Nonlinear Discrete Optimization

Shmuel Onn

Technion – Israel Institute of Technology

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

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

Outline

0. The Setup

1. Convex Discrete Maximization

2. Nonlinear Integer Programming

3. Nonlinear Combinatorial Optimization

Setup for Nonlinear Discrete Optimization

The **problem** is:

$$\min/\max \{ f(Wx) : x \text{ in } S \}$$

Shmuel Onn

Setup for Nonlinear Discrete Optimization

The **problem** is: min/max { $f(Wx)$: x in S }

The data is a triple:

Setup for Nonlinear Discrete Optimization

The **problem** is:

$$\min/\max \{ f(Wx) : x \text{ in } S \}$$

The data is a triple:

S: Set of feasible points in Z^n given by either

Shmuel Onn

Setup for Nonlinear Discrete Optimization

The **problem** is:

$$\min/\max \{ f(Wx) : x \text{ in } S \}$$

The data is a triple:

S: Set of feasible points in Z^n given by either

- inequalities $S = \{x \text{ in } Z^n : A(x) \leq 0 \}$ (integer programming)

Setup for Nonlinear Discrete Optimization

The **problem** is:

$$\min/\max \{ f(Wx) : x \text{ in } S \}$$

The data is a triple:

S: Set of feasible points in Z^n given by either

- inequalities $S = \{x \text{ in } Z^n : A(x) \leq 0 \}$ (integer programming)
- oracle (membership, linear-optimization) (combinatorial optimization)

Setup for Nonlinear Discrete Optimization

The **problem** is: $\min/\max \{ f(Wx) : x \text{ in } S \}$

The data is a triple:

S: Set of feasible points in Z^n given by either

- inequalities $S = \{x \text{ in } Z^n : A(x) \leq 0 \}$ (integer programming)
- oracle (membership, linear-optimization) (combinatorial optimization)

W: Integer $d \times n$ matrix

Setup for Nonlinear Discrete Optimization

The **problem** is: min/max { $f(Wx)$: x in S }

The data is a triple:

S: Set of feasible points in Z^n given by either

- inequalities $S = \{x \text{ in } Z^n : A(x) \leq 0\}$ (integer programming)
- oracle (membership, linear-optimization) (combinatorial optimization)

W: Integer $d \times n$ matrix

f: Function from R^d to R, often given by comparison oracle

Shmuel Onn

Setup for Nonlinear Discrete Optimization

The **problem** is:

$$\min/\max \{ f(Wx) : x \text{ in } S \}$$

The data is a triple:

S: Set of feasible points in Z^n given by either

- inequalities $S = \{x \text{ in } Z^n : A(x) \le 0 \}$ (integer programming)
- oracle (membership, linear-optimization) (combinatorial optimization)

W: Integer $d \times n$ matrix

f: Function from R^d to R, often given by comparison oracle

This setup enables:

Shmuel Onn

Setup for Nonlinear Discrete Optimization

The **problem** is:

$$\min/\max \{ f(Wx) : x \text{ in } S \}$$

The data is a triple:

S: Set of feasible points in Z^n given by either

- inequalities $S = \{x \text{ in } Z^n : A(x) \leq 0 \}$ (integer programming)
- oracle (membership, linear-optimization) (combinatorial optimization)

W: Integer $d \times n$ matrix

f: Function from R^d to R, often given by comparison oracle

This setup enables:

Determination of broad classes of triples **S**, **W**, **f** where we can solve efficiently (deterministically, randomly, or approximately)

Shmuel Onn

Setup for Nonlinear Discrete Optimization

The **problem** is: $\min/\max \{ f(Wx) : x \text{ in } S \}$

The data is a triple:

S: Set of feasible points in Z^n given by either

- inequalities $S = \{x \text{ in } Z^n : A(x) \leq 0 \}$ (integer programming)
- oracle (membership, linear-optimization) (combinatorial optimization)

W: Integer $d \times n$ matrix

f: Function from R^d to R, often given by comparison oracle

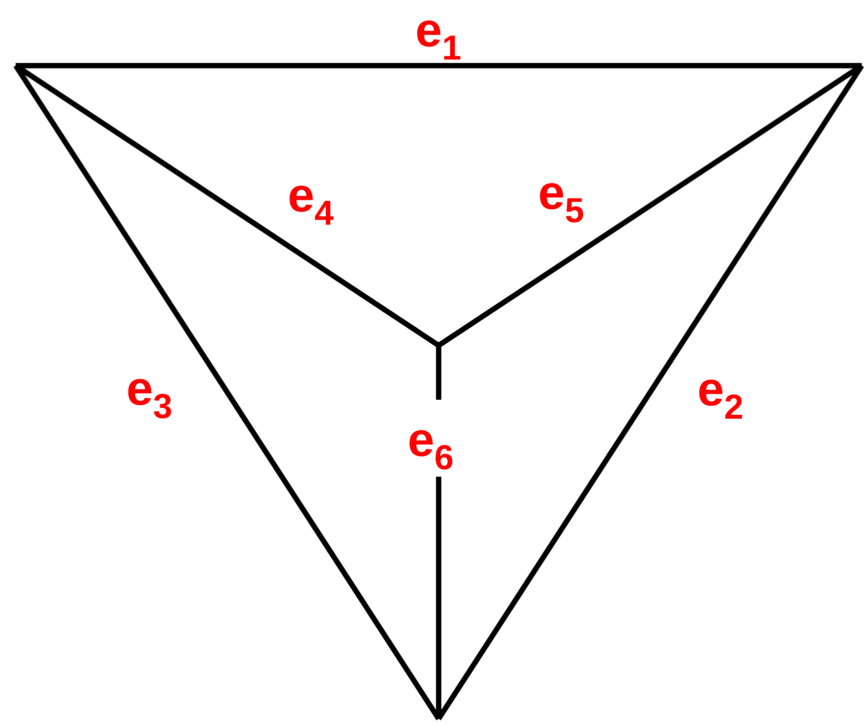
This setup enables:

Interpretation as multi-objective optimization with objective $f(Wx) = f(W_1x, \ldots, W_dx)$ balancing criteria W_ix of d players

