$  so  $m(n) = n$
3. **Partitioning**  $n$  items to  $p$  players: **edge-well-behaved** with  $E_n$  the set of all  $n \times p$  **circuit** matrices so  $m(n) = O(n^p)$
4. **Minimum variance clustering**: same as **partitioning**,  $m(n) = O(n^p)$

**Corollary:** All are **strongly polynomially time solvable**

Proof ingredient 1: membership  $\rightarrow$  augmentation  $\rightarrow$  linear optimization

Consider  $LCO$  with weight  $w$  over family  $F$  edge-guaranteed by  $E$

Proof ingredient 1: membership  $\rightarrow$  augmentation  $\rightarrow$  linear optimization

Consider  $LCO$  with weight  $w$  over family  $F$  edge-guaranteed by  $E$

Lemma 1: Membership  $\rightarrow$  Augmentation

Proof:  $F$  in  $F$  can be improved if and only if there is an edge direction  $e$  in  $E$  such that  $w \cdot e > 0$  and  $1_F + e = 1_G$  for some  $G$  in  $F$ .

## Proof ingredient 1: membership $\rightarrow$ augmentation $\rightarrow$ linear optimization

Consider  $LCO$  with weight  $w$  over family  $F$  edge-guaranteed by  $E$

Lemma 1: Membership  $\rightarrow$  Augmentation

Proof:  $F$  in  $F$  can be improved if and only if there is an edge direction  $e$  in  $E$  such that  $w \cdot e > 0$  and  $1_F + e = 1_G$  for some  $G$  in  $F$ .

Lemma 2: Augmentation  $\rightarrow$  linear Optimization

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

## Proof ingredient 1: membership $\rightarrow$ augmentation $\rightarrow$ linear optimization

Consider LCO with weight  $w$  over family  $F$  edge-guaranteed by  $E$

Lemma 1: Membership  $\rightarrow$  Augmentation

Proof:  $F$  in  $F$  can be improved if and only if there is an edge direction  $e$  in  $E$  such that  $w \cdot e > 0$  and  $1_F + e = 1_G$  for some  $G$  in  $F$ .

Lemma 2: Augmentation  $\rightarrow$  linear Optimization

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

Lemma 3: Polynomial time LCO  $\rightarrow$  Strongly polynomial time LCO

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

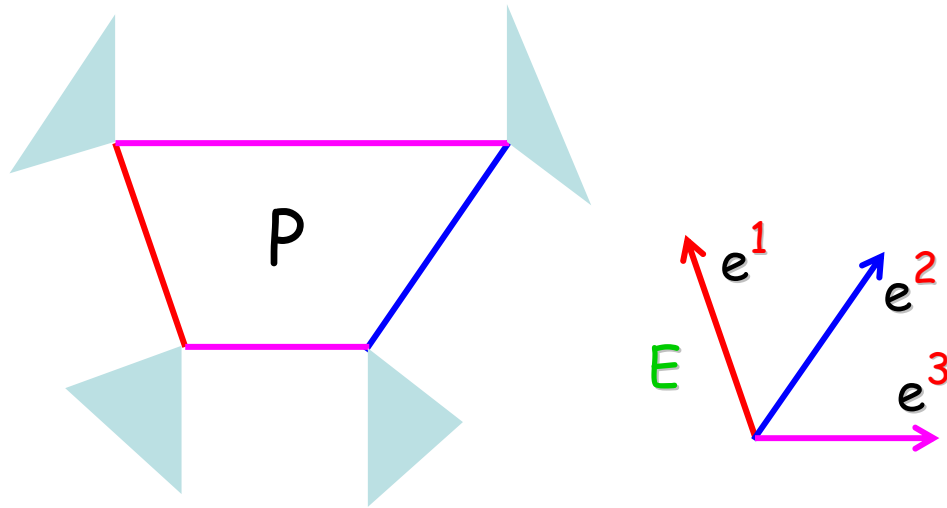
## Proof ingredient 2: zonotope refinement and construction

Lemma 4: If  $E = \{e^1, \dots, e^m\}$  contains all edge directions of a polytope  $P$  then the zonotope  $Z = [-1, 1]e^1 + \dots + [-1, 1]e^m$  is a refinement of  $P$ .

(Minkowsky, Grunbaum , ... , )

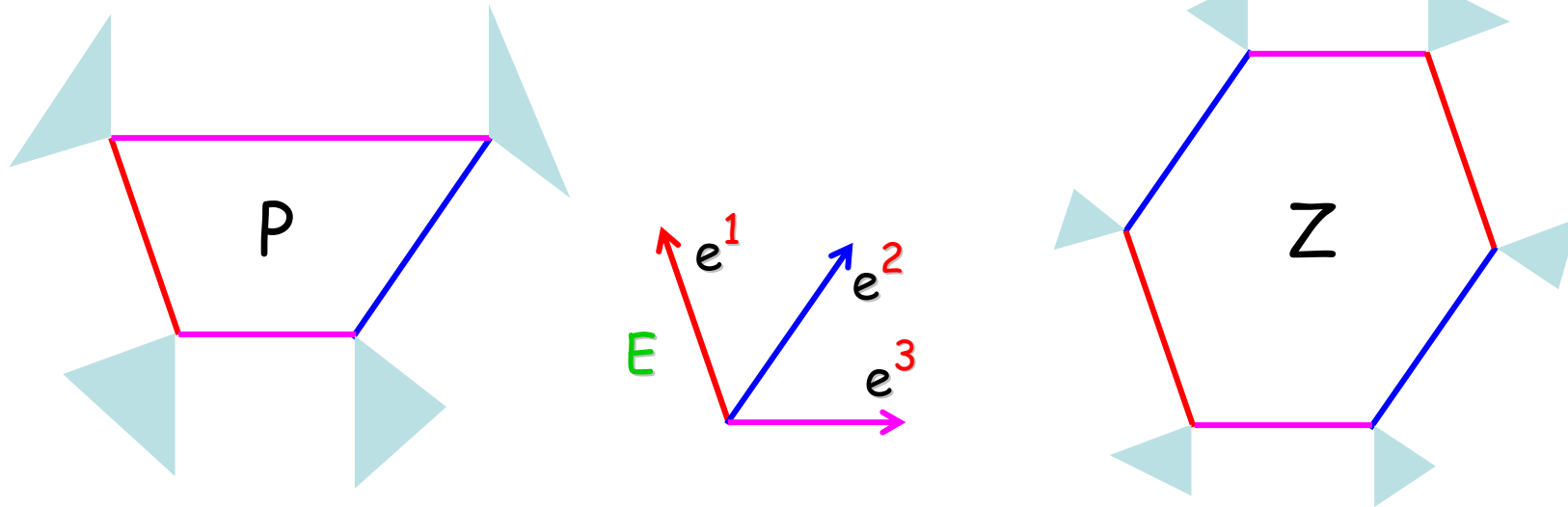
## Proof ingredient 2: zonotope refinement and construction

**Lemma 4:** If  $E = \{e^1, \dots, e^m\}$  contains all **edge directions** of a polytope  $P$  then the **zonotope**  $Z = [-1, 1]e^1 + \dots + [-1, 1]e^m$  is a **refinement** of  $P$ .



