Advances in Integer Programming

Shmuel Onn

Technion – Israel Institute of Technology

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

(Non)-Linear Integer Programming

The problem is: $\min/\max \{ f(x) \ : \ Ax \le b, \ \ l \le x \le u, \ \ x \text{ in } Z^n \}$

with data: A: integer $m \times n$ matrix

b: right-hand side in Z^m

l,u: lower/upper bounds in Z^n

f: function from Z^n to R

(Non)-Linear Integer Programming

The problem is: $\min/\max \{ f(x) \ : \ Ax \le b, \ \ l \le x \le u, \ \ x \text{ in } Z^n \}$

Has generic modeling power but NP-hard even for linear $f(x)=wx$

(Non)-Linear Integer Programming

The problem is: $\min/\max \{ f(x) \ : \ Ax \le b, \ \ l \le x \le u, \ \ x \text{ in } Z^n \}$

Has generic modeling power but NP-hard even for linear $f(x)=wx$

Classical theory enable efficient solution in fixed dimension

(Non)-Linear Integer Programming

The problem is: $\min/\max \{ f(x) \ : \ Ax \le b, \ \ l \le x \le u, \ \ x \text{ in } Z^n \}$

Has generic modeling power but NP-hard even for linear $f(x)=wx$

Classical theory enable efficient solution in fixed dimension

New theory enables the efficient solution of broad natural universal (non)-linear integer programs in variable dimension

(Non)-Linear Integer Programming

The problem is: $\min/\max \{ f(x) \ : \ Ax \le b, \ \ l \le x \le u, \ \ x \text{ in } Z^n \}$

Has generic modeling power but NP-hard even for linear $f(x)=wx$

Classical theory enable efficient solution in fixed dimension

New theory enables the efficient solution of broad natural universal (non)-linear integer programs in variable dimension

Recently there have been several applications of this theory to parameterized complexity and approximation algorithms

Outline: Lecture 1

Graver Bases and Integer Programming

Shmuel Onn

Outline: Lecture 1

Graver Bases and Integer Programming

N-Fold Integer Programming

Outline: Lecture 1

Graver Bases and Integer Programming

N-Fold Integer Programming

Generic Example: Multiway Tables

Outline: Lecture 2

N-Fold IP is Fixed-Parameter Tractable

Outline: Lecture 2

N-Fold IP is Fixed-Parameter Tractable

Huge Multicommodity Flows

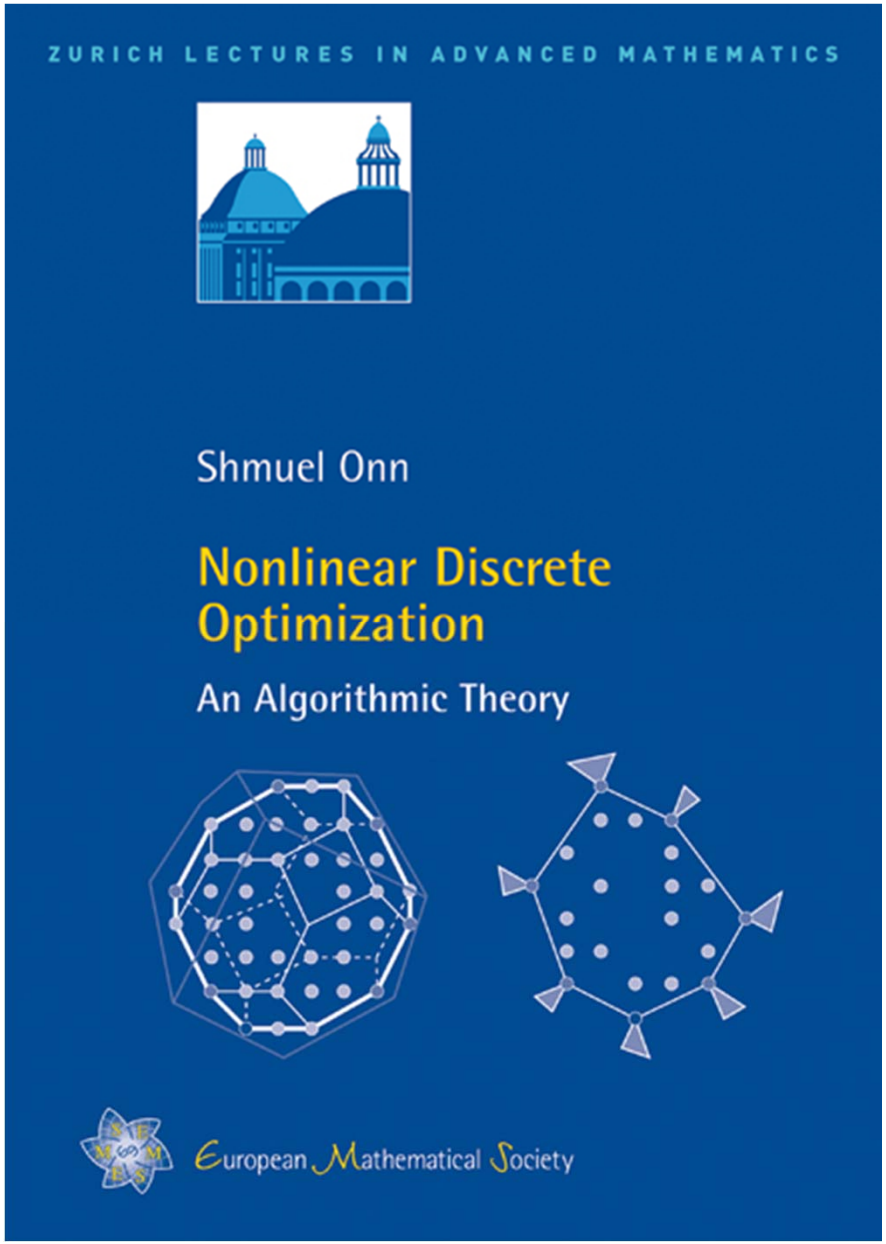
Outline: Lecture 2

N-Fold IP is Fixed-Parameter Tractable

Huge Multicommodity Flows

More Applications, Extensions, Directions

Lecture 1



Background in my Book:

Theory of Graver bases for integer programming

(and more)

Available electronically from my homepage

(with kind permission of EMS)

Graver Bases

and

Nonlinear Integer Programming

Graver Bases

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

Graver Bases

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

x is conformal to y if $x_i y_i \geq 0$ (same orthant) and $|x_i| \leq |y_i|$ for all i

Graver Bases

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

x is conformal to y if $x_i y_i \geq 0$ (same orthant) and $|x_i| \leq |y_i|$ for all i

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

Graver Bases

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

x is conformal to y if $x_i y_i \geq 0$ (same orthant) and $|x_i| \leq |y_i|$ for all i

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

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

Graver Bases

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

x is conformal to y if $x_i y_i \geq 0$ (same orthant) and $|x_i| \leq |y_i|$ for all i

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

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

non-circuits: $\pm(1\ -1\ 1)$

Some Theorems on
(Non)-Linear Integer Programming

Shmuel Onn

Some Theorems on
(Non)-Linear Integer Programming

Theorem 1: Linear optimization in polytime with $G(A)$:

$$\max \{ wx \ : \ Ax = b, \ \ l \le x \le u, \ \ x \text{ in } Z^n \}$$

Shmuel Onn

Some Theorems on
(Non)-Linear Integer Programming

Theorem 1: Linear optimization in polytime with $G(A)$:

$$\max \{ wx \ : \ Ax = b, \ \ l \le x \le u, \ \ x \text{ in } Z^n \}$$

Reference: N-fold integer programming, (De Loera, Hemmecke, Onn, Weismantel)
Discrete Optimization (Volume in memory of George Dantzig)

Shmuel Onn

Some Theorems on
(Non)-Linear Integer Programming

Theorem 2: **Separable convex minimization** in **polytime** with **$G(A)$**:

$$\min \{ \sum f_i(x_i) \ : \ Ax = b, \quad l \le x \le u, \quad x \text{ in } Z^n \}$$

Shmuel Onn

Some Theorems on (Non)-Linear Integer Programming

Theorem 2: **Separable convex minimization** in **polytime** with **$G(A)$**:

$$\min \{ \sum f_i(x_i) \ : \ Ax = b, \quad l \le x \le u, \quad x \text{ in } Z^n \}$$

Reference: A polynomial oracle-time algorithm for convex integer minimization, (Hemmecke, Onn, Weismantel), Mathematical Programming

Shmuel Onn

Some Theorems on
(Non)-Linear Integer Programming

Theorem 3: **Multicriteria maximization** in **polytime** with $G(A)$:

$$\max \{ f(Wx) : \ Ax = b, \ \ l \le x \le u, \ \ x \text{ in } Z^n \}$$

where W is $d \times n$ criteria matrix and f convex function on Z^d
which balances d linear criteria or player utilities $W_i x$

Shmuel Onn

Some Theorems on (Non)-Linear Integer Programming

Theorem 3: **Multicriteria maximization** in **polytime** with **G(A)**:

$$\max \{ f(Wx) : \ Ax = b, \ \ l \le x \le u, \ \ x \text{ in } Z^n \}$$

where W is $d \times n$ criteria matrix and f convex function on Z^d
which balances d linear criteria or player utilities $W_i x$

Reference: Convex integer maximization via Graver bases,
(De Loera, Hemmecke, Onn, Rothblum, Weismantel), J. Pure & Applied Algebra

Some Theorems on (Non)-Linear Integer Programming

Theorem 4: **Integer point closest** to x in **polytime** with $G(A)$:

$$\min \{ |x - x|_p \ : \ Ax = b, \ \ l \le x \le u, \ \ x \text{ in } Z^n \}$$

Reference: A polynomial oracle-time algorithm for convex integer minimization, (Hemmecke, Onn, Weismantel), Mathematical Programming

Some Theorems on (Non)-Linear Integer Programming

Theorem 5: Quadratic minimization in **polytime** with **$G(A)$**:

$$\min \{x^T V x \ : \ Ax = b, \quad l \le x \le u, \quad x \text{ in } Z^n\}$$

where V lies in cone $K_2(A)$ of possibly indefinite matrices, enabling minimization of some convex and some non-convex quadratics

Some Theorems on (Non)-Linear Integer Programming

Theorem 5: **Quadratic minimization** in **polytime** with **G(A)**:

$$\min \{x^T V x \ : \ Ax = b, \quad l \le x \le u, \quad x \text{ in } Z^n\}$$

where V lies in cone $K_2(A)$ of possibly indefinite matrices, enabling minimization of some convex and some non-convex quadratics

Reference: Quadratic Graver cones, quadratic integer minimization & extensions, (Lee, Onn, Romanchuk, Weismantel), Mathematical Programming

Some Theorems on
(Non)-Linear Integer Programming

Theorem 6: **Polynomial minimization** in **polytime** with **$G(A)$**:

$$\min \{ p(x) \ : \ Ax = b, \ \ l \le x \le u, \ \ x \text{ in } Z^n \}$$

where p is possibly indefinite polynomial of degree d in cone $K_d(A)$,
enabling minimization of some (non)-convex degree d polynomials

Reference: Quadratic Graver cones, quadratic integer minimization & extensions,
(Lee, Onn, Romanchuk, Weismantel), Mathematical Programming

Some Theorems on (Non)-Linear Integer Programming

Theorem 7: **Robust optimization** in **polytime** with $G(A)$:

$$\min \{ \max \{ cx : d \le c \le e \} : Ax = b, \quad l \le x \le u, \quad x \text{ in } Z^n \}$$

that is, minimum worst case cost where the cost of each variable can vary in an interval

Some Theorems on (Non)-Linear Integer Programming

Theorem 7: **Robust optimization** in **polytime** with $G(A)$:

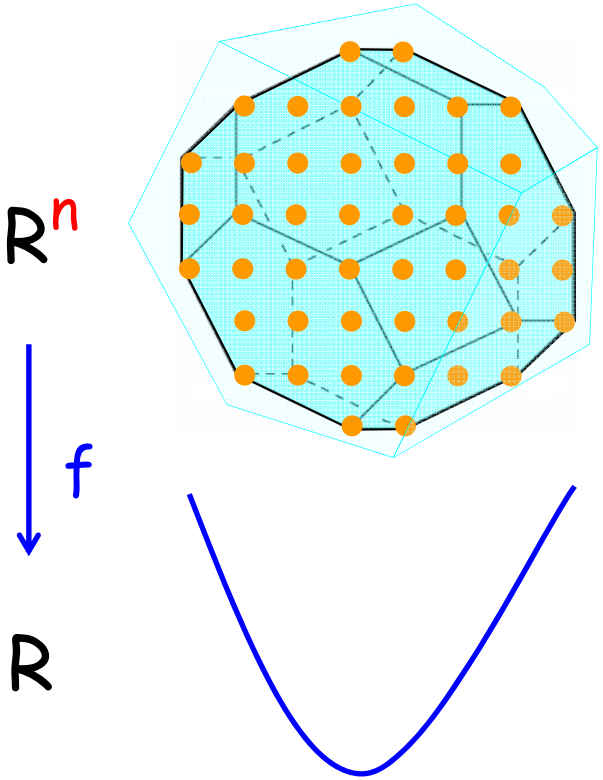
$$\min \{ \max \{ cx : d \le c \le e \} : Ax = b, \quad l \le x \le u, \quad x \text{ in } Z^n \}$$

that is, minimum worst case cost where the cost of each variable can vary in an interval