Small Example

Small Example

S in $\{0,1\}^6$ consists of all spanning trees in the graph K_4

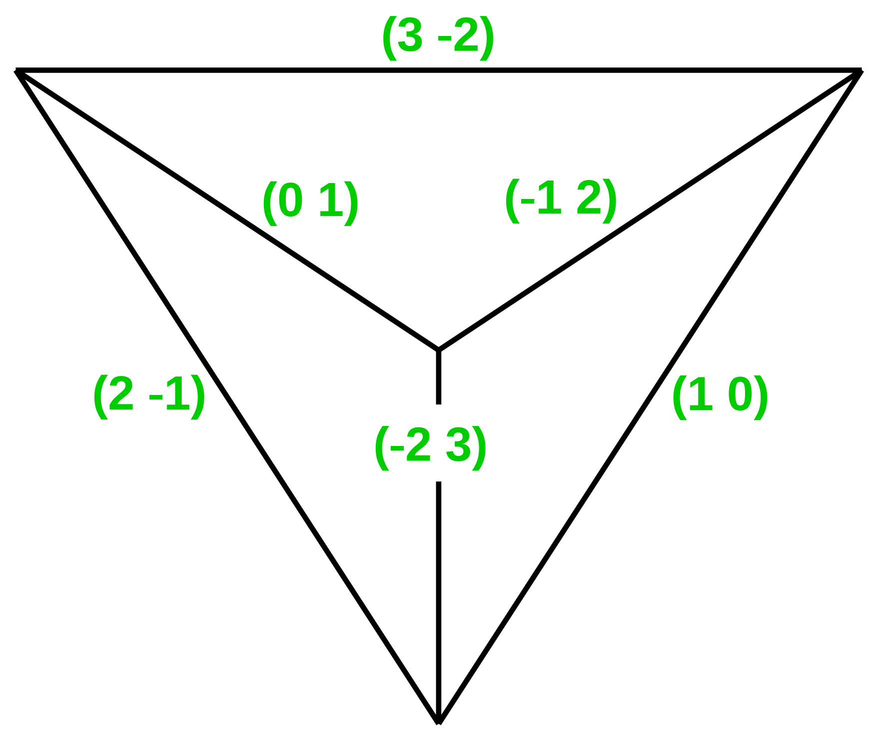


Small Example

S in $\{0,1\}^6$ consists of all spanning trees in the graph K_4

W =

3	1	2	0	-1	-2	criterion/player 1
-2	0	-1	1	2	3	criterion/player 2



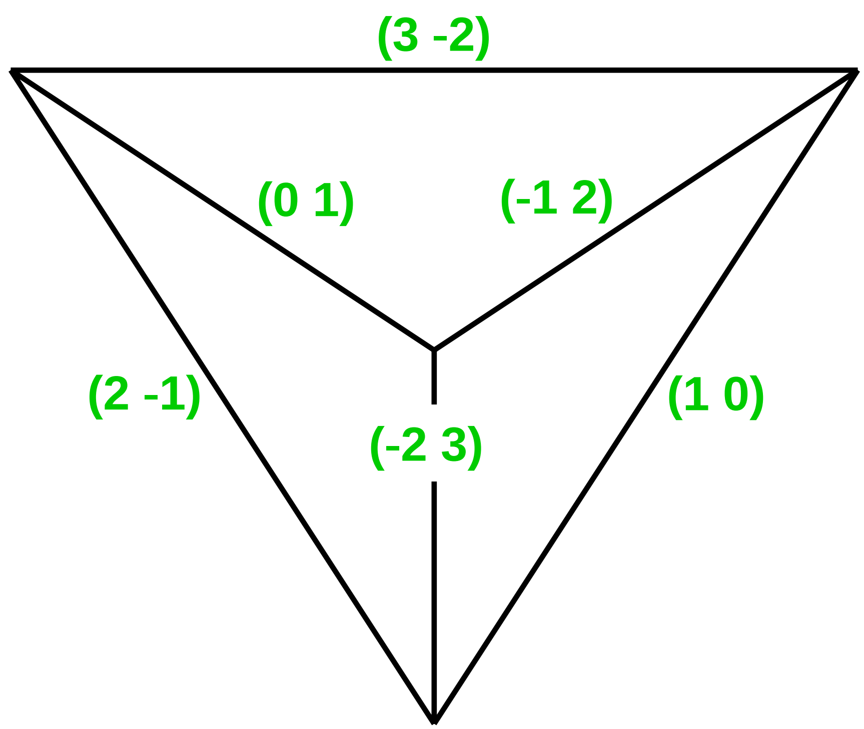
Small Example

S in $\{0,1\}^6$ consists of all spanning trees in the graph K_4

W =

3	1	2	0	-1	-2	criterion/player 1
-2	0	-1	1	2	3	criterion/player 2

f is Euclidean norm $f(y) = |y|^2 = y_1^2 + y_2^2$ balancing criteria

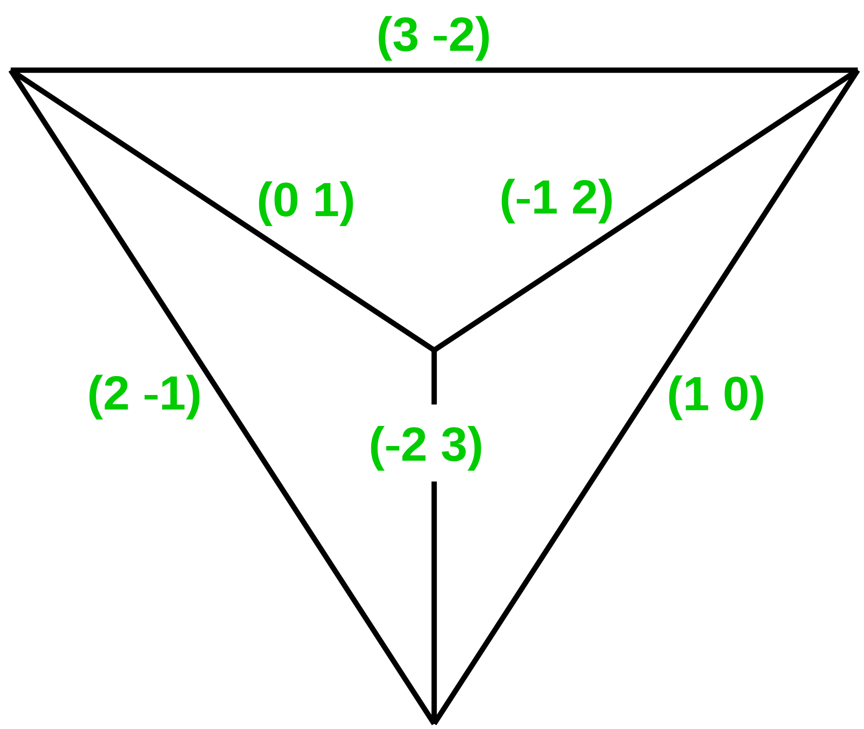


Small Example

S in $\{0,1\}^6$ consists of all spanning trees in the graph K_4

$$W = \begin{bmatrix} 3 & 1 & 2 & 0 & -1 & -2 \\ -2 & 0 & -1 & 1 & 2 & 3 \end{bmatrix}$$

f is Euclidean norm $f(y) = |y|^2 = y_1^2 + y_2^2$



The nonlinear problem is:

$\max \{ f(Wx) : x \text{ in } \{0,1\}^6 \text{ spaning tree} \}$

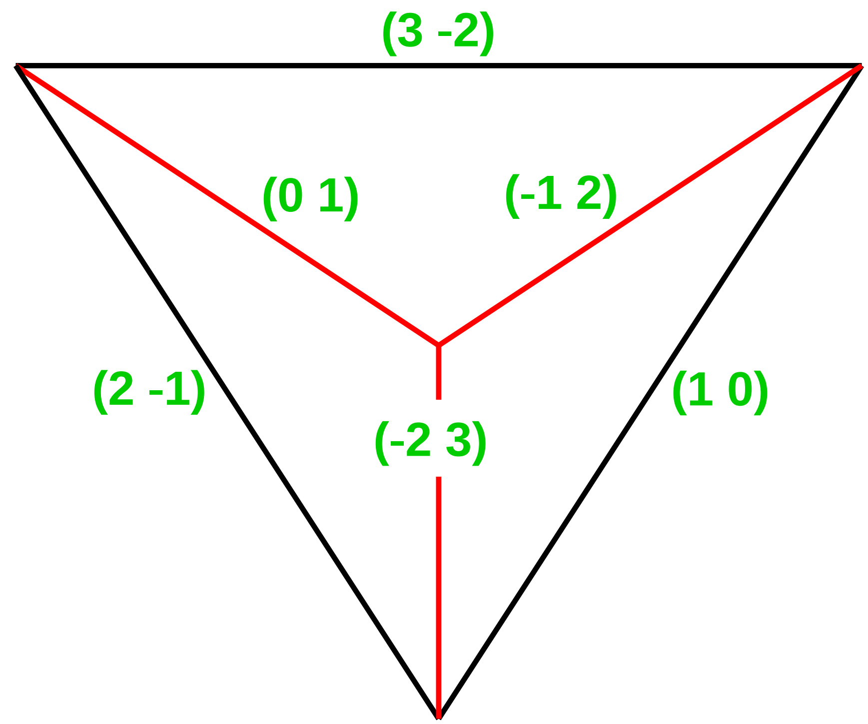
Small Example

S in $\{0,1\}^6$ consists of all spanning trees in the graph K_4

W =

3	1	2	0	-1	-2
-2	0	-1	1	2	3

f is Euclidean norm $f(y) = |y|^2 = y_1^2 + y_2^2$



The nonlinear problem is:

$\max\{ f(Wx) : x \text{ in } \{0,1\}^6 \text{ spaning tree} \}$

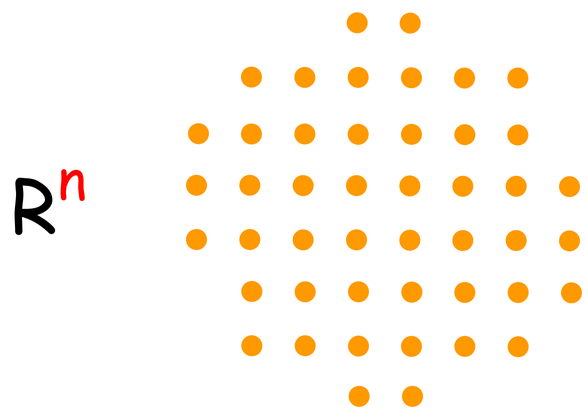
The optimal tree is $x = (0\ 0\ 0\ 1\ 1\ 1)$

with $Wx = (-3\ \ 6)$ and $f(Wx) = 45$

Shmuel Onn

Some Possible Assumptions on the Data S,W,f

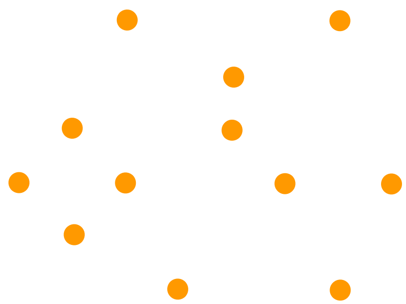
Some Possible Assumptions on the Data S,W,f



- Feasible set S in Z^n presented by

Some Possible Assumptions on the Data S,W,f

R^n

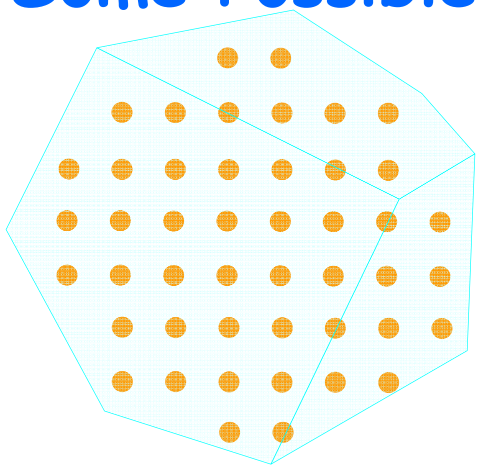


- Feasible set S in Z^n presented by
  - membership oracle: very little information

Shmuel Onn

Some Possible Assumptions on the Data S,W,f

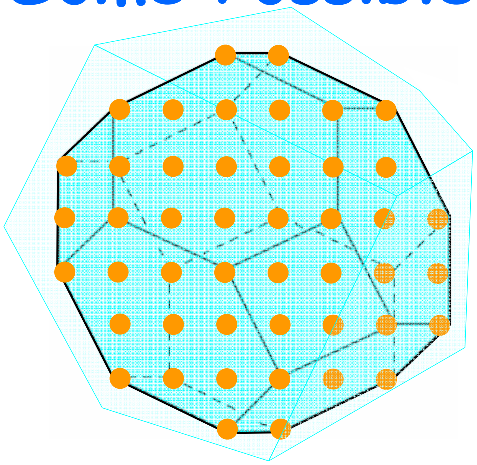
R^n



- Feasible set S in Z^n presented by
  - membership oracle: very little information
  - inequalities: $S = \{x \text{ in } Z^n : A(x) \leq 0\}$

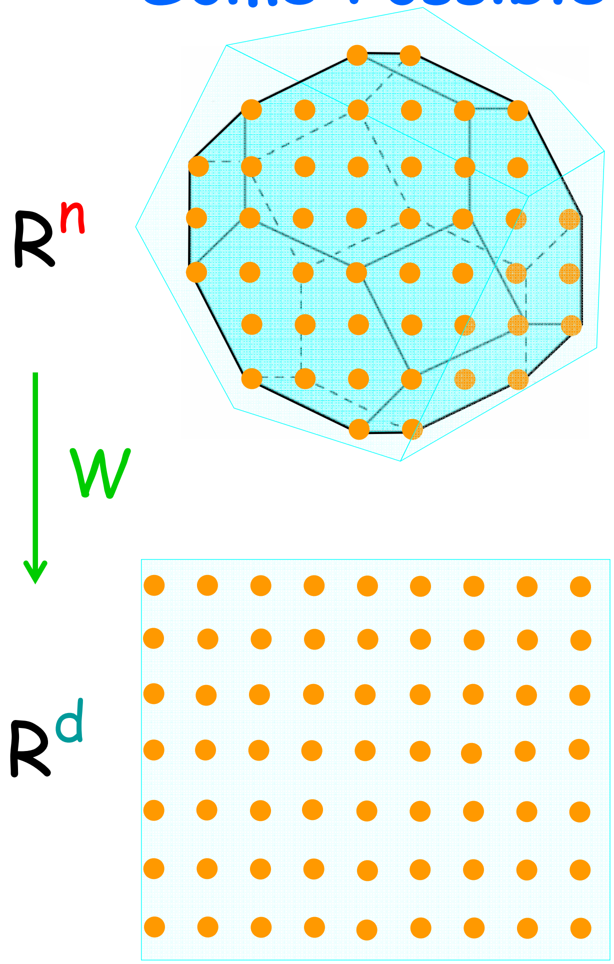
Some Possible Assumptions on the Data S,W,f

R^n



- Feasible set S in Z^n presented by
  - membership oracle: very little information
  - inequalities: $S = \{x \text{ in } Z^n : A(x) \leq 0\}$
  - linear-optimization oracle: access to conv{S}