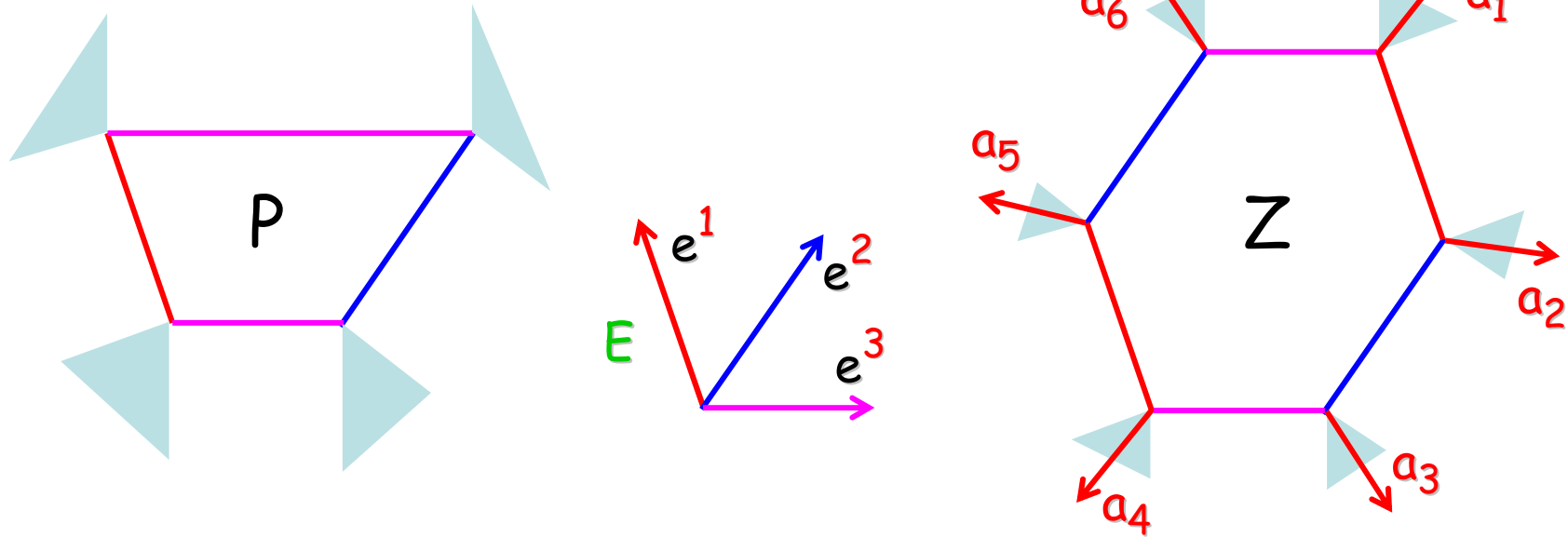
## Proof ingredient 2: zonotope refinement and construction

**Lemma 4:** If  $E = \{e^1, \dots, e^m\}$  contains all **edge directions** of a polytope  $P$  then the **zonotope**  $Z = [-1, 1]e^1 + \dots + [-1, 1]e^m$  is a **refinement** of  $P$ .



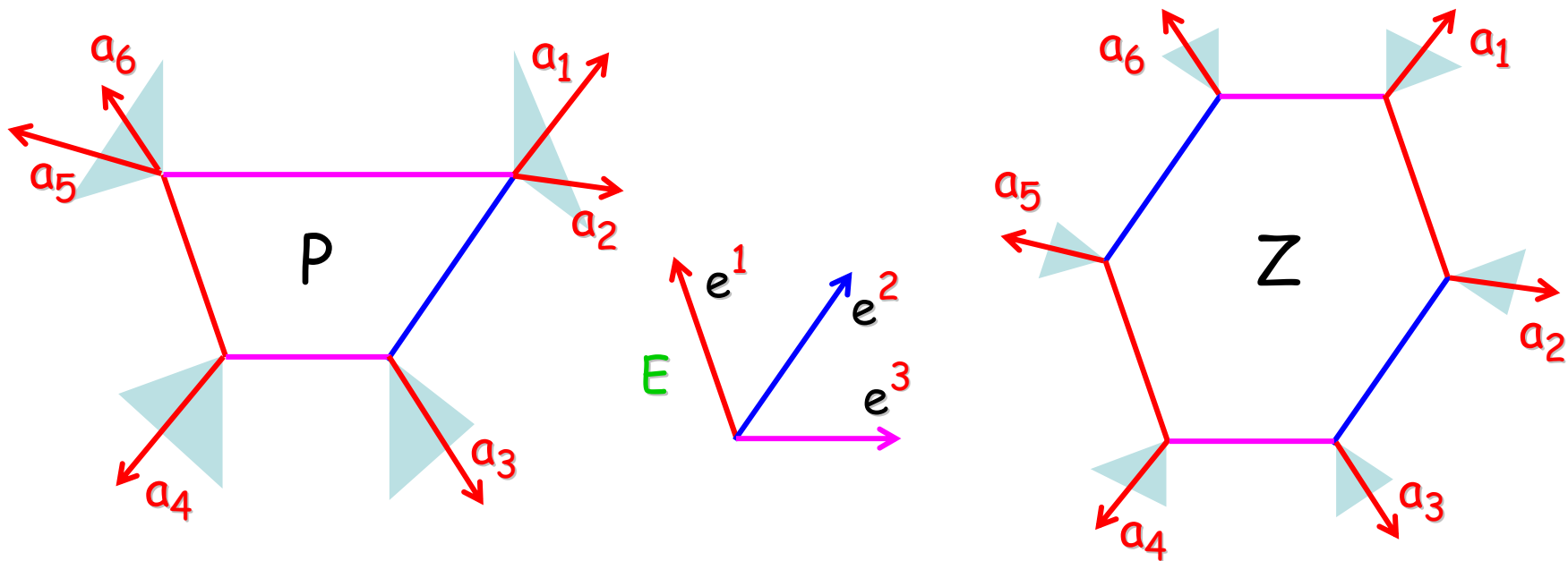
## Proof ingredient 2: zonotope refinement and construction

**Lemma 4:** If  $E = \{e^1, \dots, e^m\}$  contains all **edge directions** of a polytope  $P$  then the **zonotope**  $Z = [-1, 1]e^1 + \dots + [-1, 1]e^m$  is a **refinement** of  $P$ .



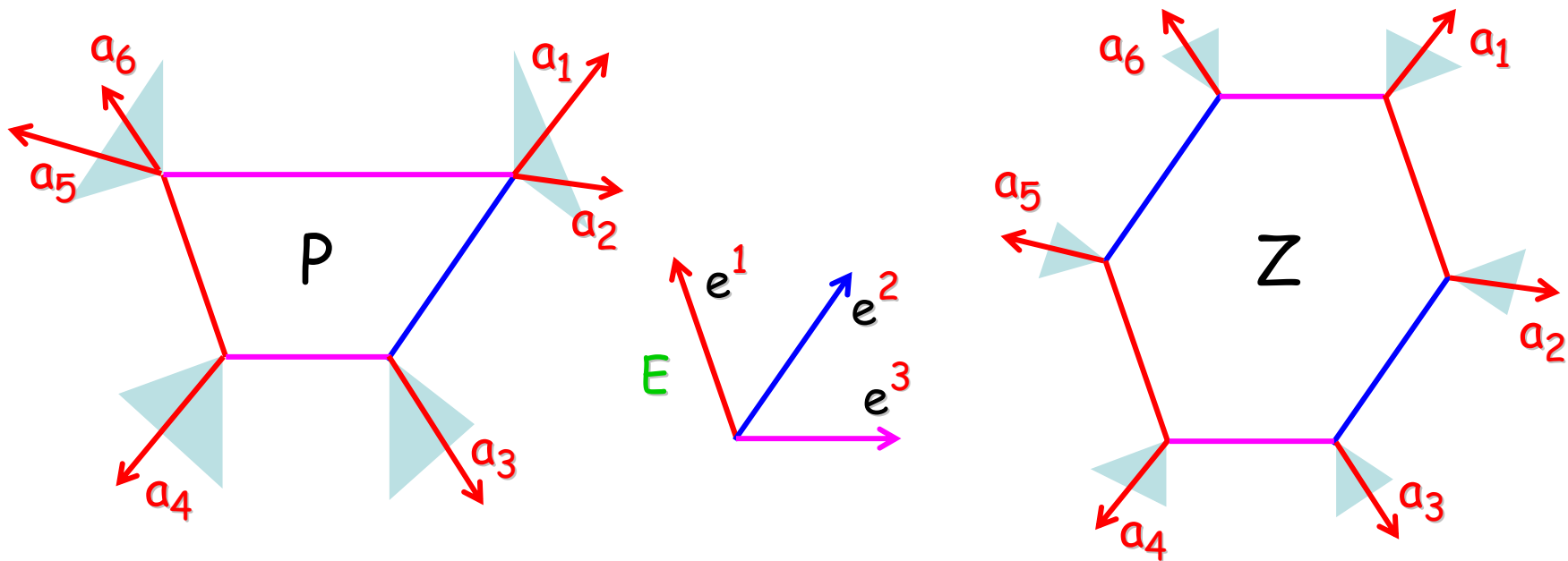
## Proof ingredient 2: zonotope refinement and construction

**Lemma 4:** If  $E = \{e^1, \dots, e^m\}$  contains all **edge directions** of a polytope  $P$  then the **zonotope**  $Z = [-1, 1]e^1 + \dots + [-1, 1]e^m$  is a **refinement** of  $P$ .