Reference: Robust integer programming, (Onn), Operations Research Letters

Some Proofs

Proof of Theorems 1-2
(separable convex minimization)

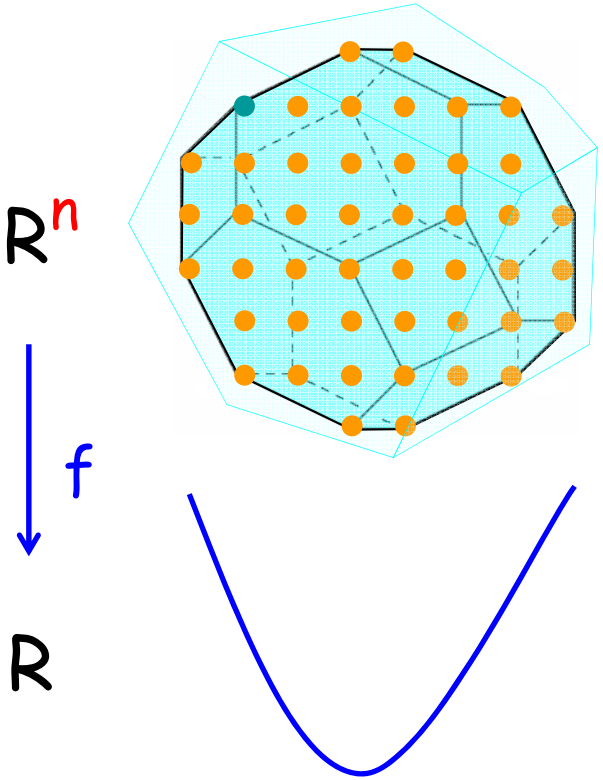


To solve $\min\{\sum f_i(x_i) : Ax = b,\ l \le x \le u,\ x \in Z^n\}$

with the Graver basis $G(A)$

Do:

Proof of Theorems 1-2
(separable convex minimization)



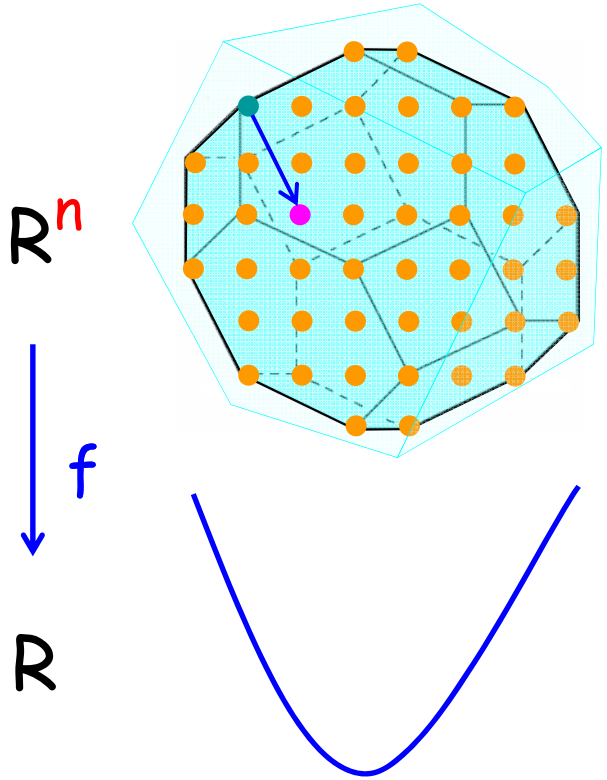
To solve $\min\{\sum f_i(x_i) : Ax = b,\ l \le x \le u,\ x \in Z^n\}$

with the Graver basis $G(A)$

Do:

1. Find initial point by auxiliary program

Proof of Theorems 1-2
(separable convex minimization)



To solve $\min\{\sum f_i(x_i) : Ax = b,\ l \le x \le u,\ x \in Z^n\}$

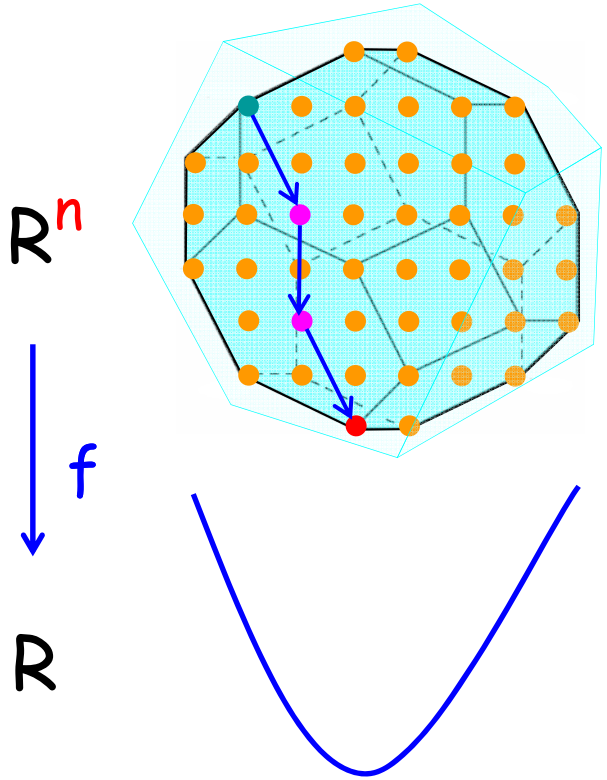
with the Graver basis $G(A)$

Do:

1. Find initial point by auxiliary program

2. Iteratively improve by Graver-best steps, that is, by best cz with $c \in Z$ and $z \in G(A)$.

Proof of Theorems 1-2
(separable convex minimization)



To solve $\min\{\sum f_i(x_i) : Ax = b,\ l \le x \le u,\ x \in Z^n\}$

with the Graver basis $G(A)$

Do:

1. Find initial point by auxiliary program

2. Iteratively improve by Graver-best steps, that is, by best cz with $c \in Z$ and $z \in G(A)$.

Integer Caratheodory Theorem and Structure of f imply polytime convergence to some optimal solution

Proof of Convergence for Linear Maximization

$$\max \{ wx \ : \ Ax = b, \ \ l \le x \le u, \ \ x \in Z^n \}$$

Proof of Convergence for Linear Maximization

$$\max \{ wx \ : \ Ax = b, \ \ l \le x \le u, \ \ x \in Z^n \}$$

Lemma: (Integer Caratheodory) Any x with $Ax = 0$ is expressible as a conformal sum $x = \sum c_i z_i$ of $p \le 2n-2$ elements with $c_i \in Z$ and $z_i \in G(A)$.

Proof of Convergence for Linear Maximization

$$\max \{ wx \ : \ Ax = b, \ \ l \le x \le u, \ \ x \in Z^n \}$$

Lemma: (Integer Caratheodory) Any x with $Ax = 0$ is expressible as a conformal sum $x = \sum c_i z_i$ of $p \le 2n-2$ elements with $c_i \in Z$ and $z_i \in G(A)$.

Let x_k be a point at iteration and x^* optimal and write $x^* - x_k = \sum c_i z_i$

Proof of Convergence for Linear Maximization

$$\max \{ wx \ : \ Ax = b, \ \ l \le x \le u, \ \ x \in Z^n \}$$

Lemma: (Integer Caratheodory) Any x with $Ax = 0$ is expressible as a conformal sum $x = \sum c_i z_i$ of $p \le 2n-2$ elements with $c_i \in Z$ and $z_i \in G(A)$.

Let x_k be a point at iteration and x^* optimal and write $x^* - x_k = \sum c_i z_i$

Then $wx^* - wx_k = \sum wc_i z_i$ so for some i we have $\frac{1}{p}(wx^* - wx_k) \le wc_i z_i$

Proof of Convergence for Linear Maximization

$$\max \{ wx \ : \ Ax = b, \ \ l \le x \le u, \ \ x \in Z^n \}$$

Lemma: (Integer Caratheodory) Any x with $Ax = 0$ is expressible as a conformal sum $x = \sum c_i z_i$ of $p \le 2n-2$ elements with $c_i \in Z$ and $z_i \in G(A)$.

Let x_k be a point at iteration and x^* optimal and write $x^* - x_k = \sum c_i z_i$

Then $wx^* - wx_k = \sum wc_i z_i$ so for some i we have $\frac{1}{p}(wx^* - wx_k) \le wc_i z_i$

Now $x_k + c_i z_i$ is feasible since the sum is conformal

Proof of Convergence for Linear Maximization

$$\max\{\, wx \;:\; Ax = b,\;\; l \le x \le u,\;\; x \in Z^n \,\}$$

Lemma: (Integer Caratheodory) Any x with $Ax = 0$ is expressible as a conformal sum $x = \sum c_i z_i$ of $p \le 2n-2$ elements with $c_i \in Z$ and $z_i \in G(A)$.

Let x_k be a point at iteration and x^* optimal and write $x^* - x_k = \sum c_i z_i$

Then $wx^* - wx_k = \sum wc_i z_i$ so for some i we have $\frac{1}{p}(wx^* - wx_k) \le wc_i z_i$

Now $x_k + c_i z_i$ is feasible since the sum is conformal

So if cz is a Graver-best step and $x_{k+1} = x_k + cz$ then we obtain

$$wx^* - wx_{k+1} = wx^* - w(x_k + cz) \le wx^* - w(x_k + c_i z_i) \le \frac{p-1}{p}(wx^* - wx_k)$$

Shmuel Onn

Proof of Convergence for Linear Maximization

$$\max\{\, wx \;:\; Ax = b,\;\; l \le x \le u,\;\; x \in Z^n \,\}$$

Lemma: The number of iterations is $O(nL)$ with L the bit size of w, A, b, l, u

Let x_k be a point at iteration and x^* optimal and write $x^* - x_k = \sum c_i z_i$

Then $wx^* - wx_k = \sum w c_i z_i$ so for some i we have $\frac{1}{p}(wx^* - wx_k) \le w c_i z_i$

Now $x_k + c_i z_i$ is feasible since the sum is conformal

So if cz is a Graver-best step and $x_{k+1} = x_k + cz$ then we obtain

$$wx^* - wx_{k+1} = wx^* - w(x_k + cz) \le wx^* - w(x_k + c_i z_i) \le \frac{p-1}{p}(wx^* - wx_k)$$

Shmuel Onn

Proof of Convergence for Linear Maximization

$$\max \{ wx \ : \ Ax = b, \quad l \le x \le u, \quad x \in Z^n \}$$

Lemma: The number of iterations is $O(nL)$ with L the bit size of w, A, b, l, u

In fact, very recently Koutecký, Levin, Onn show:

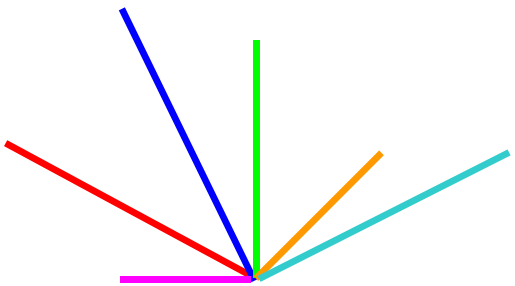
Theorem: The number of iterations is strongly polynomial $\mathrm{poly}(n)$

Proof of Theorem 3
(multicriteria maximization)

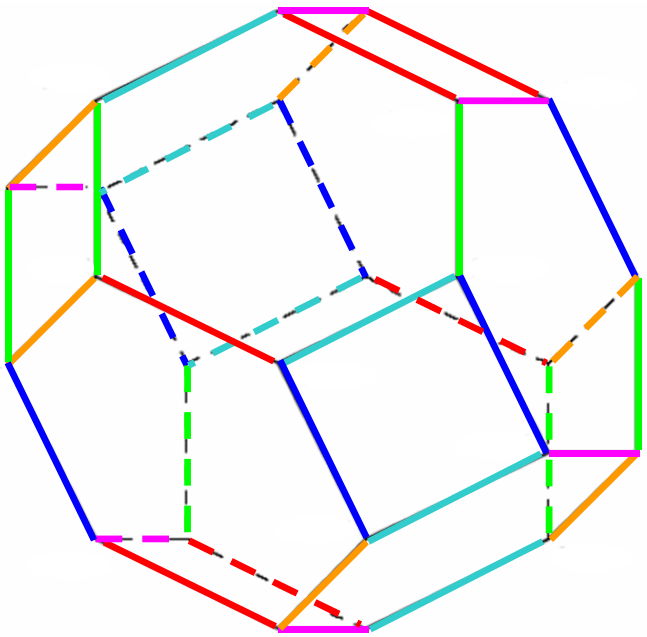
Lemma: Can solve in polytime max { $f(Wx)$: x in S } with S in Z^n if can do linear optimization over S and have set E of all edge-directions of conv(S)