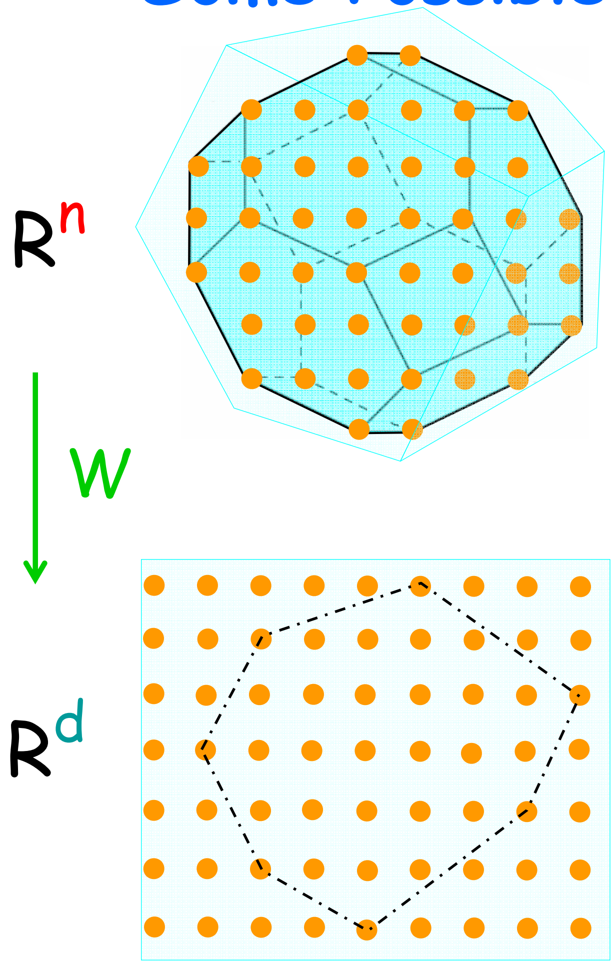
Some Possible Assumptions on the Data S,W,f



- Feasible set S in Z^n presented by
  - membership oracle: very little information
  - inequalities: $S = \{x \text{ in } Z^n : A(x) \leq 0\}$
  - linear-optimization oracle: access to conv{S}

- Integer $d \times n$ matrix W

Some Possible Assumptions on the Data S,W,f

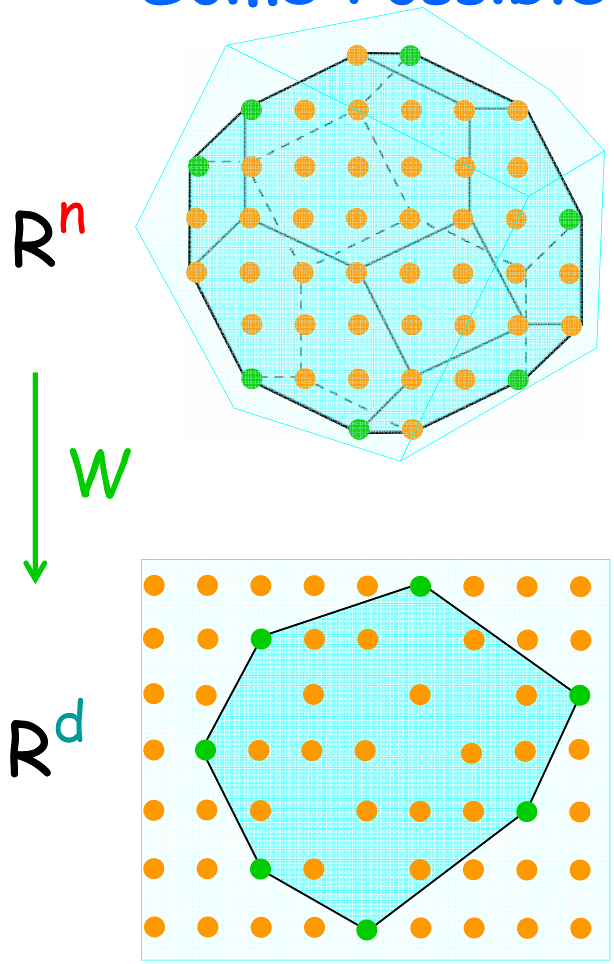


- Feasible set S in Z^n presented by
  - membership oracle: very little information
  - inequalities: $S = \{x \text{ in } Z^n : A(x) \leq 0\}$
  - linear-optimization oracle: access to conv{S}

- Integer d x n matrix W
  - W binary encoded: cannot even tell conv{WS}

Shmuel Onn

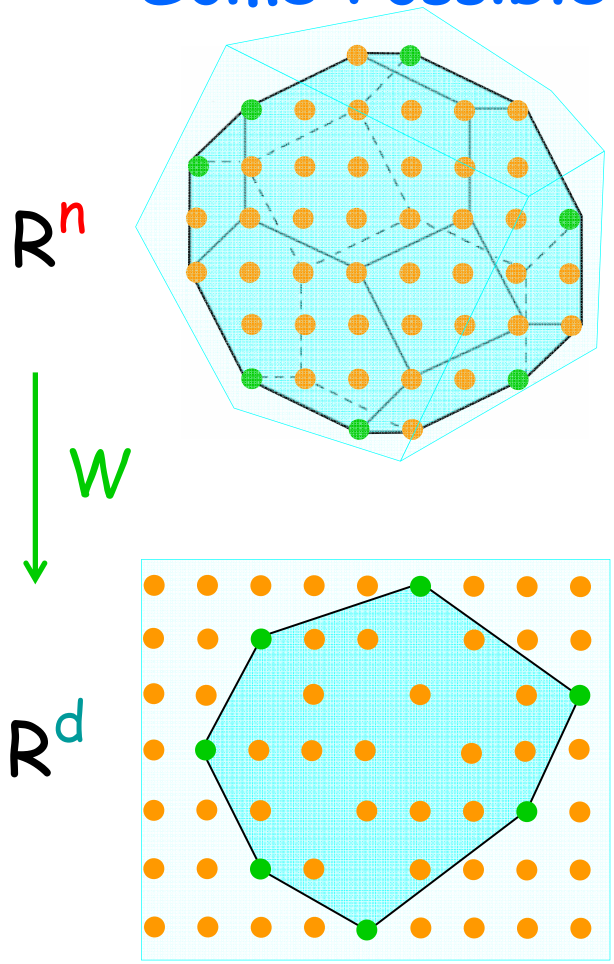
Some Possible Assumptions on the Data S,W,f



- Feasible set S in Z^n presented by
  - membership oracle: very little information
  - inequalities: $S = \{x \text{ in } Z^n : A(x) \leq 0\}$
  - linear-optimization oracle: access to conv{S}

- Integer $d \times n$ matrix W
  - W binary encoded: cannot even tell conv{WS}
  - W unary encoded: may tell conv{WS} but not holes