## Proof ingredient 2: zonotope refinement and construction

**Lemma 4:** If  $E = \{e^1, \dots, e^m\}$  contains all **edge directions** of a polytope  $P$  then the **zonotope**  $Z = [-1, 1]e^1 + \dots + [-1, 1]e^m$  is a **refinement** of  $P$ .



**Lemma 5:** In  $\mathbb{R}^d$ , the zonotope  $Z$  can be constructed from  $E = \{e^1, \dots, e^m\}$  along with a vector  $a_i$  in the **cone** of **every vertex** in  $O(m^{d-1})$  operations.

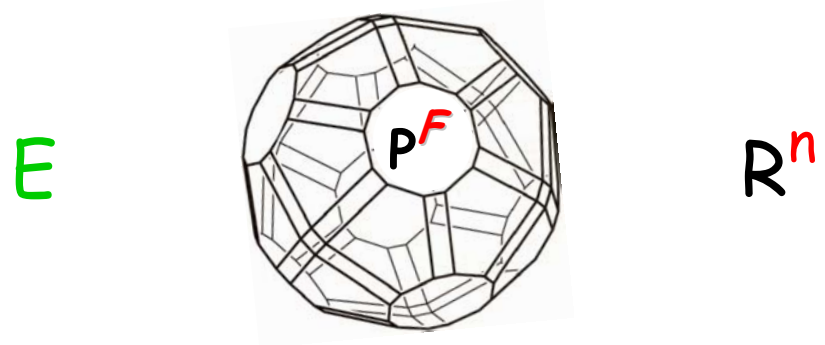
(Edelsbrunner, Gritzmann, Orourke, Seidel, Sharir, Sturmfels, ...)

## Proof - combining all ingredients: the CCO algorithm

Input: Family  $\mathcal{F}$  given by membership oracle, edge-guaranteed by  $E$  in  $\mathbb{R}^n$ ,  
dxn weight matrix  $w$ , convex functional  $c$  on  $\mathbb{R}^d$  given by evaluation oracle

## Proof - combining all ingredients: the CCO algorithm

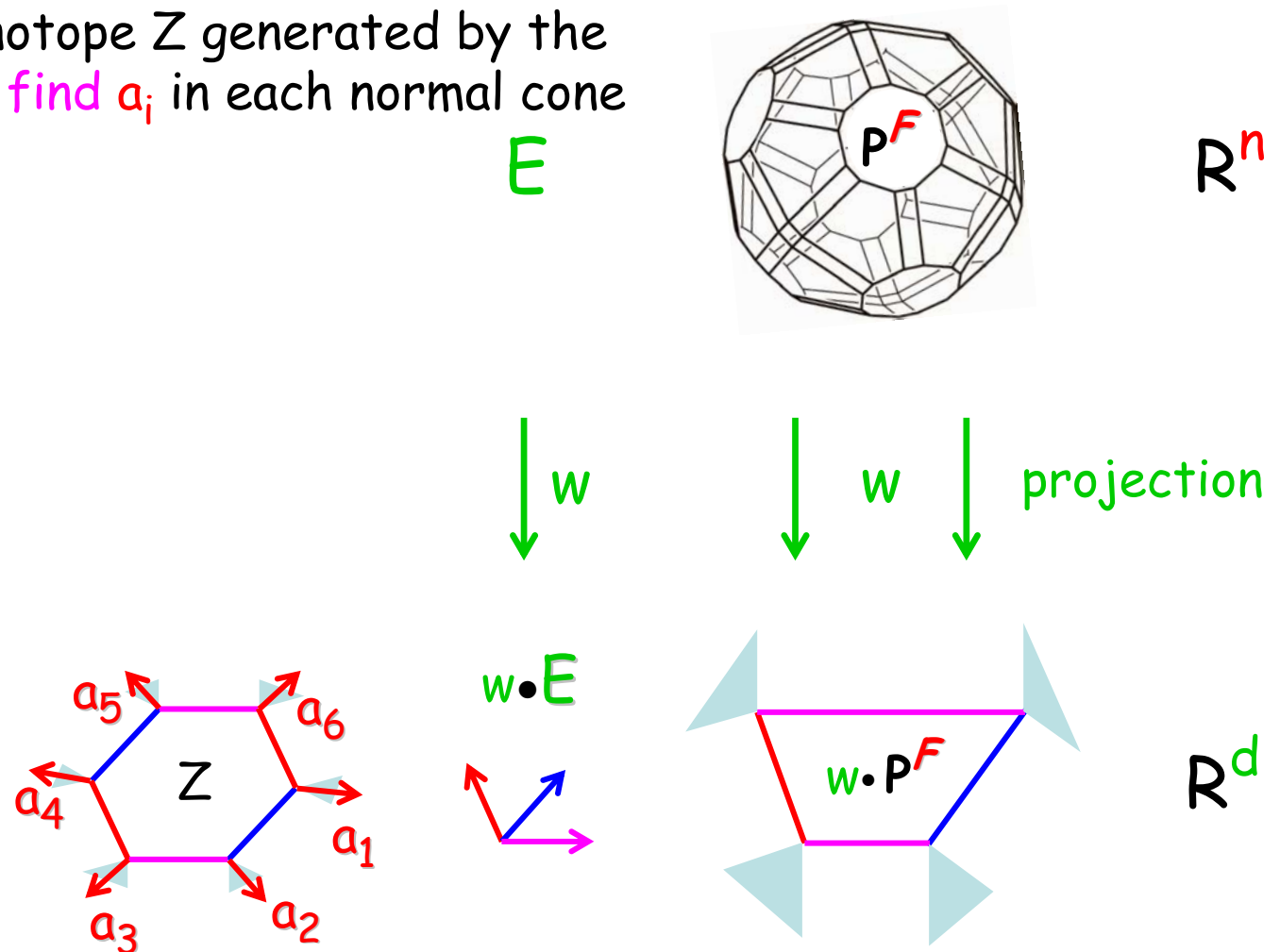
Input: Family  $F$  given by membership oracle, edge-guaranteed by  $E$  in  $\mathbb{R}^n$ ,  $d \times n$  weight matrix  $w$ , convex functional  $c$  on  $\mathbb{R}^d$  given by evaluation oracle



# Proof - combining all ingredients: the CCO algorithm

**Input:** Family  $F$  given by membership oracle, edge-guaranteed by  $E$  in  $\mathbb{R}^n$ ,  $d \times n$  weight matrix  $w$ , convex functional  $c$  on  $\mathbb{R}^d$  given by evaluation oracle

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

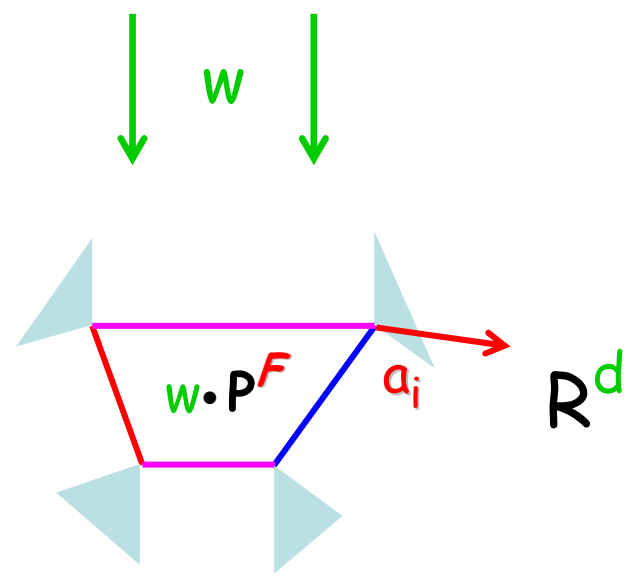
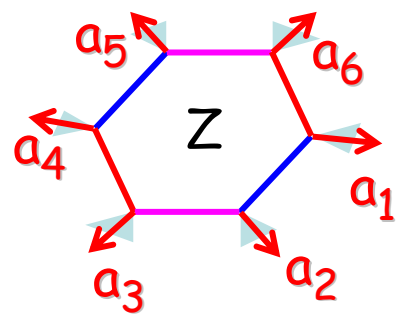
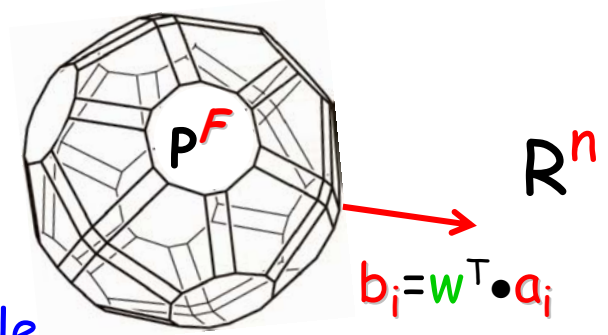