Proof of Theorem 3
(multicriteria maximization)

Lemma: Can solve in polytime max { $f(Wx)$: x in S } with S in Z^n if can do linear optimization over S and have set E of all edge-directions of conv(S)



set E of all edge-directions of conv(S)



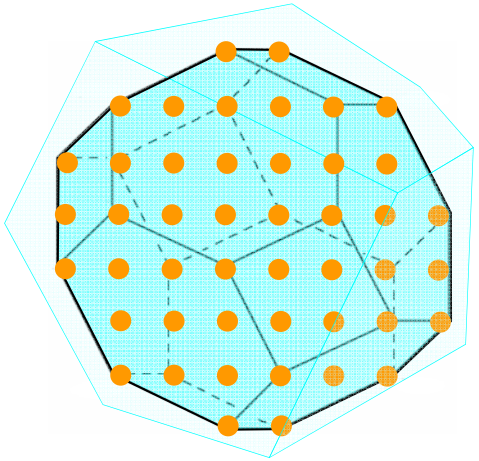
Shmuel Onn

Proof of Theorem 3
(multicriteria maximization)

Lemma: Can solve in polytime max { $f(Wx)$: x in S } with S in Z^n if can do linear optimization over S and have set E of all edge-directions of conv(S)

Reference: Convex combinatorial optimization, (Onn, Rothblum), Journal of Discrete and Computational Geometry

Proof of Theorem 3
(multicriteria maximization)



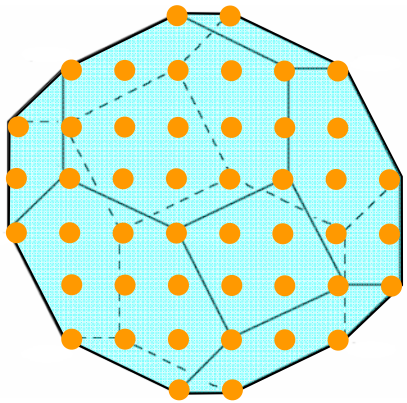
To solve $\max \{ f(Wx) \ : \ x \text{ in } S \}$

with $S := \{x \text{ in } Z^n : Ax = b, \ l \le x \le u\}$

using the Graver basis $G(A)$

Do:

Proof of Theorem 3
(multicriteria maximization)



To solve $\max \{ f(Wx) \ : \ x \text{ in } S \}$

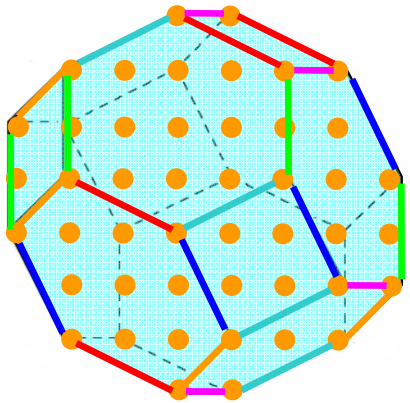
with $S := \{x \text{ in } Z^n : Ax = b, \ l \le x \le u\}$

using the Graver basis $G(A)$

Do:

1. Use $G(A)$ to simulate linear-optimization oracle over S via Theorem 1

Proof of Theorem 3
(multicriteria maximization)



To solve $\max\{ f(Wx) : x \text{ in } S \}$

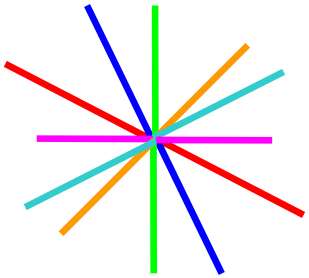
with $S := \{x \text{ in } Z^n : Ax = b,\ l \le x \le u\}$

using the Graver basis $G(A)$

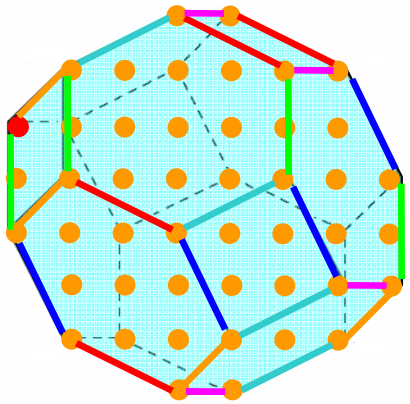
Do:

1. Use $G(A)$ to simulate linear-optimization oracle over S via Theorem 1

2. Use the Graver basis as set $E := G(A)$ of all edge-directions of $\mathrm{conv}(S)$



Proof of Theorem 3
(multicriteria maximization)



To solve $\max \{ f(Wx) \ : \ x \text{ in } S \}$

with $S := \{x \text{ in } Z^n : Ax = b, \ l \le x \le u\}$

using the Graver basis $G(A)$

Do:

1. Use $G(A)$ to simulate linear-optimization oracle over S via Theorem 1

2. Use the Graver basis as set $E := G(A)$ of all edge-directions of conv(S)

3. Apply the Lemma (use 1 for each vertex of zone($WG(A)$) and pick best)

N-Fold Integer Programming

N-Fold Products

The n-fold product of an $(r,s) \times t$ bimatrix $A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$ is the $(r+ns) \times nt$ matrix

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

N-Fold Products

The n-fold product of an $(r,s) \times t$ bimatrix $A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$ is the $(r+ns) \times nt$ matrix

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

Lemma: The Graver basis $G(A^{(n)})$ can be computed in polytime $O(n^{g(A)})$ where $g(A)$ is the so-called Graver complexity of the bimatrix A.

(Non)-Linear N-Fold Integer Programming

Theorem: for various f can solve in polynomial time $O(n^{g(A)} L)$:

$$\min \{f(x) : A^{(n)}x = b,\ l \le x \le u,\ x \in Z^{nt}\}$$

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

(Non)-Linear N-Fold Integer Programming

Theorem: for various f can solve in polynomial time $O(n^{g(A)} L)$:

$$\min \{f(x) : A^{(n)}x = b,\ l \le x \le u,\ x \in Z^{nt}\}$$

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

Proof: Use Lemma to construct in polytime the Graver base $G(A^{(n)})$.

Now apply and use Theorems 1 - 7 to optimize in polytime.

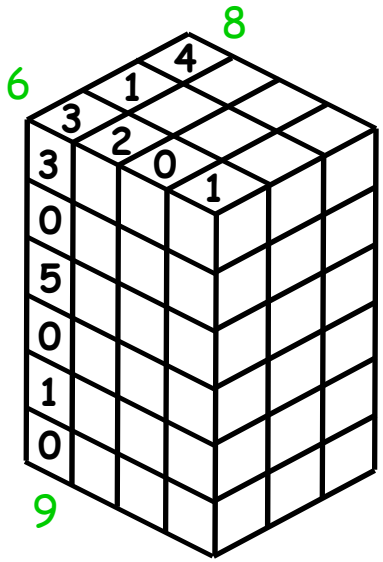
Generic Example:

Multiway Tables

Multiway Tables

Complexity of deciding the existence of

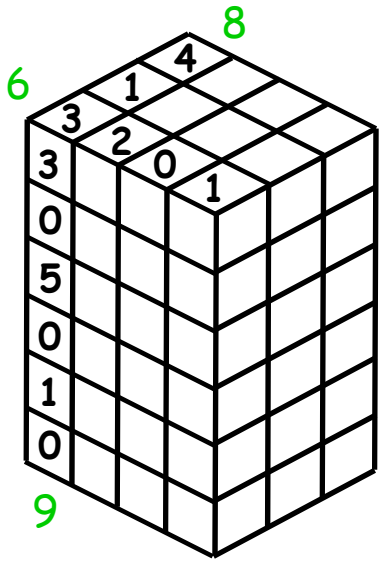
$l \times m \times n$ tables with given line sums:



Multiway Tables

Complexity of deciding the existence of

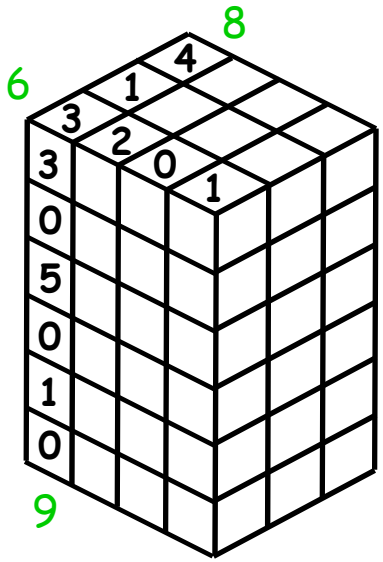
$l \times m \times n$ tables with given line sums:



$$\{\, x \in Z^{l \times m \times n} \;:\; \Sigma_i x_{i,j,k} = a_{j,k}\,,\; \Sigma_j x_{i,j,k} = b_{i,k}\,,\; \Sigma_k x_{i,j,k} = c_{i,j}\,,\; x \geq 0 \,\}$$

Shmuel Onn

Multiway Tables



Complexity of deciding the existence of

$l \times m \times n$ tables with given line sums:

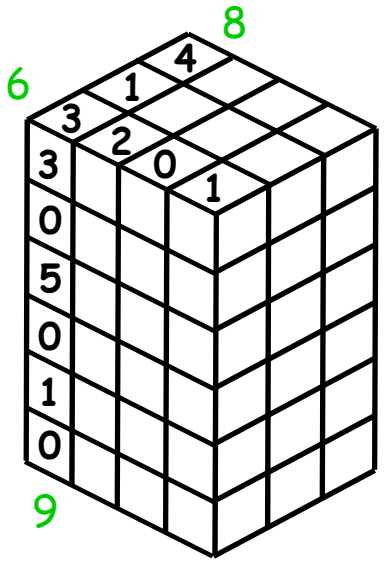
- l, m, n variable:

- l fixed, m, n variable:

- l, m fixed, n variable:

- l, m, n fixed:

Multiway Tables



Complexity of deciding the existence of

$l \times m \times n$ tables with given line sums:

- l, m, n variable: NP-complete

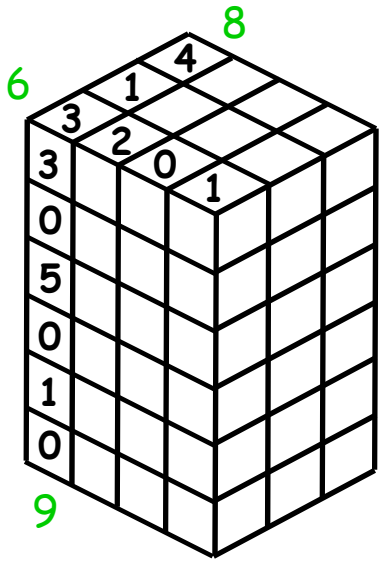
  Via 3-dimensional matching, Irving, Jerrum

- l fixed, m, n variable:

- l, m fixed, n variable:

- l, m, n fixed:

Multiway Tables



Complexity of deciding the existence of

$l \times m \times n$ tables with given line sums:

- l, m, n variable: NP-complete

  Via 3-dimensional matching, Irving, Jerrum

- l fixed, m, n variable:

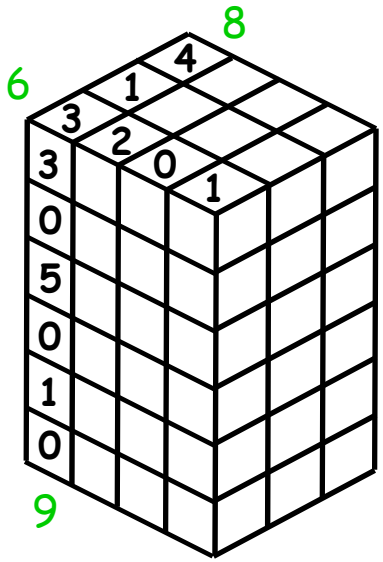
- l, m fixed, n variable:

- l, m, n fixed: Polytime

  Integer programming in fixed dimension, Lenstra

Shmuel Onn

Multiway Tables



Complexity of deciding the existence of

$l \times m \times n$ tables with given line sums:

- l, m, n variable: NP-complete

  Via 3-dimensional matching, Irving, Jerrum

- l fixed, m, n variable: Universal for IP (even with $l=3$)