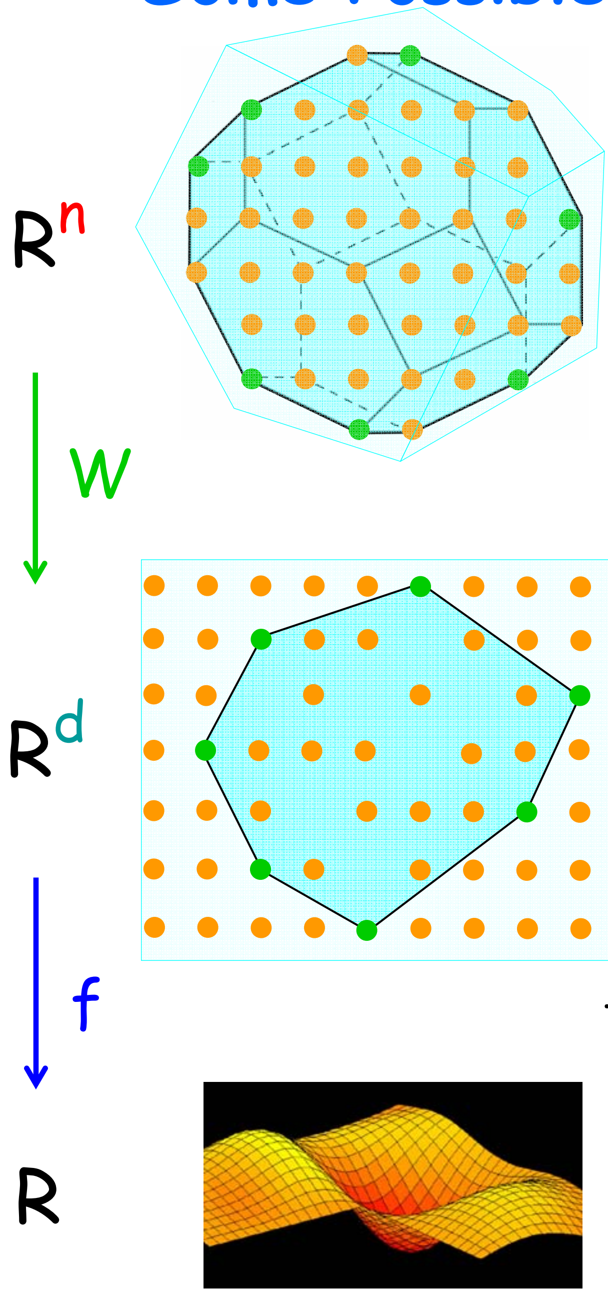
Some Possible Assumptions on the Data S,W,f



- Feasible set S in Z^n presented by
  - membership oracle: very little information
  - inequalities: $S = \{x \text{ in } Z^n : A(x) \leq 0\}$
  - linear-optimization oracle: access to conv{S}

- Integer d x n matrix W
  - W binary encoded: cannot even tell conv{WS}
  - W unary encoded: may tell conv{WS} but not holes
  - W takes fixed values $a_1,\ldots,a_p$: may tell holes partly

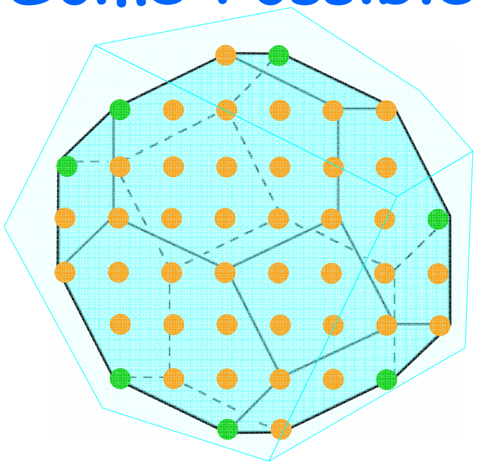
Some Possible Assumptions on the Data S,W,f



- Feasible set S in Z^n presented by
  - membership oracle: very little information
  - inequalities: $S = \{x \text{ in } Z^n : A(x) \leq 0\}$
  - linear-optimization oracle: access to conv{S}
- Integer d x n matrix W
  - W binary encoded: cannot even tell conv{WS}
  - W unary encoded: may tell conv{WS} but not holes
  - W takes fixed values $a_1,\dots,a_p$: may tell holes partly
- Function f on R^d presented by comparison oracle

Some Possible Assumptions on the Data S,W,f

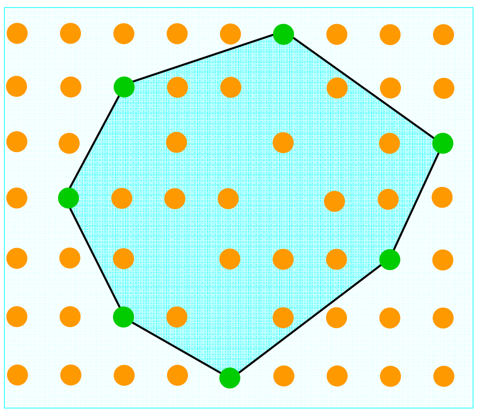
R^n



- Feasible set S in Z^n presented by
  - membership oracle: very little information
  - inequalities: $S = \{x \text{ in } Z^n : A(x) \le 0\}$
  - linear-optimization oracle: access to conv{S}

W

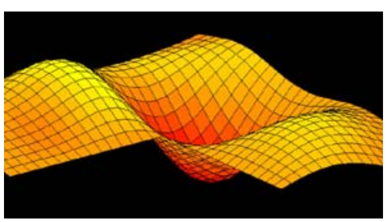
R^d



- Integer $d \times n$ matrix W
  - W binary encoded: cannot even tell conv{WS}
  - W unary encoded: may tell conv{WS} but not holes
  - W takes fixed values $a_1,\ldots,a_p$: may tell holes partly

f

R

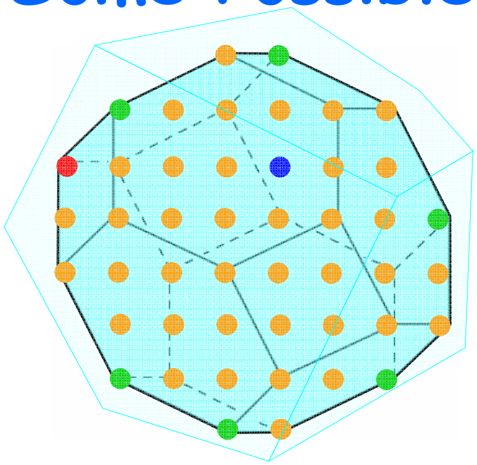


- Function f on R^d presented by comparison oracle
  - arbitrary nonlinear
  - separable
  - convex

Shmuel Onn

Some Possible Assumptions on the Data S,W,f

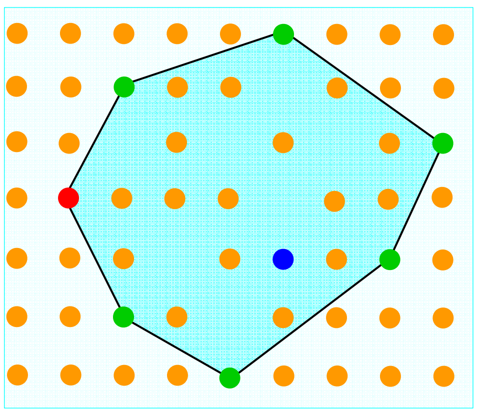
R^n



- Feasible set S in Z^n presented by
  - membership oracle: very little information
  - inequalities: $S = \{x \text{ in } Z^n : A(x) \le 0\}$
  - linear-optimization oracle: access to conv{S}

$\downarrow W$

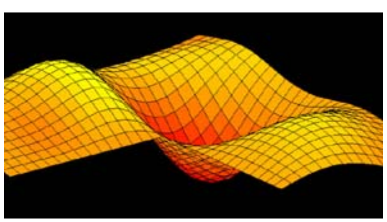
R^d



- Integer d x n matrix W
  - W binary encoded: cannot even tell conv{WS}
  - W unary encoded: may tell conv{WS} but not holes
  - W takes fixed values $a_1,\ldots,a_p$: may tell holes partly

$\downarrow f$

R



- Function f on R^d presented by comparison oracle
  - arbitrary nonlinear
  - separable
  - convex

min/max

Shmuel Onn

1. Convex Discrete Maximization

Data:

- S given by linear-optimization oracle and edge-behaved
- W binary encoded
- f convex to be maximized given by comparison oracle

1. Convert Discrete Maximization

Data:

- S given by linear-optimization oracle and edge-behaved
- W binary encoded
- f convex to be maximized given by comparison oracle