# Proof - combining all ingredients: the CCO algorithm

**Input:** Family  $F$  given by membership oracle, edge-guaranteed by  $E$  in  $\mathbb{R}^n$ ,  $d \times n$  weight matrix  $w$ , convex functional  $c$  on  $\mathbb{R}^d$  given by evaluation 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  $\mathbb{R}^d$  to  $b_i = w^T \bullet a_i$  in  $\mathbb{R}^n$  and solve LCO with weight  $b_i$  over  $F$  using membership oracle



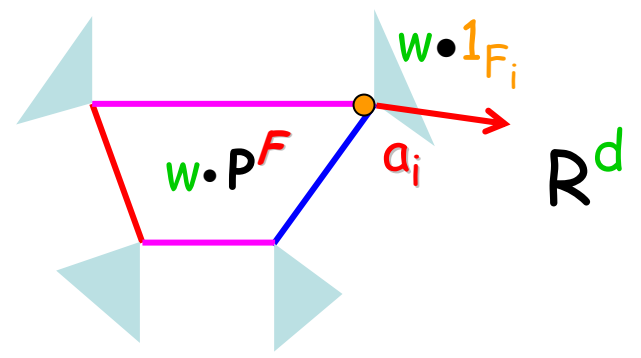
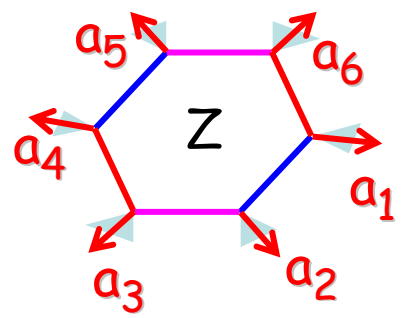
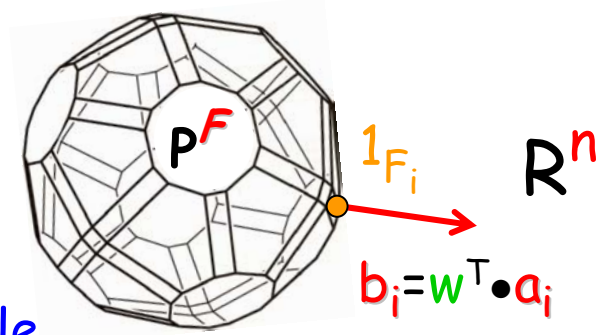
# Proof - combining all ingredients: the CCO algorithm

**Input:** Family  $F$  given by membership oracle, edge-guaranteed by  $E$  in  $\mathbb{R}^n$ ,  $d \times n$  weight matrix  $w$ , convex functional  $c$  on  $\mathbb{R}^d$  given by evaluation 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  $\mathbb{R}^d$  to  $b_i = w^T \bullet a_i$  in  $\mathbb{R}^n$  and solve LCO with weight  $b_i$  over  $F$  using membership oracle

3. Obtain the vertex  $1_{F_i}$  of  $P^F$  and the vertex  $w \bullet 1_{F_i}$  of  $w \bullet P^F$



# Proof - combining all ingredients: the CCO algorithm

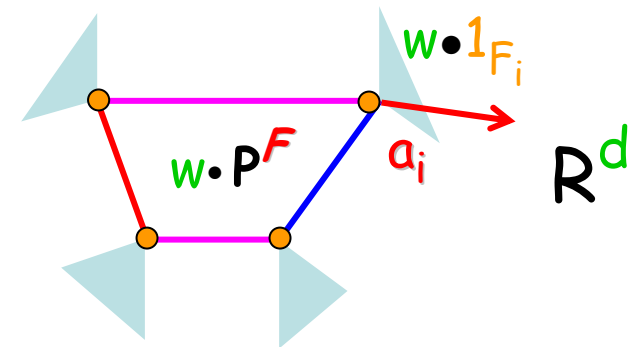
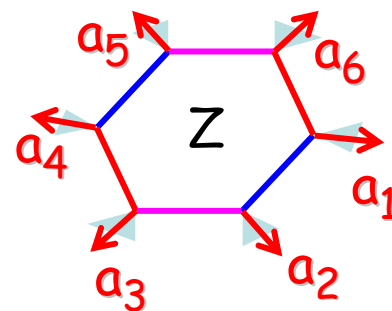
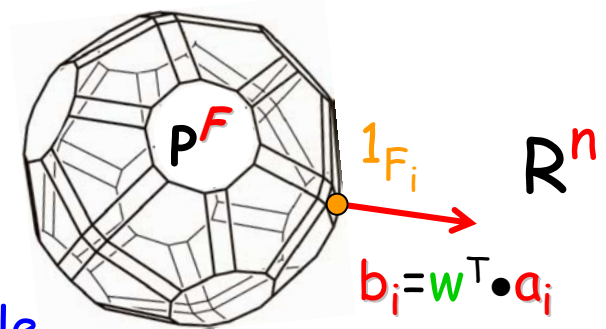
**Input:** Family  $F$  given by membership oracle, edge-guaranteed by  $E$  in  $\mathbb{R}^n$ ,  $d \times n$  weight matrix  $w$ , convex functional  $c$  on  $\mathbb{R}^d$  given by evaluation oracle

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  $\mathbb{R}^d$  to  $b_i = w^T \bullet a_i$  in  $\mathbb{R}^n$  and solve LCO with weight  $b_i$  over  $F$  using membership oracle

3. Obtain the vertex  $1_{F_i}$  of  $P^F$  and the vertex  $w \bullet 1_{F_i}$  of  $w \bullet P^F$

4. Output an  $F_i$  attaining maximum value  $c(w \bullet 1_{F_i})$  using evaluation oracle



## Part 2. Convex Integer Programming

Lecture 3. The Problem; Applications; The Theorems.

# The Convex Integer Optimization Problem

We consider the following convex integer programming problem:

$$\max \{c(w_1x, \dots, w_dx) : x \geq 0, Ax = b, x \text{ integer}\}$$

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

# The Convex Integer Optimization Problem

We consider the following **convex integer programming problem**:

$$\max \{c(w_1x, \dots, w_dx) : x \geq 0, Ax = b, x \text{ integer}\}$$

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

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

# The Convex Integer Optimization Problem

We consider the following **convex integer programming problem**:

$$\max \{c(w_1x, \dots, w_dx) : x \geq 0, Ax = b, x \text{ integer}\}$$

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

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  $\mathbb{R}$ .

# The Convex Integer Optimization Problem

We consider the following **convex integer programming** problem:

$$\max \{c(w_1x, \dots, w_dx) : x \geq 0, Ax = b, x \text{ integer}\}$$

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

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  $\mathbb{R}$ .

The **CCO problem** of before can also be interpreted that way with  $w_i$  the  $i$ -th coordinate of the given **vectorial weighting**.