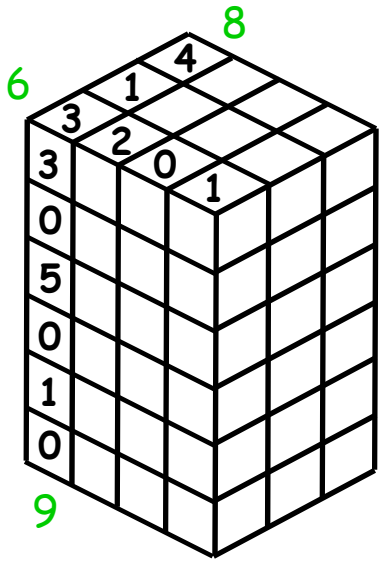
  De Loera, Onn

- l, m fixed, n variable:

- l, m, n fixed: Polytime

  Integer programming in fixed dimension, Lenstra

Multiway Tables



Complexity of deciding the existence of

$l \times m \times n$ tables with given line sums:

- l, m, n variable: NP-complete

  Via 3-dimensional matching, Irving, Jerrum

- l fixed, m, n variable: Universal for IP (even with $l=3$)

  De Loera, Onn

- l, m fixed, n variable: Polytime

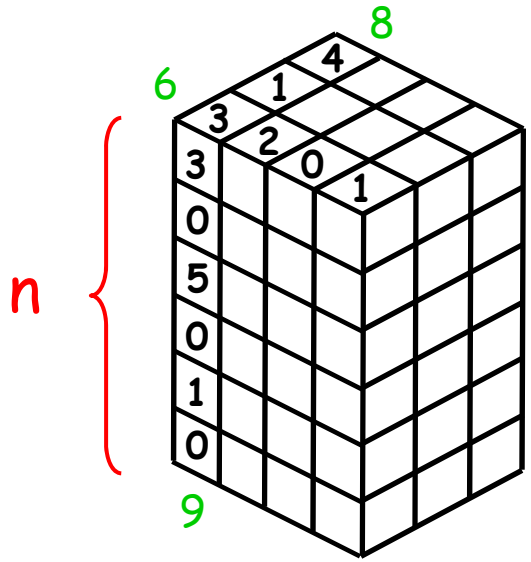
  Consequence of linear n-fold IP, De Loera, Hemmecke, Onn, Weismantel

- l, m, n fixed: Polytime

  Integer programming in fixed dimension, Lenstra

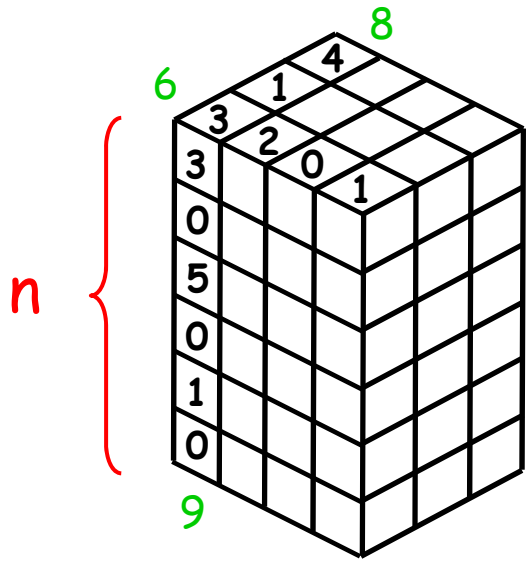
Multiway Tables

Much more generally, consider the multi-index transportation problem studied by Motzkin in 1952, of minimization over $m_1 \times \cdots \times m_k \times n$ tables with given margins:



Multiway Tables

Much more generally, consider the multi-index transportation problem studied by Motzkin in 1952, of minimization over $m_1 \times \cdots \times m_k \times n$ tables with given margins:



It is an n-fold program

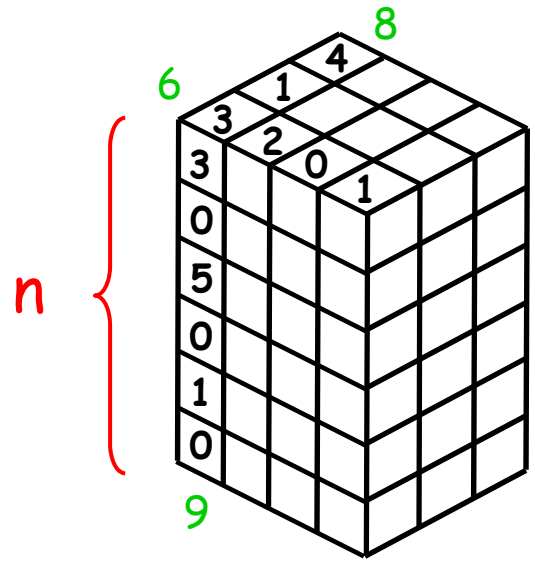
$$\min\{f(x) \;:\; A^{(n)}x = b, \quad x \geq 0, \quad x \text{ integer}\}$$

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

for suitable A depending on $m_1, \ldots, m_k$ where:

Multiway Tables

Much more generally, consider the multi-index transportation problem studied by Motzkin in 1952, of minimization over $m_1 \times \cdots \times m_k \times n$ tables with given margins:



It is an n-fold program

$$\min\{f(x) \;:\; A^{(n)}x = b, \quad x \geq 0, \quad x \text{ integer}\}$$

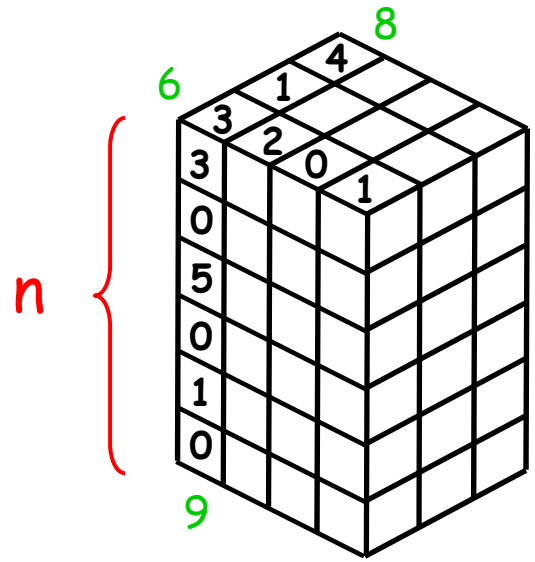
$$A^{(n)} = \underbrace{\begin{pmatrix} A_1 & A_1 & A_1 & \cdots & A_1 \\ A_2 & 0 & 0 & \cdots & 0 \\ 0 & A_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A_2 \end{pmatrix}}_{n}$$

for suitable A depending on $m_1, \ldots, m_k$ where:

- A_2 gives equations of margins summing within a single layer at a time

Multiway Tables

Much more generally, consider the multi-index transportation problem studied by Motzkin in 1952, of minimization over $m_1 \times \cdots \times m_k \times n$ tables with given margins:



It is an n-fold program

$$\min\{f(x) \;:\; A^{(n)}x = b, \quad x \geq 0, \quad x \text{ integer}\}$$

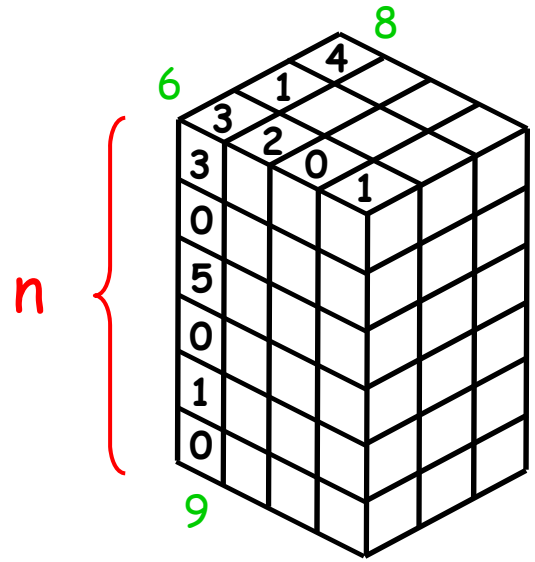
$$A^{(n)} = \underbrace{\begin{pmatrix} A_1 & A_1 & A_1 & \cdots & A_1 \\ A_2 & 0 & 0 & \cdots & 0 \\ 0 & A_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A_2 \end{pmatrix}}_{n}$$

for suitable A depending on $m_1, \ldots, m_k$ where:

- A_2 gives equations of margins summing within a single layer at a time
- A_1 gives equations of margins summing over layers (for lines $A_1 = I$)

Multiway Tables

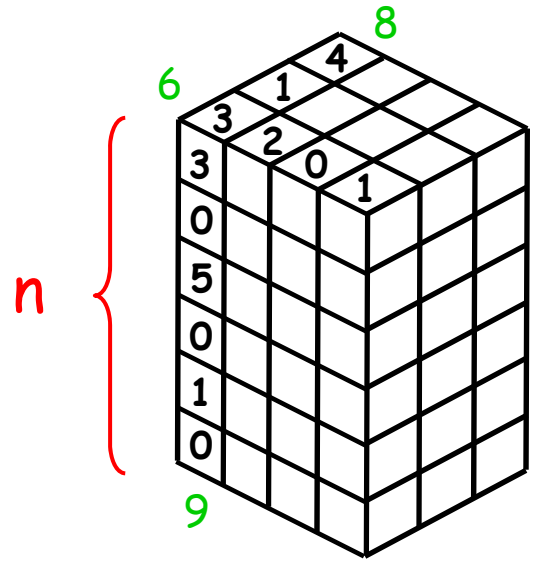
Much more generally, consider the multi-index transportation problem studied by Motzkin in 1952, of minimization over $m_1 \times \cdots \times m_k \times n$ tables with given margins:



Corollary: (Non)-linear optimization over $m_1 \times \cdots \times m_k \times n$ tables with given margins can be done in polynomial time $O(n^{g(m_1,\ldots,m_k)} L)$

Multiway Tables

Much more generally, consider the multi-index transportation problem studied by Motzkin in 1952, of minimization over $m_1 \times \cdots \times m_k \times n$ tables with given margins:

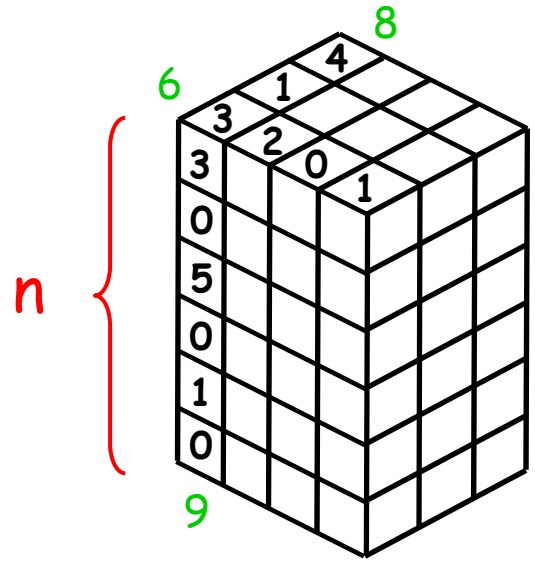


Corollary: (Non)-linear optimization over $m_1 \times \cdots \times m_k \times n$ tables with given margins can be done in polynomial time $O(n^{g(m_1,\ldots,m_k)} L)$

Universality of Three-Way Tables (De Loera, Onn):
Every integer program is one over $3 \times m \times n$ tables with given line-sums

Multiway Tables

Much more generally, consider the multi-index transportation problem studied by Motzkin in 1952, of minimization over $m_1 \times \cdots \times m_k \times n$ tables with given margins:



Corollary: (Non)-linear optimization over $m_1 \times \cdots \times m_k \times n$ tables with given margins can be done in polynomial time $O(n^{g(m_1,\ldots,m_k)} L)$

Universality of Three-Way Tables (De Loera, Onn):
Every integer program is one over $3 \times m \times n$ tables with given line-sums

Natural parameterization by width m of tables and integer programming

Lecture 2

Outline: Lecture 2

N-Fold IP is Fixed-Parameter Tractable

Outline: Lecture 2

N-Fold IP is Fixed-Parameter Tractable

Huge Multicommodity Flows

Outline: Lecture 2

N-Fold IP is Fixed-Parameter Tractable

Huge Multicommodity Flows

More Applications, Extensions, Directions

N-Fold Integer Programming

is

Fixed-Parameter Tractable

Parameterization

Consider n-fold integer programming over $(r,s) \times t$ bimatrix $A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$

$$\min \{wx : A^{(n)}x = b,\ \ l \le x \le u,\ \ x \in Z^{nt}\}.$$

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

Parameterization

Consider n-fold integer programming over $(r,s) \times t$ bimatrix $A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$

$$\min \{wx : A^{(n)}x = b,\ \ l \le x \le u,\ \ x \in Z^{nt}\}.$$

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

The parameter is A or rather r,s,t, max A

Parameterization

Consider n-fold integer programming over $(r,s) \times t$ bimatrix $A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$

$$\min \{wx : A^{(n)}x = b,\ \ l \le x \le u,\ \ x \in Z^{nt}\}.$$

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

The parameter is A or rather r,s,t, max A

The input is n and the bit size L of w,b,l,u

Graver Complexity

Lemma: Every bimatrix A has finite Graver complexity $g(A)$ so that for all n, any $z = (z^1, z^2, z^3, \ldots, z^n) \in G(A^{(n)})$ has at most $g(A)$ nonzero bricks.