Methods: **mostly geometric - zonotopes**

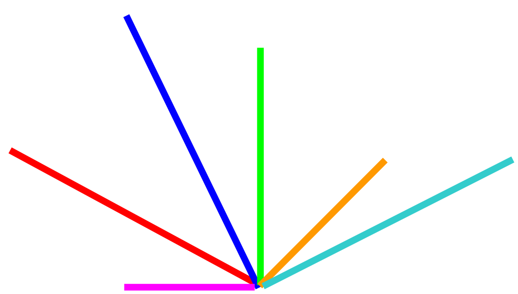
References: **Onn, Rothblum, Schulman**

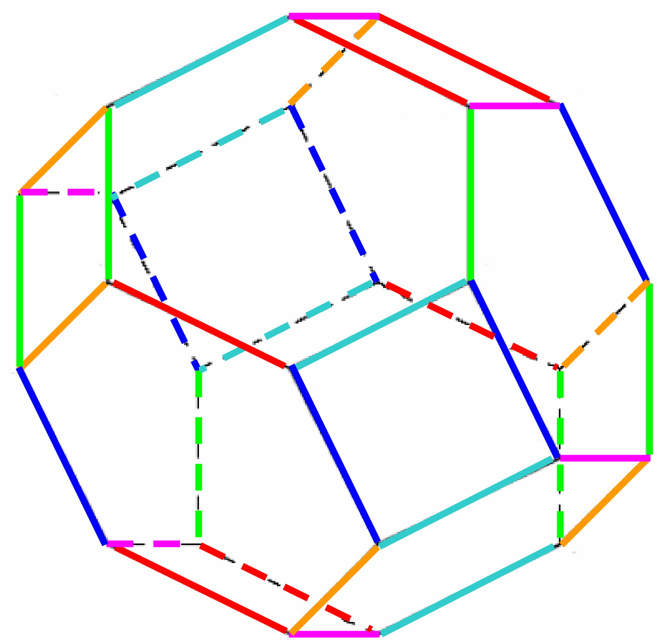
- Partition problems with convex objectives (Math. OR)
- Convex matroid optimization (SIAM Disc. Math.)
- Convex combinatorial optimization (Disc. Comp. Geom.)

Shmuel Onn

Convex Discrete Maximization

If we have a set E covering all edge-directions of conv{S}





then we can reduce convex maximization over S to
polynomially many linear optimization counterparts.

Shmuel Onn

Convex Discrete Maximization

Theorem 1.1: Fix any d. Then can solve in strongly polynomial-time

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

for S in Z^n given by linear-optimization oracle and endowed with set E covering edge-directions of conv$\{S\}$, any W, any convex f.

Convex Combinatorial Maximization:

membership oracle suffices

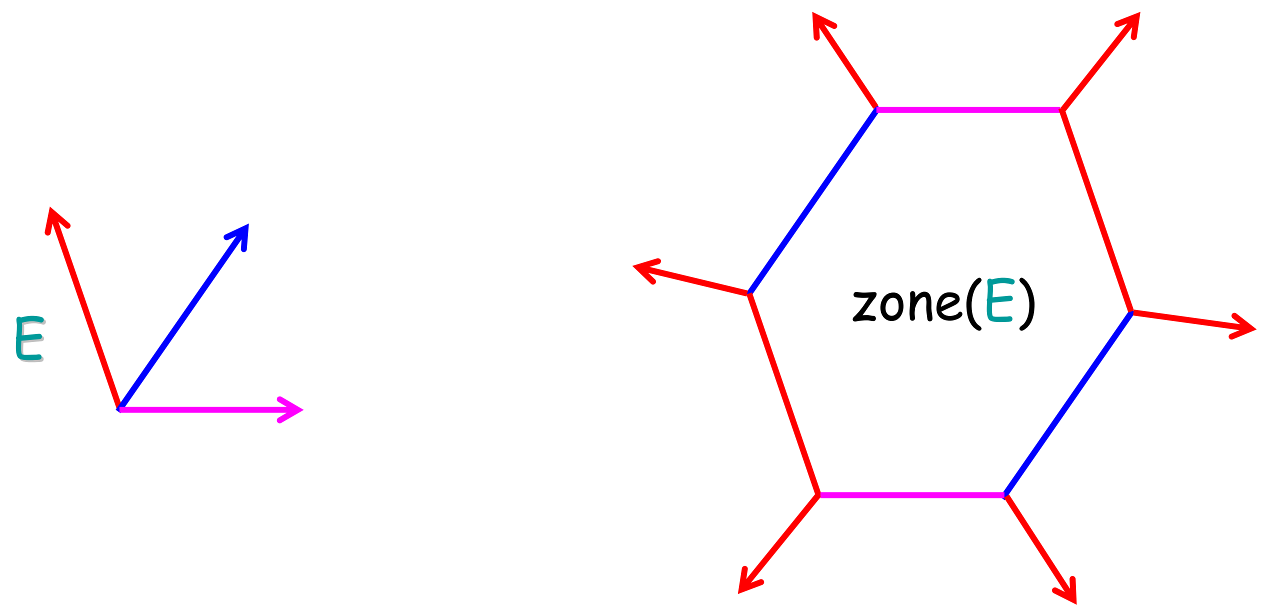
Theorem 1.2: Fix any d. Then can solve in strongly polynomial-time

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

for S in $\{0,1\}^n$ given by membership oracle and endowed with set E covering edge-directions of conv$\{S\}$, any W, any convex f.

Preliminaries on Zonotopes

The zonotope of E is the convex hull zone(E) of all subset sums of E



Lemma: For fixed d can compute zone(E) and normals in polynomial-time

(Edelsbrunner, Gritzmann, Orourk, Seidel, Sharir)