# Edge-Directions Revisited

## Edge-Directions Revisited

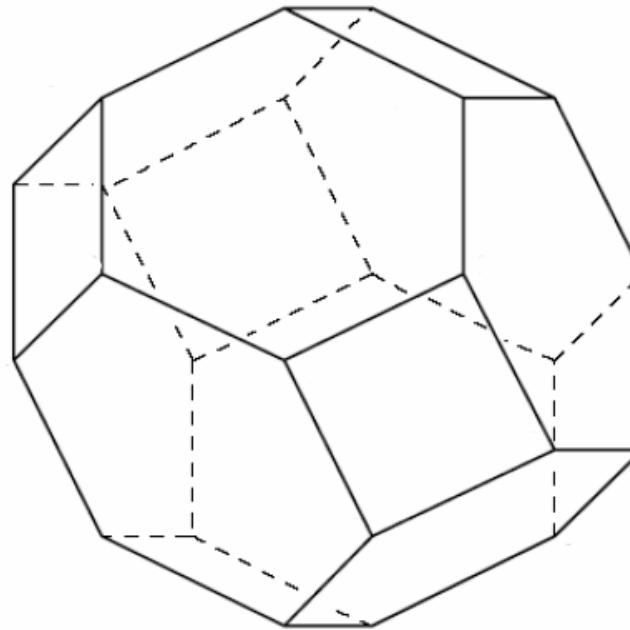
Exploit edge symmetry of the integer hull (in analogy with  $P^F$  in the CCO)

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

## Edge-Directions Revisited

Exploit edge symmetry of the integer hull (in analogy with  $P^F$  in the CCO)

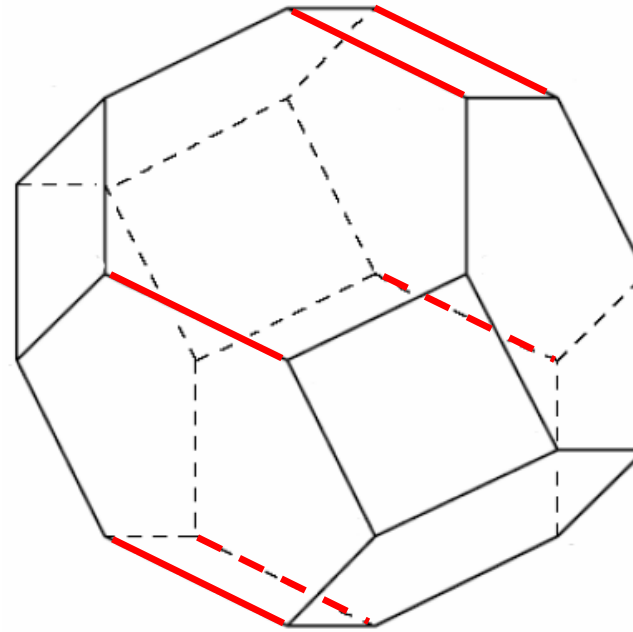
$$P = \text{conv}\{x : x \geq 0, Ax = b, x \text{ integer}\} \subseteq \mathbb{R}^n$$



# Edge-Directions Revisited

Exploit edge symmetry of the integer hull (in analogy with  $P^F$  in the CCO)

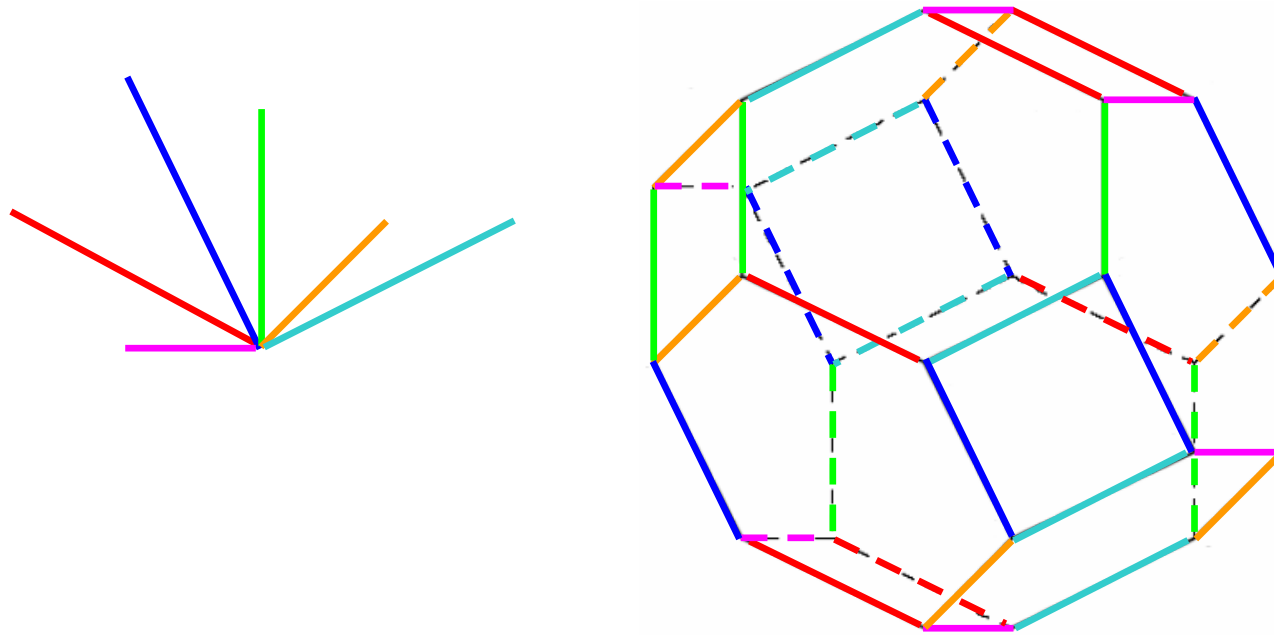
$$P = \text{conv}\{x : x \geq 0, Ax = b, x \text{ integer}\} \subseteq \mathbb{R}^n$$



# Edge-Directions Revisited

Exploit edge symmetry of the integer hull (in analogy with  $P^F$  in the CCO)

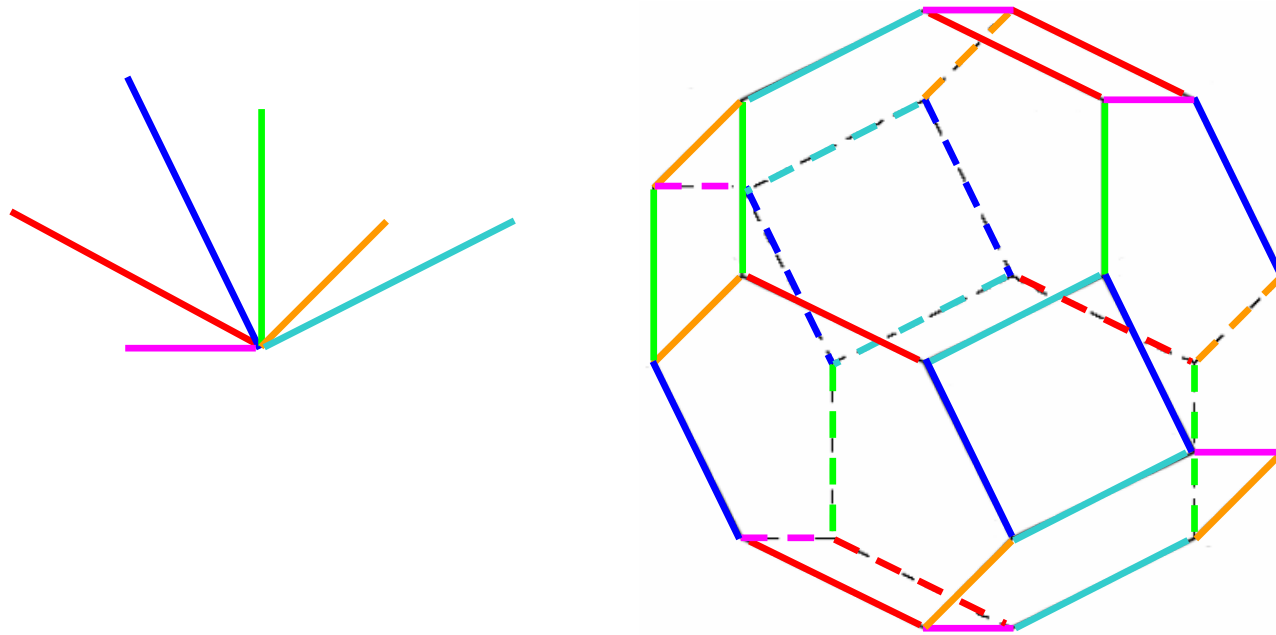
$$P = \text{conv}\{x : x \geq 0, Ax = b, x \text{ integer}\} \subseteq \mathbb{R}^n$$



# Edge-Directions Revisited

Exploit edge symmetry of the integer hull (in analogy with  $P^F$  in the CCO)