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

Graver Complexity

Lemma: Every bimatrix A has finite Graver complexity $g(A)$ so that for all n, any $z = (z^1, z^2, z^3, \ldots, z^n) \in G(A^{(n)})$ has at most $g(A)$ nonzero bricks.

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

Moreover, $g(A) = \max \{ |v|_1 \ : \ v \in G(A_1 G(A_2)) \}$

Graver Complexity

Lemma: Every bimatrix A has finite Graver complexity $g(A)$ so that for all n, any $z = (z^1, z^2, z^3, \ldots, z^n) \in G(A^{(n)})$ has at most $g(A)$ nonzero bricks.

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

Moreover, $g(A) = \max \{ |v|_1 \ : \ v \in G(A_1 G(A_2)) \}$

Idea: $A^{(n)}z = 0$ implies $A_2 z^i = 0$ for all i. Call z pure if $z^i \in G(A_2)$ for all i. Enough to bound the number of bricks of pure z. Then suitable counting of the number of bricks z^i equal to each $h \in G(A_2)$ leads to the result.

Graver Complexity

Lemma: Every bimatrix A has finite Graver complexity $g(A)$ so that for all n, any $z = (z^1, z^2, z^3, \ldots, z^n) \in G(A^{(n)})$ has at most $g(A)$ nonzero bricks.

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

Lemma: The Graver basis $G(A^{(n)})$ can be computed in polytime $O(n^{g(A)})$.

Graver Complexity

Lemma: Every bimatrix A has finite Graver complexity $g(A)$ so that for all n, any $z = (z^1, z^2, z^3, \ldots, z^n) \in G(A^{(n)})$ has at most $g(A)$ nonzero bricks.

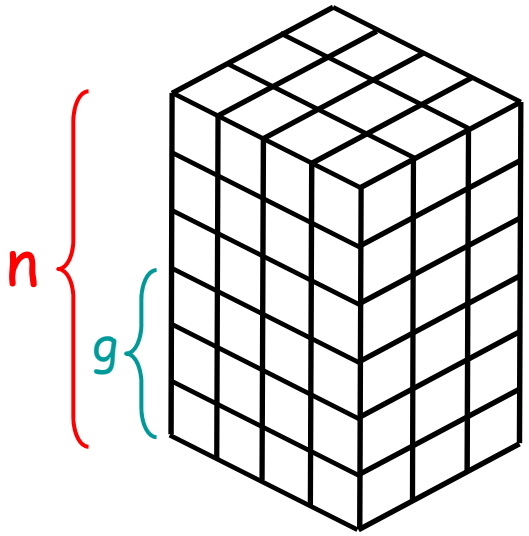
$$A^{(n)} = \begin{pmatrix} A_1 & A_1 & A_1 & \cdots & A_1 \\ A_2 & 0 & 0 & \cdots & 0 \\ 0 & A_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A_2 \end{pmatrix} .$$

Lemma: The Graver basis $G(A^{(n)})$ can be computed in polytime $O(n^{g(A)})$.

Proof: $G(A^{(n)})$ consists of the $\binom{n}{g(A)}$ liftings of elements of $G(A^{(g(A))})$.

Graver Complexity

Lemma: Every bimatrix A has finite Graver complexity $g(A)$ so that for all n, any $z = (z^1, z^2, z^3, \ldots, z^n) \in G(A^{(n)})$ has at most $g(A)$ nonzero bricks.



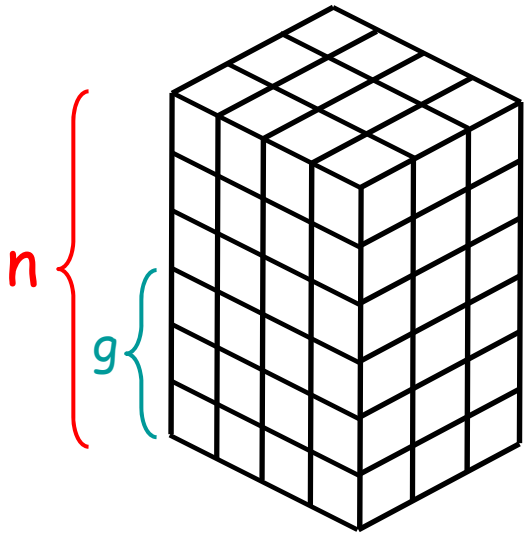
The Graver complexity is large and hard: for example, all we know about $g(m)$ of the universal n-fold program over $3 \times m \times n$ tables is

$g(3) = 9$, $g(4) = 27$, $g(5) \geq 75$, and $\Omega(2^m) = g(m) = O(6^m)$.

Shmuel Onn

Graver Complexity

Lemma: Every bimatrix A has finite Graver complexity $g(A)$ so that for all n, any $z = (z^1, z^2, z^3, \ldots, z^n) \in G(A^{(n)})$ has at most $g(A)$ nonzero bricks.



The Graver complexity is large and hard: for example, all we know about $g(m)$ of the universal n-fold program over $3 \times m \times n$ tables is

$g(3) = 9$, $g(4) = 27$, $g(5) \geq 75$, and $\Omega(2^m) = g(m) = O(6^m)$.

Is $g(m) = 3^{m-1}$?

Shmuel Onn

Fixed-Parameter Tractability

Reference: N-fold integer programming in cubic time,
(Hemmecke, Onn, Romanchuk), Mathematical Programming

Lyubov Romanchuk, former graduate student with me at Technion

Finding a Graver-Best Step

Let x be feasible at some iteration of solving the n-fold program

$$\min \{wx : A^{(n)}x = b,\ \ l \le x \le u,\ \ x \in Z^{nt}\}.$$

Finding a Graver-Best Step

Let x be feasible at some iteration of solving the n-fold program

$$\min \{wx : A^{(n)}x = b,\ \ l \le x \le u,\ \ x \in Z^{nt}\}.$$

A Graver-best step for x is vector h such that $x+h$ is feasible and at least as good as any feasible $x+cz$ with $c \in Z$ and $z \in G(A^{(n)})$.

Finding a Graver-Best Step

Let x be feasible at some iteration of solving the n-fold program

$$\min \{wx : A^{(n)}x = b,\ \ l \le x \le u,\ \ x \in Z^{nt}\}.$$

A Graver-best step for x is vector h such that $x+h$ is feasible and at least as good as any feasible $x+cz$ with $c \in Z$ and $z \in G(A^{(n)})$.

Previous way: for each of $O(n^{g(A)})$ elements $z \in G(A^{(n)})$ find best $c \in Z$

Finding a Graver-Best Step

Let x be feasible at some iteration of solving the n-fold program

$$\min \{wx : A^{(n)}x = b,\ \ l \le x \le u,\ \ x \in Z^{nt}\}.$$

A Graver-best step for x is vector h such that $x+h$ is feasible and at least as good as any feasible $x+cz$ with $c \in Z$ and $z \in G(A^{(n)})$.

Previous way: for each of $O(n^{g(A)})$ elements $z \in G(A^{(n)})$ find best $c \in Z$

New way: for each c find ch at least as good as any cz with $z \in G(A^{(n)})$ without computing the entire Graver basis.

Finding Graver-Best Step: New Way

Let x be feasible at some iteration of solving the n-fold program

$$\min \{wx : A^{(n)}x = b,\ \ l \le x \le u,\ \ x \in Z^{nt}\}.$$

For each $c \in Z$ construct a dynamic program as follows:

Finding Graver-Best Step: New Way

Let x be feasible at some iteration of solving the n-fold program

$$\min \{wx : A^{(n)}x = b,\ \ l \le x \le u,\ \ x \in Z^{nt}\}.$$

For each $c \in Z$ construct a dynamic program as follows:

Let $V(A) := \{v \in Z^{t} : v$ is the sum of at most $g(A)$ elements of $G(A_2)\}$

Finding Graver-Best Step: New Way

Let x be feasible at some iteration of solving the n-fold program

$$\min \{wx : A^{(n)}x = b,\ \ l \le x \le u,\ \ x \in Z^{nt}\}.$$

For each $c \in Z$ construct a dynamic program as follows:

Let $V(A) := \{v \in Z^{t} : v$ is the sum of at most $g(A)$ elements of $G(A_2)\}$

Note that $|V(A)| = O(|G(A_2)|^{g(A)})$

Finding Graver-Best Step: New Way

Let x be feasible at some iteration of solving the n-fold program

$$\min \{wx : A^{(n)}x = b,\ \ l \le x \le u,\ \ x \in Z^{nt}\}.$$

For each $c \in Z$ construct a dynamic program as follows:

Let $V(A) := \{v \in Z^{t} : v$ is the sum of at most $g(A)$ elements of $G(A_2)\}$

Note that $|V(A)| = O(|G(A_2)|^{g(A)})$

Lemma: Any $h = (h^1, \ldots, h^n) \in G(A^{(n)})$ and any i have $h^1 + \ldots + h^i \in V(A)$.

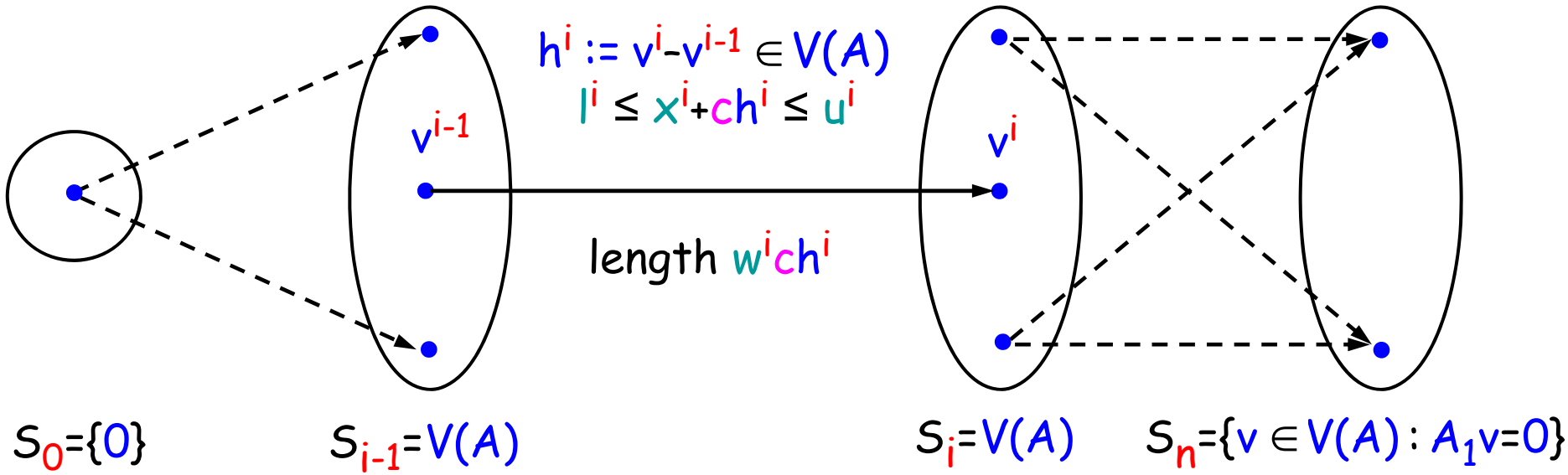
Finding Graver-Best Step: New Way

Let x be feasible at some iteration of solving the n-fold program

$$\min \{wx : A^{(n)}x = b,\ \ l \le x \le u,\ \ x \in Z^{nt}\}.$$

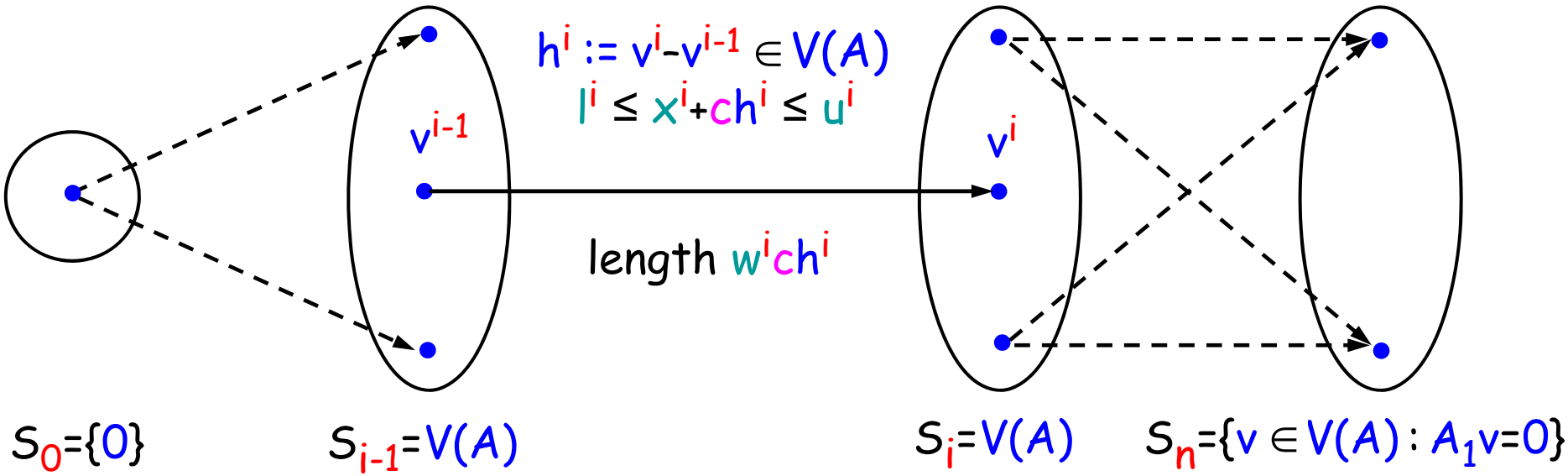
For each $c \in Z$ construct a dynamic program as follows:

Let $V(A) := \{v \in Z^t : v$ is the sum of at most $g(A)$ elements of $G(A_2)\}$



Finding Graver-Best Step: New Way

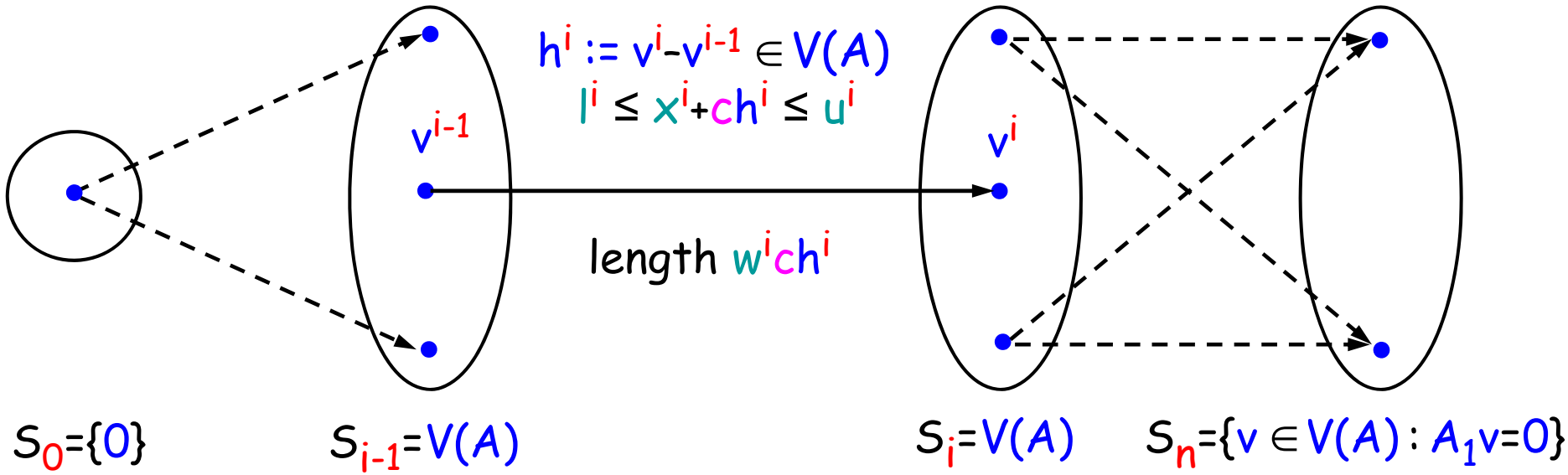
Idea: Each feasible step $ch = (ch^1, \ldots, ch^n)$ with $h \in G(A^{(n)})$ gives dipath $(0, v^1, \ldots, v^n)$ in the dynamic program, of length wch.



Finding Graver-Best Step: New Way

Idea: Each feasible step $ch = (ch^1, \ldots, ch^n)$ with $h \in G(A^{(n)})$ gives dipath $(0, v^1, \ldots, v^n)$ in the dynamic program, of length wch.

Lemma: A Graver-best step for x can be computed in $O(n^2)$ time, solving $O(|G(A_2)|^{g(A)} n)$ dynamic programs each in time $O(|G(A_2)|^{2g(A)} n)$.



Cubic Running Time and Fixed-Parameter Tractability

Theorem: Linear n-fold integer programming parameterized by r,s,t, max A, is solvable in fixed-parameter time $O(n^3 L)$:

$$\min \{wx : A^{(n)}x = b,\ l \le x \le u,\ x \in Z^{nt}\}$$

Reference: N-fold integer programming in cubic time, (Hemmecke, Onn, Romanchuk), Mathematical Programming

Cubic Running Time and Fixed-Parameter Tractability

Theorem: Linear n-fold integer programming parameterized by r,s,t, max A, is solvable in fixed-parameter time $O(n^3 L)$:

$$\min \{wx : A^{(n)}x = b,\ l \le x \le u,\ x \in Z^{nt}\}$$

Instead of $O(n^{g(A)} L)$

Reference: N-fold integer programming in cubic time, (Hemmecke, Onn, Romanchuk), Mathematical Programming

Cubic Running Time and Fixed-Parameter Tractability

Theorem: Linear n-fold integer programming parameterized by r,s,t, max A, is solvable in fixed-parameter time $O(n^3 L)$:

$$\min \{wx : A^{(n)}x = b,\ l \le x \le u,\ x \in Z^{nt}\}$$

With more work we can obtain the following:

Theorem: Separable convex p-piecewise affine n-fold integer programming with parameters r,s,t,p, max A is solvable in fixed-parameter time $O(n^3 L)$:

$$\min \{\sum f_i(x_i) : A^{(n)}x = b,\ l \le x \le u,\ x \in Z^{nt}\}$$

Reference: N-fold integer programming in cubic time, (Hemmecke, Onn, Romanchuk), Mathematical Programming

Example – the Constant for 3x3xn Tables

For $3 \times 3 \times n$ table problems $V(A)$ consists of 42931 matrices such as

$$\begin{pmatrix} 9 & -2 & -7 \\ -4 & 5 & -1 \\ -5 & -3 & 8 \end{pmatrix}$$

Example – the Constant for 3x3xn Tables

For $3 \times 3 \times n$ table problems $V(A)$ consists of 42931 matrices such as

$$\begin{pmatrix} 9 & -2 & -7 \\ -4 & 5 & -1 \\ -5 & -3 & 8 \end{pmatrix}$$

So finding a single Graver-best step in single iteration involves solving $42931 \cdot n$ dynamic programs each consisting of $42931 \cdot n$ matrices using some $10^{14} \cdot n^2$ arithmetic operations per iteration.

Example – the Constant for 3x3xn Tables

For $3 \times 3 \times n$ table problems $V(A)$ consists of 42931 matrices such as

$$\begin{pmatrix} 9 & -2 & -7 \\ -4 & 5 & -1 \\ -5 & -3 & 8 \end{pmatrix}$$

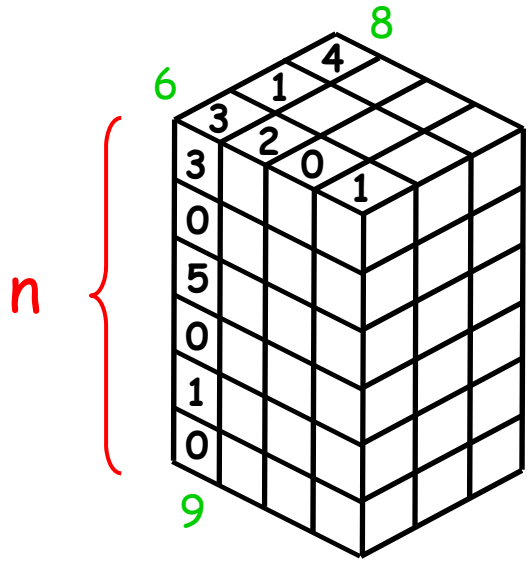
So finding a single Graver-best step in single iteration involves solving $42931 \cdot n$ dynamic programs each consisting of $42931 \cdot n$ matrices using some $10^{14} \cdot n^2$ arithmetic operations per iteration.

Is there a better algorithm for $3 \times 3 \times n$ tables or better constant ?

Multiway Tables are Fixed-Parameter Tractable

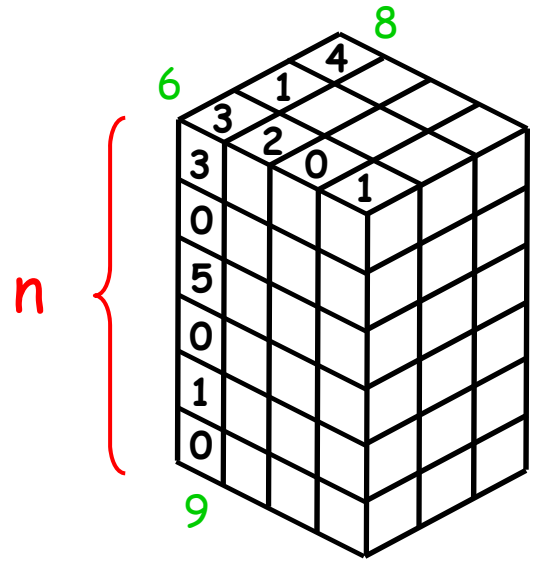
Consider again the problem of minimizing over

$m_1 \times \cdots \times m_k \times n$ tables with given margins:



Multiway Tables are Fixed-Parameter Tractable

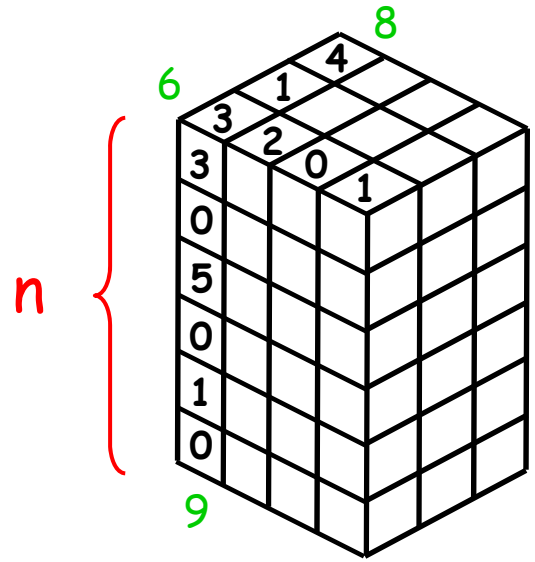
Consider again the problem of minimizing over $m_1 \times \cdots \times m_k \times n$ tables with given margins:



Corollary: (Non)-linear optimization over $m_1 \times \cdots \times m_k \times n$ tables with given margins can be done in fixed-parameter cubic time $O(n^3 L)$

Multiway Tables are Fixed-Parameter Tractable

Consider again the problem of minimizing over $m_1 \times \cdots \times m_k \times n$ tables with given margins:



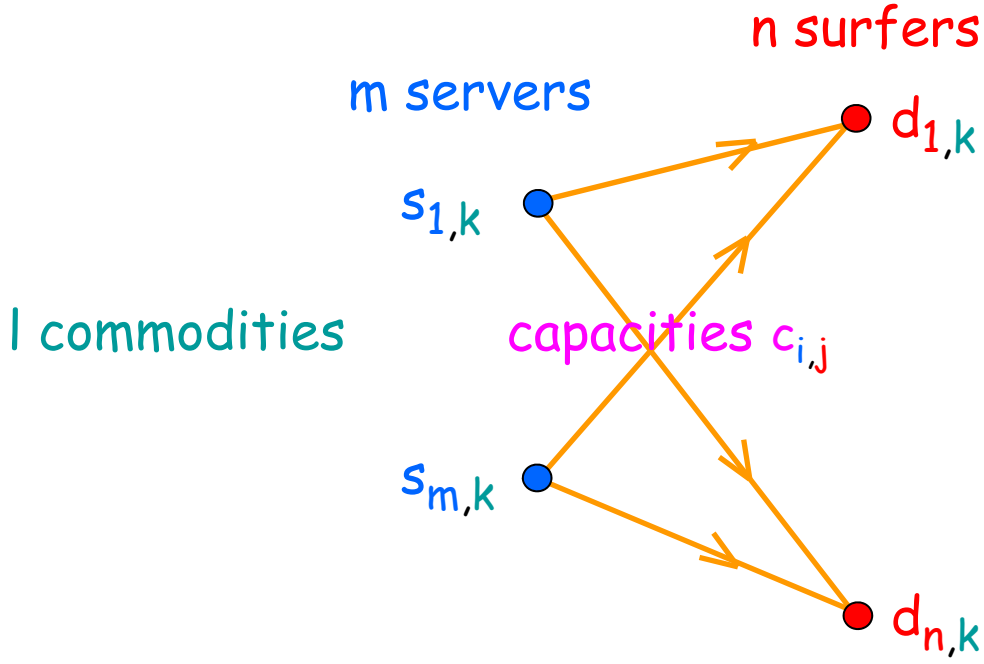
Corollary: (Non)-linear optimization over $m_1 \times \cdots \times m_k \times n$ tables with given margins can be done in fixed-parameter cubic time $O(n^3 L)$