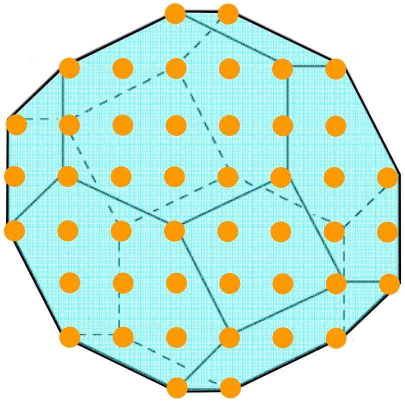
Shmuel Onn

Proof of Theorem 1.1: Convex Discrete Maximization

Shmuel Onn

Proof of Theorem 1.1: Convex Discrete Maximization

R^n

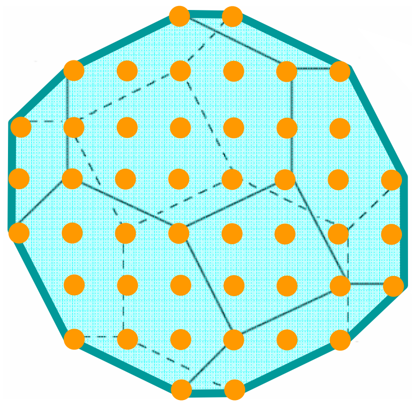


- Set S in Z^n given by linear-optimization oracle

Shmuel Onn

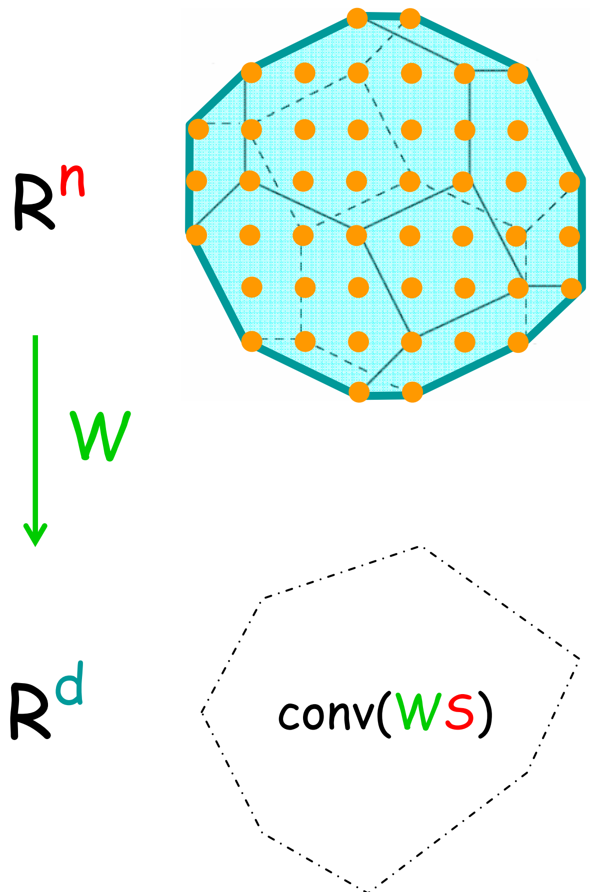
Proof of Theorem 1.1: Convex Discrete Maximization

R^n



- Set S in Z^n given by linear-optimization oracle
    - set E covering edge-directions of conv(S)

Proof of Theorem 1.1: Convex Discrete Maximization

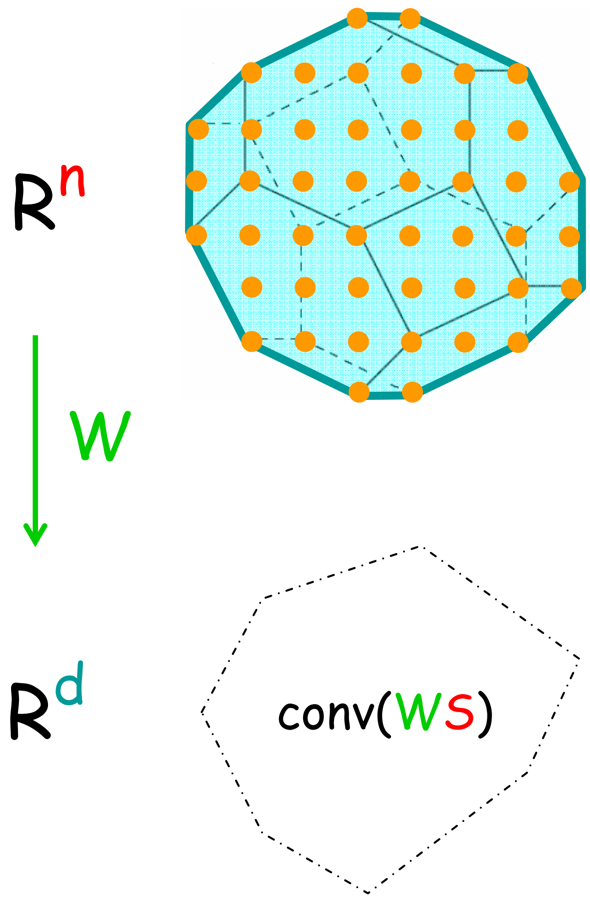


- Set S in Z^n given by linear-optimization oracle
    - set E covering edge-directions of conv(S)

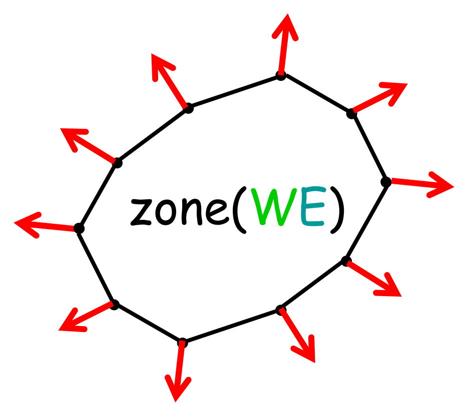
- Binary encoded $d \times n$ matrix W

Proof of Theorem 1.1: Convex Discrete Maximization

- Set S in Z^n given by linear-optimization oracle
    - set E covering edge-directions of conv(S)
- Binary encoded d x n matrix W



Construct zone(WE)

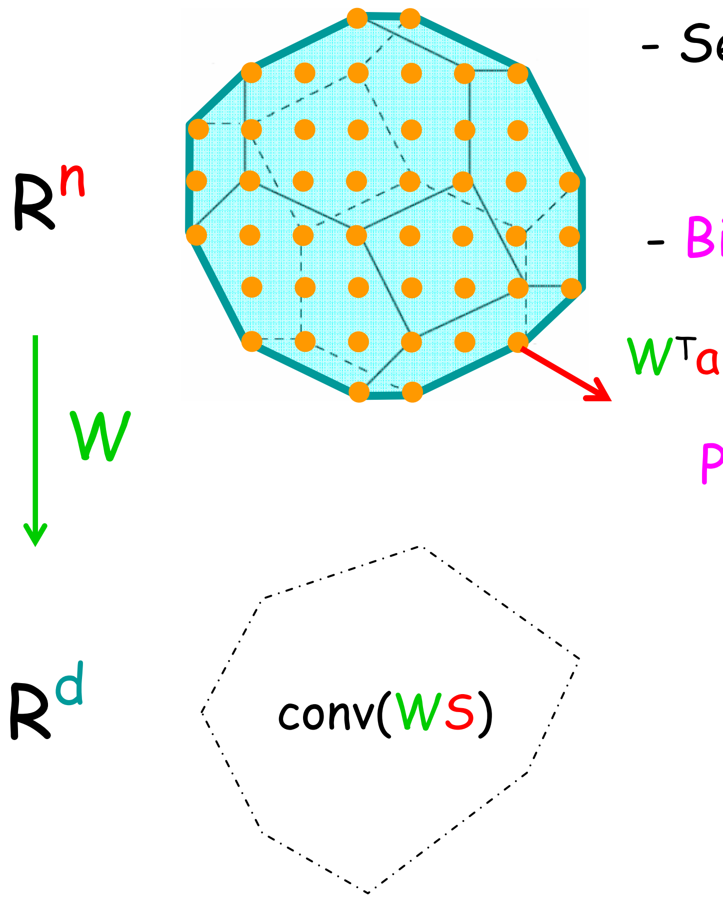


Shmuel Onn

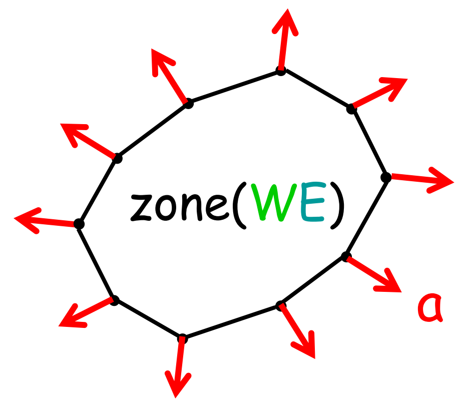
Proof of Theorem 1.1: Convex Discrete Maximization

- Set S in Z^n given by linear-optimization oracle
  - set E covering edge-directions of conv(S)