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



**Theorem:** 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  $P$ .

# Totally Unimodular Systems ?

## Totally Unimodular Systems ?

By our **theorem**, to solve the **convex integer programming problem**

$$\max \{c(w_1x, \dots, w_dx) : x \geq 0, Ax = b, x \text{ integer}\},$$

# Totally Unimodular Systems ?

By our **theorem**, to solve the **convex integer programming problem**

$$\max \{c(w_1x, \dots, w_dx) : x \geq 0, Ax = b, x \text{ integer}\},$$

**need:**

1. Solve linear integer programming counterparts
2. Get hold of the set of edge-directions of the integer hull

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

## Totally Unimodular Systems ?

By our **theorem**, to solve the **convex integer programming problem**

$$\max \{c(w_1x, \dots, w_dx) : x \geq 0, Ax = b, x \text{ integer}\},$$

**need:**

1. Solve linear integer programming counterparts
2. Get hold of the set of edge-directions of the integer hull

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

If **A** is totally unimodular then:

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

# Totally Unimodular Systems ?

By our **theorem**, to solve the **convex integer programming problem**

$$\max \{c(w_1x, \dots, w_dx) : x \geq 0, Ax = b, x \text{ integer}\},$$

**need:**

1. Solve linear integer programming counterparts
2. Get hold of the set of edge-directions of the integer hull

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

**If A is totally unimodular then:**

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

**So:**

1. Can solve linear integer programming by LP over P
2. Edge-directions of P are circuits of A, have at most  $\binom{n}{m+1}$

# Totally Unimodular Systems ?

By our **theorem**, to solve the **convex integer programming problem**

$$\max \{c(w_1x, \dots, w_dx) : x \geq 0, Ax = b, x \text{ integer}\},$$

**need:**

1. Solve linear integer programming counterparts
2. Get hold of the set of edge-directions of the integer hull

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

If **A** is totally unimodular then:

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

**So:**

1. Can solve linear integer programming by LP over  $P$
2. Edge-directions of  $P$  are circuits of  $A$ , have at most  $\binom{n}{m+1}$

Quite useful, but not polynomial time solvable

# N-Fold Systems

# 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,

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

# 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,

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

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,

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

We establish the following theorem.

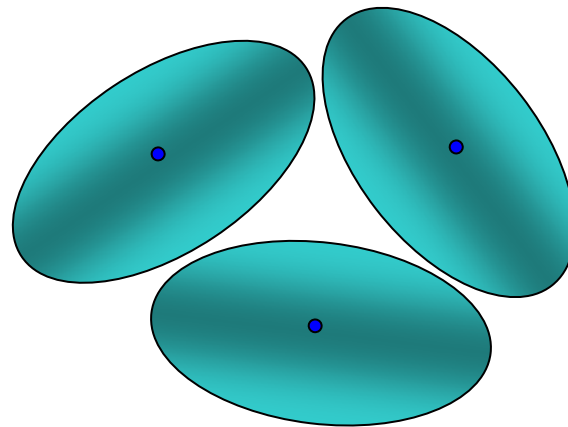
**Theorem:** For any fixed  $d$  and  $(r+s) \times t$  matrix  $A$ , there is a **polynomial oracle-time** algorithm that, given  $n, b, w_1, \dots, w_d$ , and convex  $c$  presented by **comparison oracle**, solves the **(convex) integer programming problem**

$$\max \{ c(w_1x, \dots, w_dx) : A^{(n)}x = b, x \in \mathbb{N}^{nt} \}$$



## Applications 3,4 revisited: Partitioning and Clustering

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



$P=3$ ,  $k=3$ ,  $m$  large

## Applications 3,4 revisited: Partitioning and Clustering

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  $A_2 = (1, \dots, 1)$ . So we deduce:

## Applications 3,4 revisited: Partitioning and Clustering

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  $A_2 = (1, \dots, 1)$ . **So we deduce:**

**Corollary:** Partitioning and Clustering with fixed  $p, k$  are **polytime solvable**.

# Application 5: Bin Packing

## Application 5: 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 5: 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 5: 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  $A_2 = (v_1, \dots, v_t)$ . **So we deduce:**

## Application 5: 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  $A_2 = (v_1, \dots, v_t)$ . **So we deduce:**

**Corollary:** Fix  $d, t, v_1, \dots, v_t$ . Then **convex bin packing** is **polytime solvable**.

## Application 6: Multiway Transportation

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.

## Application 6: Multiway Transportation

A **k-way table** is an  $m_1 \times \dots \times m_k$  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  $2 \times 3$ :

0	1	2
2	2	0

## Application 6: Multiway Transportation

A **k-way table** is an  $m_1 \times \dots \times m_k$  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  $2 \times 3$  with **line-sums**:

0	1	2	3
2	2	0	

## Application 6: Multiway Transportation

A **k-way table** is an  $m_1 \times \dots \times m_k$  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  $2 \times 3$  with **line-sums**:

0	1	2	3
2	2	0	2

## Application 6: Multiway Transportation

A **k-way table** is an  $m_1 \times \dots \times m_k$  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  $2 \times 3$  with **line-sums**:

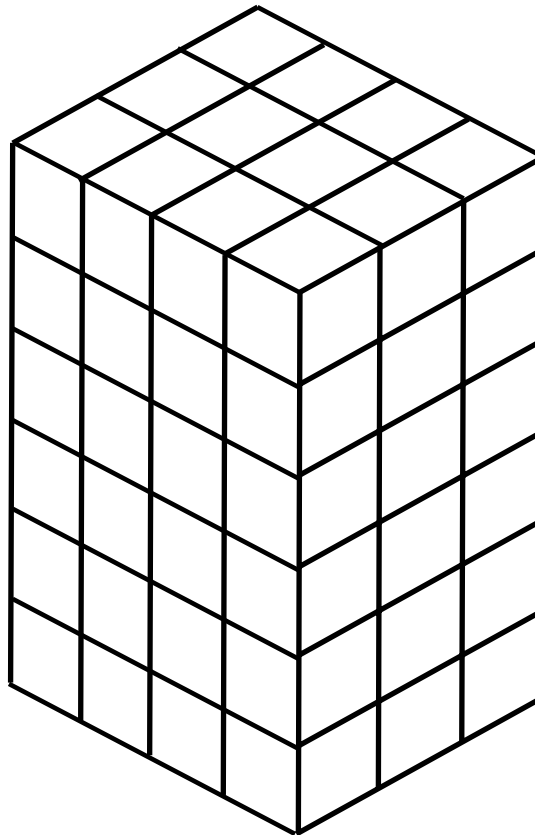
0	1	2	3
2	2	0	4
2	3	2	

## Application 6: Multiway Transportation

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  $3 \times 4 \times 6$ :

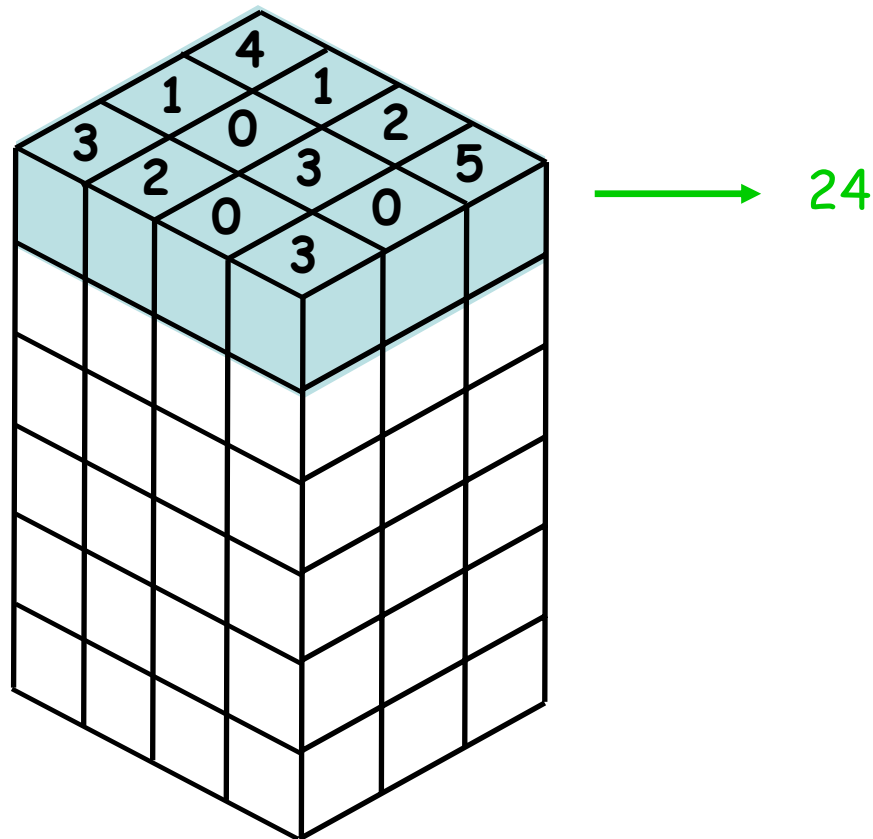


## Application 6: Multiway Transportation

A **k-way table** is an  $m_1 \times \dots \times m_k$  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**:

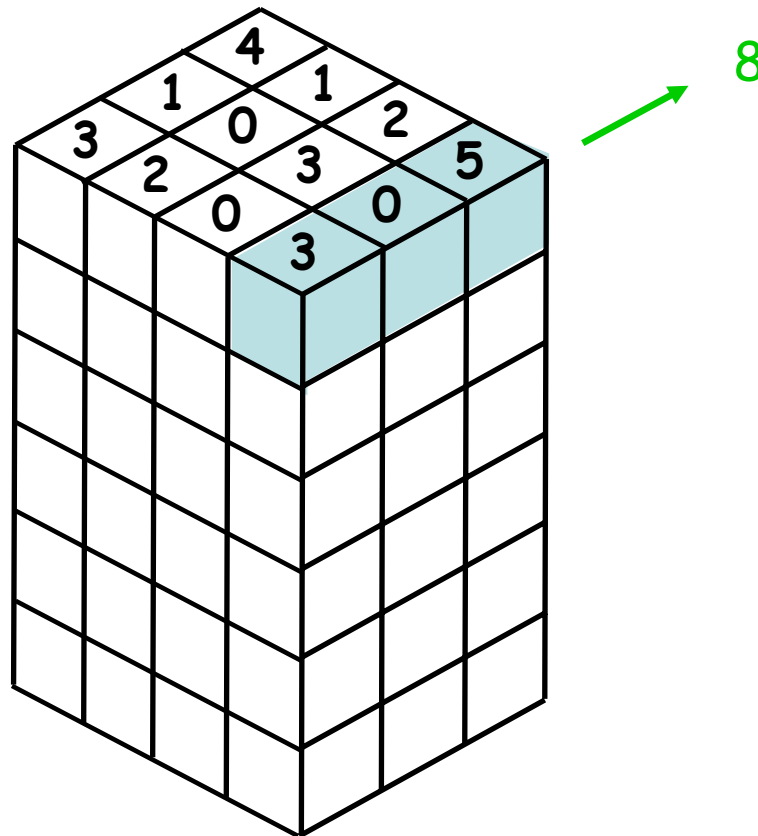


## Application 6: Multiway Transportation

A **k-way table** is an  $m_1 \times \dots \times m_k$  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**:



## Application 6: Multiway Transportation

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

## Application 6: Multiway Transportation

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**.

## Application 6: Multiway Transportation

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**.

The **multiway transportation problem** is the **linear integer programming problem** over the **multiway polytope**.

# Application 6: Multiway Transportation

More formally, a  $k$ -way polytope is the set of all  $m_1 \times \cdots \times m_k$  nonnegative arrays  $x = (x_{i_1, \dots, i_k})$  such that the sums of the entries over some of their lower dimensional sub-arrays (margins) are specified. More precisely, for any tuple  $(i_1, \dots, i_k)$  with  $i_j \in \{1, \dots, m_j\} \cup \{+\}$ , the corresponding *margin*  $x_{i_1, \dots, i_k}$  is the sum of entries of  $x$  over all coordinates  $j$  with  $i_j = +$ . The *support* of  $(i_1, \dots, i_k)$  and of  $x_{i_1, \dots, i_k}$  is the set  $\text{supp}(i_1, \dots, i_k) := \{j : i_j \neq +\}$  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  $\{1, \dots, k\}$ , it consists of all margins  $u_{i_1, \dots, i_k}$  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  $\{1, \dots, k\}$ . Given a hierarchical collection of margins  $u_{i_1, \dots, i_k}$  supported on a family  $\mathcal{F}$  of subsets of  $\{1, \dots, k\}$ , the corresponding  $k$ -way polytope is the set of nonnegative arrays with these margins,

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

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

## Application 6: Multiway Transportation

The **margin equations** for any  $m_1 \times \dots \times m_k \times n$  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 6: Multiway Transportation

The **margin equations** for any  $m_1 \times \dots \times m_k \times n$  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  $i = 1, \dots, n$ , the  $(9+6) \times 9$  matrix  $A$  whose n-fold product  $A^{(n)}$  defines the  $3 \times 3 \times n$  multiway polytope has  $A_1 = I_9$ ,

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

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.

# Application 6: Multiway Transportation

We deduce the **optimization theorem** for long  $k$ -way transportation:

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

# Application 6: Multiway Transportation

We deduce the **optimization theorem** for long  $k$ -way transportation:

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

This is in contrast with our following **universality theorem** for **short 3-way polytopes** (to be discussed in last lecture):

**Theorem:** Any rational polytope is an  $r \times c \times 3$  **line-sum 3-way polytope**, and any linear/integer program is an  $r \times c \times 3$  **line-sum 3-way program**, and hence the corresponding transportation problem is **NP-hard**.

## Part 2. Convex Integer Programming

### Lecture 4. Proofs and Algorithms.

## Recall: 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,

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

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  $n, b, w_1, \dots, w_d$ , and convex  $c$  presented by **comparison oracle**, solves the **(convex) integer programming problem**

$$\max \{ c(w_1x, \dots, w_dx) : A^{(n)}x = b, x \in \mathbb{N}^{nt} \}$$

# Proof: Graver Bases

## Proof: 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: 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: 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 1:** 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: 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 1:** 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 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: 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 1:** 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 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**  
(Schultz-Weismantel)

## Proof: 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 1:** 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 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**  
(Schultz-Weismantel)

**Lemma 3:** The Graver basis of the product  $A^{(n)}$  is **polytime computable**.

## Proof: 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 1:** 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 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**.  
(Schultz-Weismantel)

**Lemma 3:** The Graver basis of the product  $A^{(n)}$  is **polytime computable**.

**Proof:** use Graver basis **stabilization**

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

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

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

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  $\phi_{k_1, k_2} : \mathbb{Z}^{2 \cdot 2} \rightarrow \mathbb{Z}^{4 \cdot 2}$  for  $1 \leq k_1 < k_2 \leq 4$ , and we obtain

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

Combining **Edge-Behavior Theorem**, **Lemmas 1 - 3**, and some **more components**, we obtain the **Theorem on N-Fold Systems**:

Combining **Edge-Behavior Theorem**, **Lemmas 1 - 3**, and some **more components**, we obtain the **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  $n$ ,  $b$ ,  $w_1, \dots, w_d$ , and convex  $c$  presented by **comparison oracle**, solves the **convex integer programming problem**

$$\max \{ c(w_1x, \dots, w_dx) : A^{(n)}x = b, x \in \mathbb{N}^{nt} \}$$

## Lecture 5: Universality of Multiway Transportation.

# Recall:

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**.

# Recall:

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:

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

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

# 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  $r \times c \times 3$  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  $r \times c \times 3$  multiway program.
- Optimization over  $r \times c \times 3$  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  $r \times c \times 3$  multiway program.
- Optimization over  $r \times c \times 3$  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  $r \times c \times 3$  multiway program.
- Optimization over  $r \times c \times 3$  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 10<sup>th</sup> problem on the decidability of the realization problem for polytopes?

**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.

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

## 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 ?

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	

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** 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**.