Instead of $O(n^{g(m_1,\ldots,m_k)} L)$

Multicommodity Flows

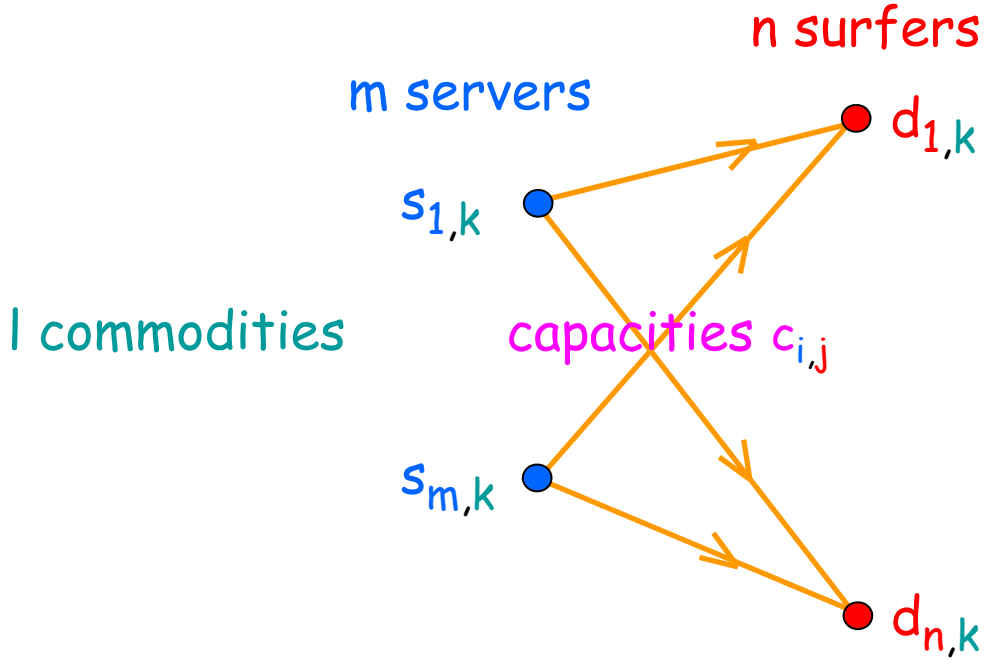
Multicommodity Flows

Find flow of l commodities from m servers to n surfers satisfying given supplies $s_{i,k}$, demands $d_{j,k}$ and capacities $c_{i,j}$ of total bit size L



Multicommodity Flows

Find flow of l commodities from m servers to n surfers satisfying given supplies $s_{i,k}$, demands $d_{j,k}$ and capacities $c_{i,j}$ of total bit size L



With $l=2$ or $m=3$ it is NP-complete so assume both l,m are parameters

Multicommodity Flows

Find flow of l commodities from m servers to n surfers satisfying given supplies $s_{i,k}$, demands $d_{j,k}$ and capacities $c_{i,j}$ of total bit size L

-- polynomial time $O(n^{g(l,m)} L)$ with Graver complexity $g(l,m)$ exponential in l,m (De Loera, Hemmecke, Onn, Weismantel) (theory of n-fold IP)

Multicommodity Flows

Find flow of l commodities from m servers to n surfers satisfying given supplies $s_{i,k}$, demands $d_{j,k}$ and capacities $c_{i,j}$ of total bit size L

-- polynomial time $O(n^{g(l,m)} L)$ with Graver complexity $g(l,m)$ exponential in l,m (De Loera, Hemmecke, Onn, Weismantel) (theory of n-fold IP)

-- fixed-parameter tractable $O(n^3 L)$ (Hemmecke, Onn, Romanchuk)

Shmuel Onn

Multicommodity Flows

Find flow of l commodities from m servers to n surfers satisfying given supplies $s_{i,k}$, demands $d_{j,k}$ and capacities $c_{i,j}$ of total bit size L

-- polynomial time $O(n^{g(l,m)} L)$ with Graver complexity $g(l,m)$ exponential in l,m
(De Loera, Hemmecke, Onn, Weismantel) (theory of n-fold IP)

-- fixed-parameter tractable $O(n^3 L)$
(Hemmecke, Onn, Romanchuk)

-- strongly polynomial $O(n^{g(l,m)})$
(De Loera, Hemmecke, Lee)

Multicommodity Flows

Find flow of l commodities from m servers to n surfers satisfying given supplies $s_{i,k}$, demands $d_{j,k}$ and capacities $c_{i,j}$ of total bit size L

-- polynomial time $O(n^{g(l,m)} L)$ with Graver complexity $g(l,m)$ exponential in l,m
(De Loera, Hemmecke, Onn, Weismantel) (theory of n-fold IP)

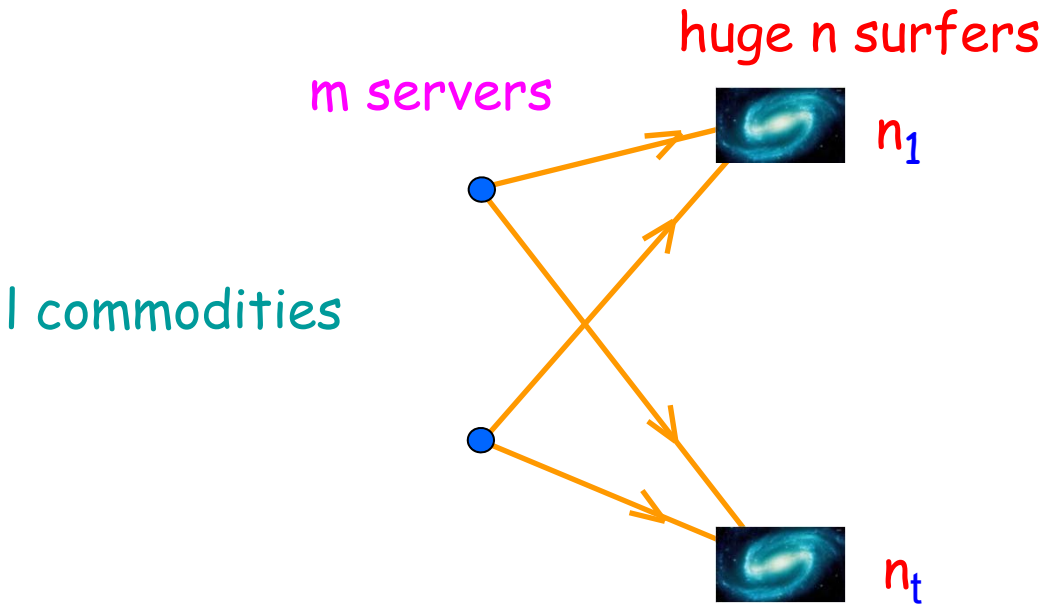
-- fixed-parameter tractable $O(n^3 L)$
(Hemmecke, Onn, Romanchuk)

-- strongly polynomial $O(n^{g(l,m)})$
(De Loera, Hemmecke, Lee)

2017 fixed-parameter tractable and strongly polynomial poly(n)
(Koutecký, Levin, Onn)

Huge Multicommodity Flows

Huge version: surfers come in huge clouds of t types



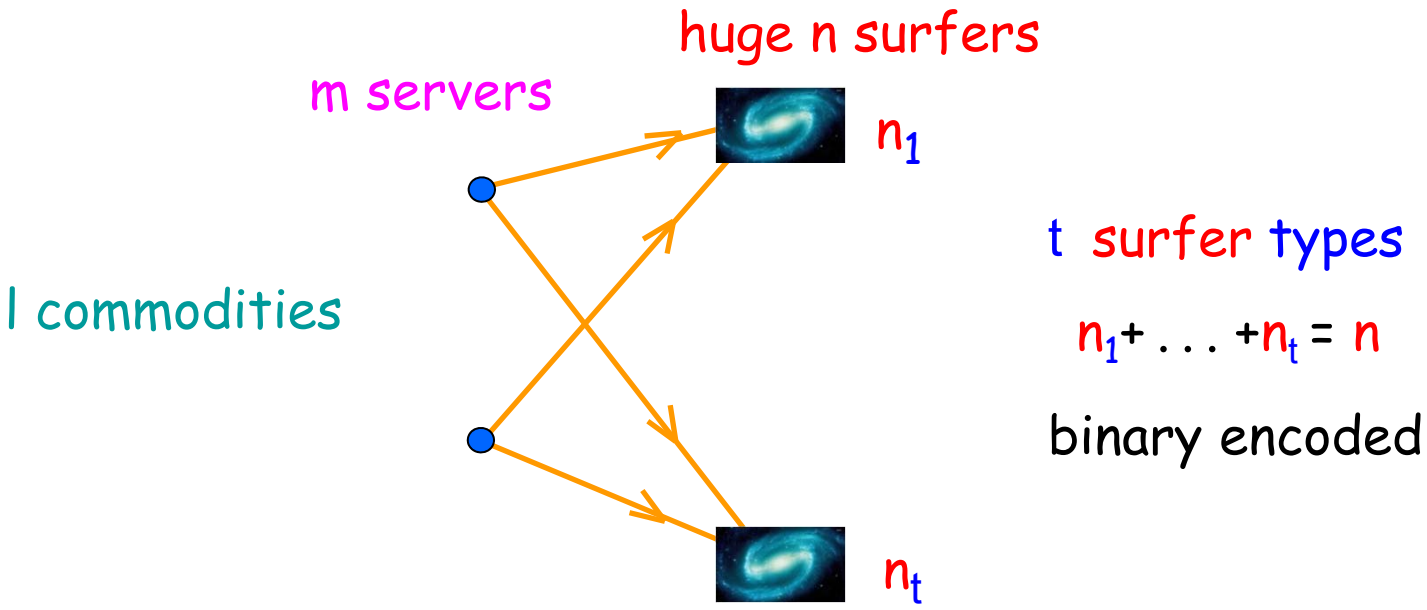
t surfer types

$n_1 + \ldots + n_t = n$

binary encoded

Huge Multicommodity Flows

Huge version: surfers come in huge clouds of t types



t surfer types

$n_1 + \ldots + n_t = n$

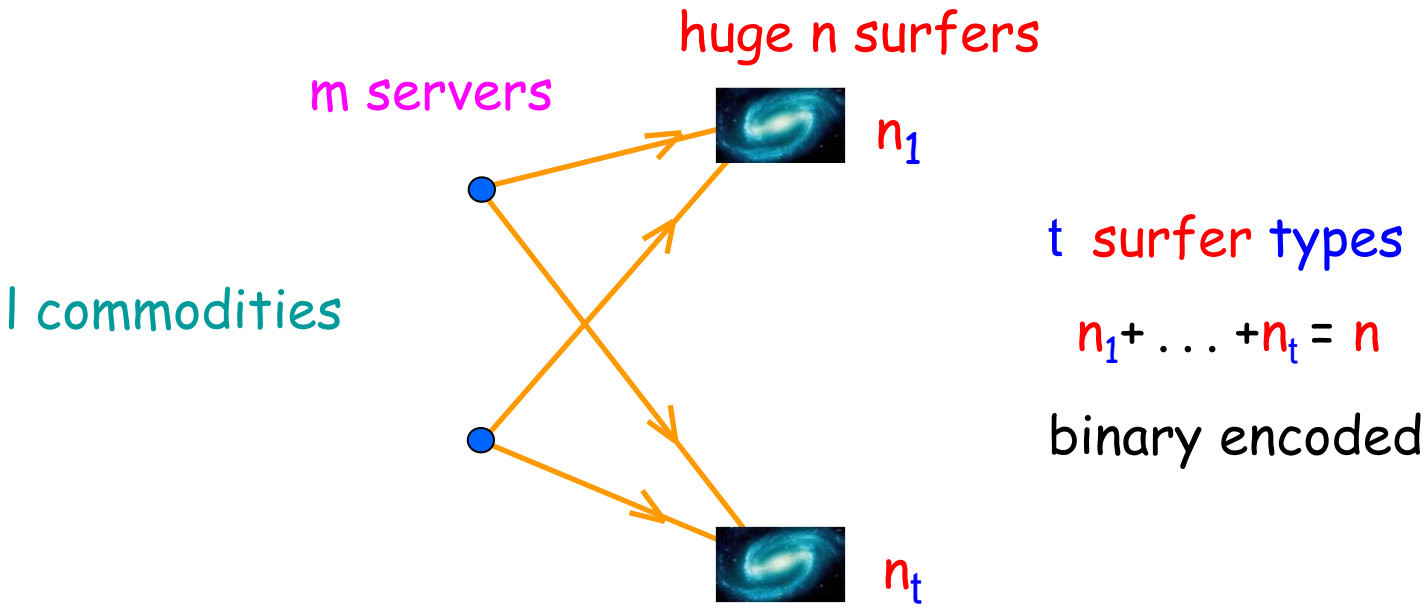
binary encoded

2017 (Onn): fixed-parameter tractable with parameters l,t, variable m, huge n

Shmuel Onn

Huge Multicommodity Flows

Huge version: surfers come in huge clouds of t types



t surfer types

$n_1 + \ldots + n_t = n$

binary encoded

2017 (Onn): fixed-parameter tractable with parameters l,t, variable m, huge n

Similarly, huge $l \times m \times n$ tables with parameters l,m, variable t, huge n, are FPT

More Applications, Extensions, Directions

Combinatorial N-Fold Integer Programming

Theorem (Knop, Koutecký, Mnich, latest ESA): combinatorial n-fold integer programming, that is, with s=1 and A_2=[1, . . ., 1], can be solved in fixed-parameter time depending polynomially instead of exponetially on t.