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** × **c** × **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  $S = \{s_1, \dots, s_m\}$ , let

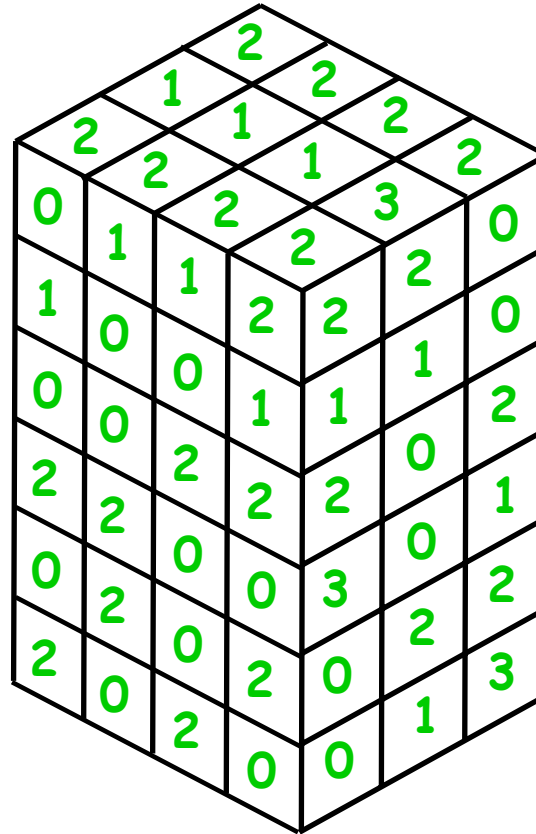
$$P := \{y \in \mathbb{R}_+^{m+1} : y_0 - \sum_{i=1}^m s_i y_i = 0, \sum_{i=1}^m y_i = 1\}.$$

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

Example: set of entry values with a gap

Example: set of entry values with a gap

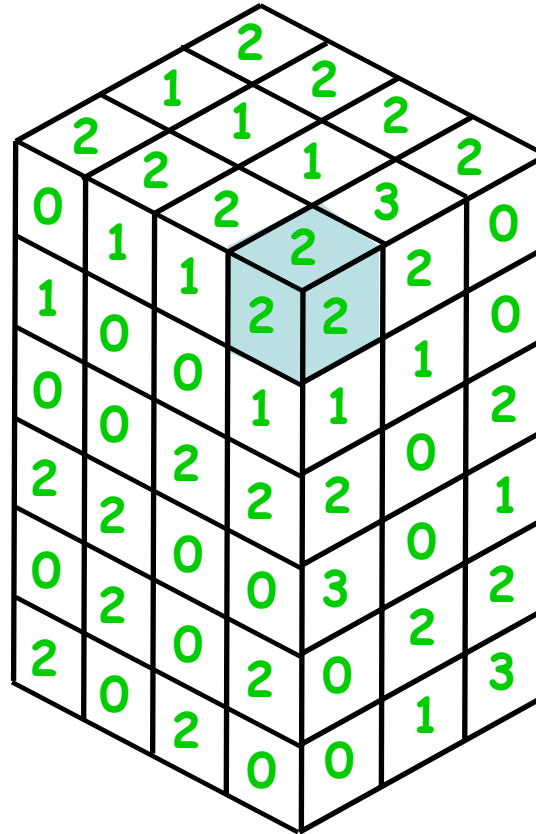
Consider the following line-sums for  $6 \times 4 \times 3$  tables:



# Example: set of entry values with a gap

Consider the following line-sums for  $6 \times 4 \times 3$  tables:

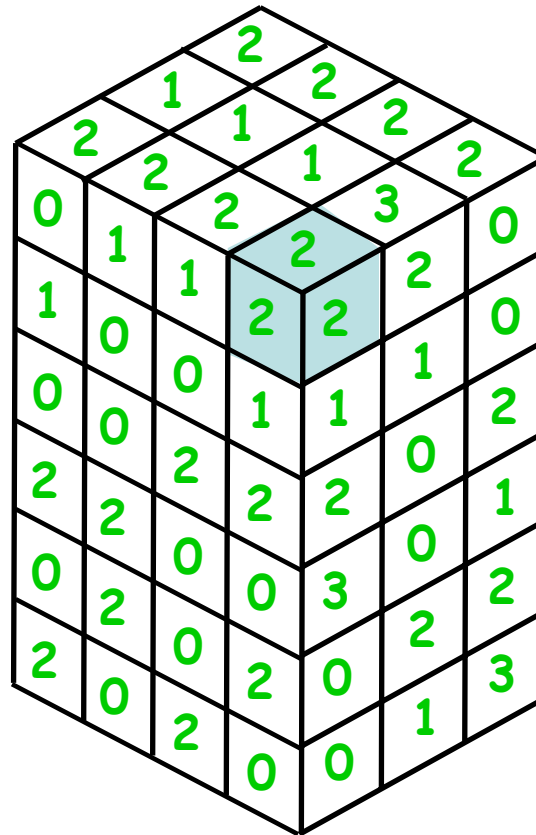
Consider the designated entry:



## Example: set of entry values with a gap

Consider the following **line-sums** for  $6 \times 4 \times 3$  tables:

Consider the designated entry:



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

# More Universality Consequences

# 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 cornerstone 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  $r \times c \times 3$  tables with fixed line-sums.

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

## 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

$$B = \left\{ (x^1, x^2) \in \oplus_2 \mathbb{R}_+^{n \times n} : \sum_j x_{i,j}^k = r_i^k, \sum_i x_{i,j}^k = c_j^k, x_{i,j}^1 + x_{i,j}^2 \leq u_{i,j} \right\}$$

**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.

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

$$(d_j^1) = (1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0), \quad (d_j^2) = (0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 2 \ 1).$$

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



# Conclusion

Convex Discrete Optimization is broadly expressive and useful framework.

# Conclusion

Convex Discrete Optimization is broadly expressive and useful framework.

- Convex Combinatorial Optimization is strongly polytime solvable over any edge-guaranteed family  $\mathcal{F}$ .

# Conclusion

Convex Discrete Optimization is broadly expressive and useful framework.

- Convex Combinatorial Optimization is strongly polytime solvable over any edge-guaranteed family  $\mathcal{F}$ .
- Convex Integer Programming is polytime solvable over any  $n$ -fold system  $A^{(n)}$ .

# Conclusion

Convex Discrete Optimization is broadly expressive and useful framework.

- Convex Combinatorial Optimization is strongly polytime solvable over any edge-guaranteed family  $\mathcal{F}$ .
- Convex Integer Programming is polytime solvable over any  $n$ -fold system  $A^{(n)}$ .

## Some Broad Research Directions

# Conclusion

Convex Discrete Optimization is broadly expressive and useful framework.

- Convex Combinatorial Optimization is strongly polytime solvable over any edge-guaranteed family  $\mathcal{F}$ .
- Convex Integer Programming is polytime solvable over any  $n$ -fold system  $A^{(n)}$ .

## Some Broad Research Directions

Approximation algorithms for classes of families and integer programs.

# Conclusion

Convex Discrete Optimization is broadly expressive and useful framework.

- Convex Combinatorial Optimization is strongly polytime solvable over any edge-guaranteed family  $\mathcal{F}$ .
- Convex Integer Programming is polytime solvable over any  $n$ -fold system  $A^{(n)}$ .

## Some Broad Research Directions

Approximation algorithms for classes of families and integer programs.

Classification of edge-well-behaved classes of families.

# Conclusion

Convex Discrete Optimization is broadly expressive and useful framework.

- Convex Combinatorial Optimization is strongly polytime solvable over any edge-guaranteed family  $\mathcal{F}$ .
- Convex Integer Programming is polytime solvable over any  $n$ -fold system  $A^{(n)}$ .

## Some Broad Research Directions

Approximation algorithms for classes of families and integer programs.

Classification of edge-well-behaved classes of families.

Convex minimization: our results enable polynomial time construction of inequality presentations of the projections  $w \bullet P^{\mathcal{F}}$ ,  $w \bullet P$  providing first step.

**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)
- Convex combinatorial optimization (Disc. Comp. Geom. 2004)

## 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)
- Convex combinatorial optimization (Disc. Comp. Geom. 2004)

### Also related:

- All linear and integer programs are  
slim 3-way transportation programs (SIAM J. Opt., to appear)
- The complexity of 3-way statistical tables (SIAM J. Comp. 2004)
- The Hilbert zonotope and a polynomial time algorithm  
for universal Gröbner bases (Adv. App. Math. 2003)