$(r,s) \times t$ bimatrix $A=\begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$

Martin Koutecký, current postdoc with Asaf Levin and me at Technion

Combinatorial N-Fold Integer Programming

Theorem (Knop, Koutecký, Mnich, latest ESA): combinatorial n-fold integer programming, that is, with s=1 and A_2=[1, . . ., 1], can be solved in fixed-parameter time depending polynomially instead of exponetially on t.

Corollary (Knop, Koutecký, Mnich): Drastic time reduction in dependence on the parameter, down form doubly to singly exponential, in a variety of applications including string matching, set covering problems, scheduling.

4-block N-Fold Integer Programming

Theorem (Hemmecke, Köppe, Weismantel, Math. Prog.):
4-block n-fold integer programming can be solved in polynomial time:

$$\begin{pmatrix} C & D \\ B & A \end{pmatrix}^{(N)} := \begin{pmatrix} C & D & D & \cdots & D \\ B & A & O & & O \\ B & O & A & & O \\ \vdots & & & \ddots & \\ B & O & O & & A \end{pmatrix}$$

4-block N-Fold Integer Programming

Theorem (Hemmecke, Köppe, Weismantel, Math. Prog.):
4-block n-fold integer programming can be solved in polynomial time:

$$\begin{pmatrix} C & D \\ B & A \end{pmatrix}^{(N)} := \begin{pmatrix} C & D & D & \cdots & D \\ B & A & O & & O \\ B & O & A & & O \\ \vdots & & & \ddots & \\ B & O & O & & A \end{pmatrix}$$

Corollary (Hemmecke, Köppe, Weismantel):
Stochastic multicommodity flows can be solved in polynomial time.

4-block N-Fold Integer Programming

Theorem (Hemmecke, Köppe, Weismantel, Math. Prog.):
4-block n-fold integer programming can be solved in polynomial time:

$$\begin{pmatrix} C & D \\ B & A \end{pmatrix}^{(N)} := \begin{pmatrix} C & D & D & \cdots & D \\ B & A & O & & O \\ B & O & A & & O \\ \vdots & & & \ddots & \\ B & O & O & & A \end{pmatrix}$$

Corollary (Hemmecke, Köppe, Weismantel):
Stochastic multicommodity flows can be solved in polynomial time.

Open: Is 4-block n-fold IP fixed-parameter tractable ?

Tree N-Fold Integer Programming

Theorem (Chen, Marx, coming SODA): Tree n-fold integer programming parameterized by s_i, t, max A_i, is solvable in fixed-parameter time $O(n^3 L)$:

$$A = \begin{bmatrix}
A_1 & A_1 & A_1 & A_1 & A_1 & A_1 & A_1 & A_1 & A_1 & A_1 & A_1 & A_1 \\
A_2 & A_2 & A_2 & A_2 & A_2 & A_2 & A_2 & A_2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_2 & A_2 & A_2 & A_2 \\
A_3 & A_3 & A_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & A_3 & A_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & A_3 & A_3 & A_3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_3 & A_3 & A_3 & A_3 \\
A_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & A_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & A_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & A_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & A_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & A_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & A_4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & A_4 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_4 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_4 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_4
\end{bmatrix}$$

Tree N-Fold Integer Programming

Theorem (Chen, Marx, coming SODA): Tree n-fold integer programming parameterized by s_i, t, max A_i, is solvable in fixed-parameter time $O(n^3 L)$:

Corollary (Chen, Marx): Covering a rooted tree with subtrees parameterized by the makespan is fixed-parameter tractable.

Some Further Developments in Theory and Applications

- Clustering and farmland consolidation, Borgwardt, Melamed, Onn
- Scheduling, Knop, Koutecký
- Stochastic integer programming, Hemmecke, Onn, Weismantel
- Portfolio optimization, Baumann, Trautmann
- Optimality certificates, Kobayashi, Murota, Saito, Weismantel
- Production scheduling, Andziulis, Dzemydien
- IP with few global variables & constraints, Dvořák, Eiben, Ganian, Knop, Ordyniak
- Matrix apportionment problems, Gaffke, Pukelsheim
- Strongly polynomial algorithms, De Loera, Hemmecke, Lee
- Rounding in integer nonlinear optimization, Hubner, Schobel
- Graver complexity, Berstein, Finhold, Hemmecke, Kudo, Nairn, Onn, Takemura
- Network design problems, Guyard, Laugier
- Games, Hemmecke, Nguyen, Onn, Ryan, Weismantel

Some Directions

More applications of n-fold integer programming and multiway tables
to parameterized complexity and approximation algorithms

Some Directions

More applications of n-fold integer programming and multiway tables to parameterized complexity and approximation algorithms

Further development of n-fold integer programming theory

Some Directions

More applications of n-fold integer programming and multiway tables to parameterized complexity and approximation algorithms

Further development of n-fold integer programming theory

Complexity of n-fold IP with r,s,t parameters but A unary/binary input?

Some Directions

More applications of n-fold integer programming and multiway tables to parameterized complexity and approximation algorithms

Further development of n-fold integer programming theory

Complexity of n-fold IP with r,s,t parameters but A unary/binary input?

Faster algorithm for tables and in particular $3 \times 3 \times n$ tables ?

Some Directions

More applications of n-fold integer programming and multiway tables to parameterized complexity and approximation algorithms

Further development of n-fold integer programming theory

Complexity of n-fold IP with r,s,t parameters but A unary/binary input?

Faster algorithm for tables and in particular $3 \times 3 \times n$ tables ?

Is the Graver complexity of $3 \times m \times n$ tables $g(m) = 3^{m-1}$?
What about $g(m_1, \ldots, m_k)$ for $m_1 \times \cdots \times m_k \times n$ tables ?

Some Directions

More applications of n-fold integer programming and multiway tables to parameterized complexity and approximation algorithms

Further development of n-fold integer programming theory

Complexity of n-fold IP with r,s,t parameters but A unary/binary input?

Faster algorithm for tables and in particular $3 \times 3 \times n$ tables ?

Is the Graver complexity of $3 \times m \times n$ tables $g(m) = 3^{m-1}$?
What about $g(m_1, \ldots, m_k)$ for $m_1 \times \cdots \times m_k \times n$ tables ?

Is 4-block n-fold IP fixed-parameter tractable ?

Some Directions

More applications of n-fold integer programming and multiway tables to parameterized complexity and approximation algorithms

Further development of n-fold integer programming theory

Complexity of n-fold IP with r,s,t parameters but A unary/binary input?

Faster algorithm for tables and in particular $3 \times 3 \times n$ tables ?

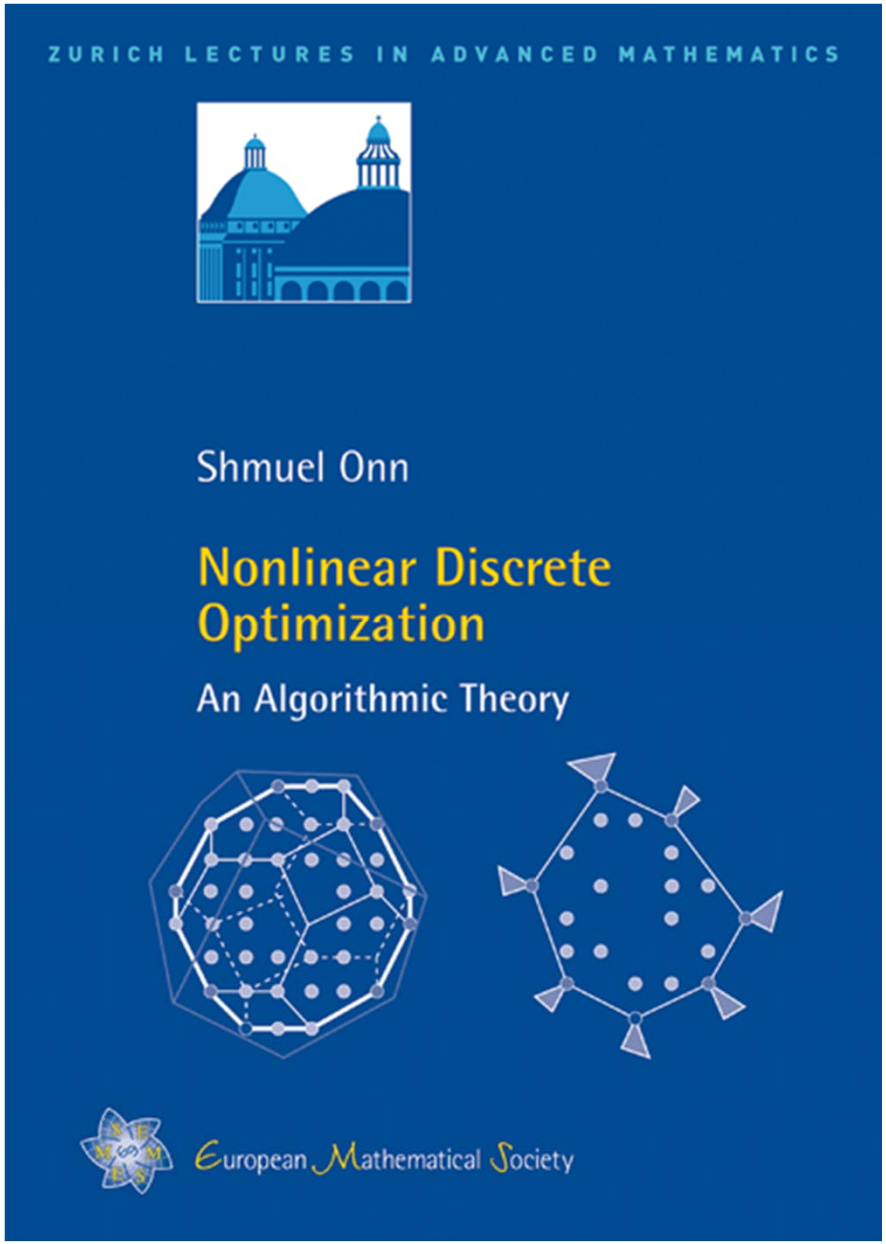
Is the Graver complexity of $3 \times m \times n$ tables $g(m) = 3^{m-1}$?
What about $g(m_1, \ldots, m_k)$ for $m_1 \times \cdots \times m_k \times n$ tables ?

Is 4-block n-fold IP fixed-parameter tractable ?

Complexity of Graver bases detection and output sensitive computation?

Bibliography (mostly available at http://ie.technion.ac.il/~onn)

- The complexity of 3-way tables (SIAM J. Comp.)
- Convex combinatorial optimization (Disc. Comp. Geom.)
- Markov bases of 3-way tables (J. Symb. Comp.)
- All linear and integer programs are slim 3-way programs (SIAM J. Opt.)
- Graver complexity of integer programming (Annals Combin.)
- N-fold integer programming (Disc. Opt. in memory of Dantzig)
- Convex integer maximization via Graver bases (J. Pure App. Algebra)
- Polynomial oracle-time convex integer minimization (Math. Prog.)
- Theory and applications of n-fold integer programming (IMA Volume on MINLP)
- Quadratic Graver cones, quadratic integer minimization & extensions (Math. Prog.)
- Robust integer programming (Operations Research Letters)
- N-fold integer programming in cubic time (Math. Prog.)
- Huge tables and multicommodity flows are fixed-parameter tractable via unimodular integer Caratheodory (J. Computer and System Sciences)



Background in my Book:

Theory of Graver bases for integer programming

(and more)

Available electronically from my homepage

(with kind permission of EMS)