- Binary encoded $d \times n$ matrix W



Pick a and Optimize W^Ta over S

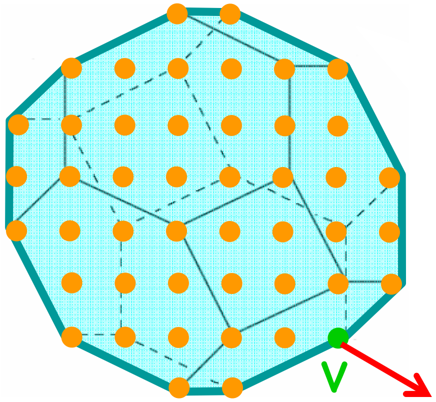


Shmuel Onn

Proof of Theorem 1.1: Convex Discrete Maximization

- Set S in Z^n given by linear-optimization oracle
    - set E covering edge-directions of conv(S)
- Binary encoded $d \times n$ matrix W

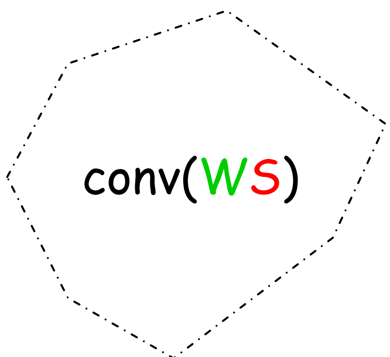
R^n

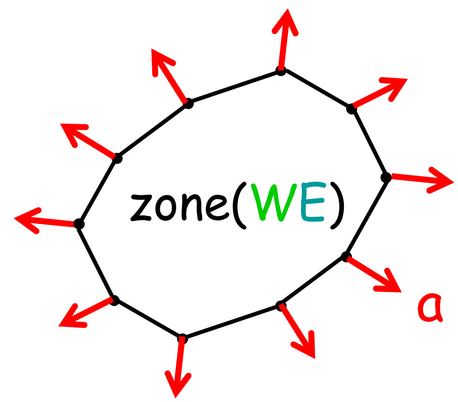


Get vertex v of conv(S)

W

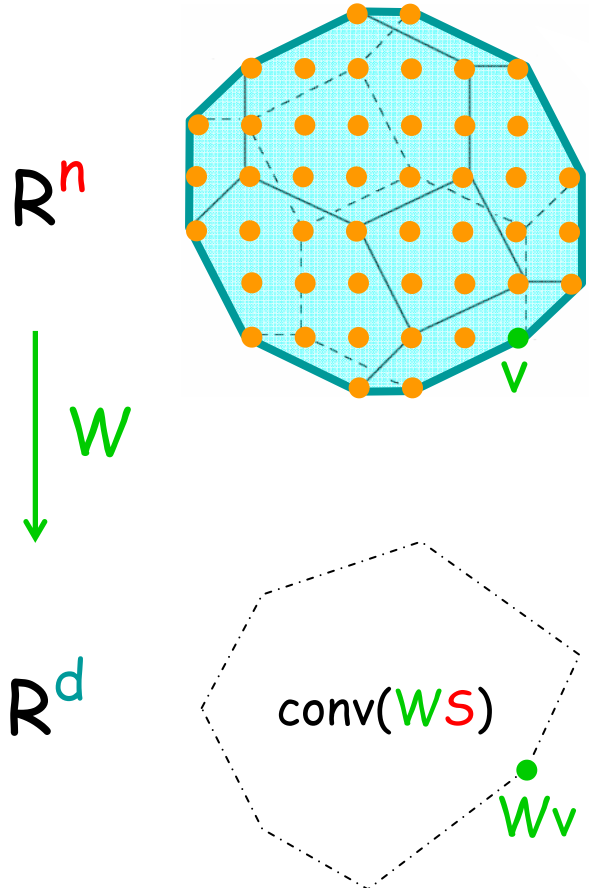
R^d

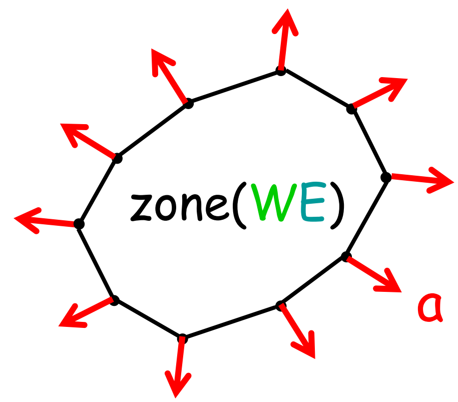




Proof of Theorem 1.1: Convex Discrete Maximization

- Set S in Z^n given by linear-optimization oracle
    - set E covering edge-directions of conv(S)
- Binary encoded $d \times n$ matrix W



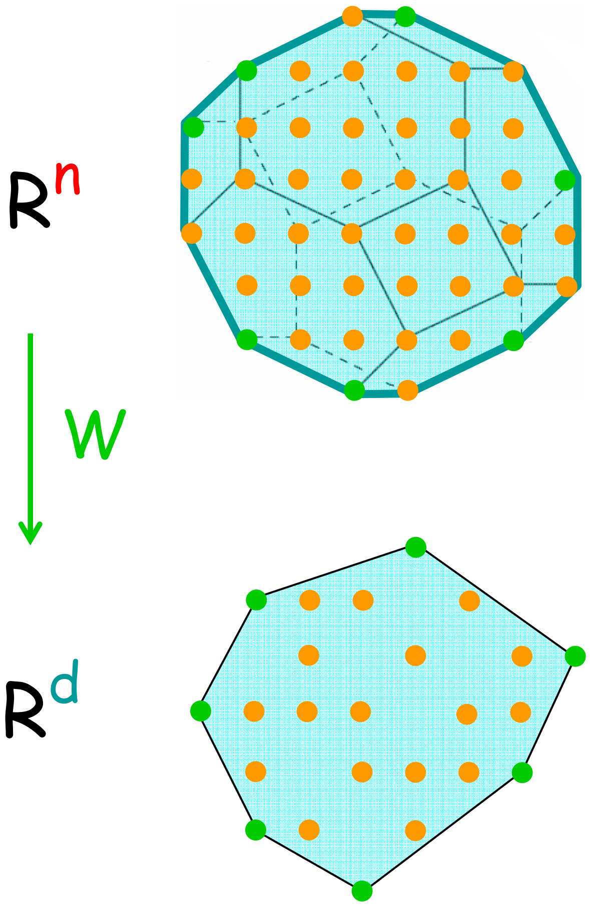


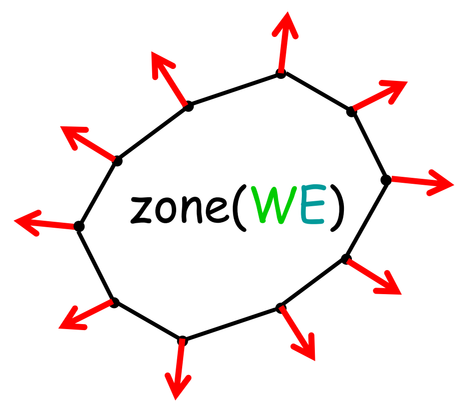
Get point Wv in conv(WS)

Shmuel Onn

Proof of Theorem 1.1: Convex Discrete Maximization

- Set S in Z^n given by linear-optimization oracle
  - set E covering edge-directions of conv(S)
- Binary encoded $d \times n$ matrix W



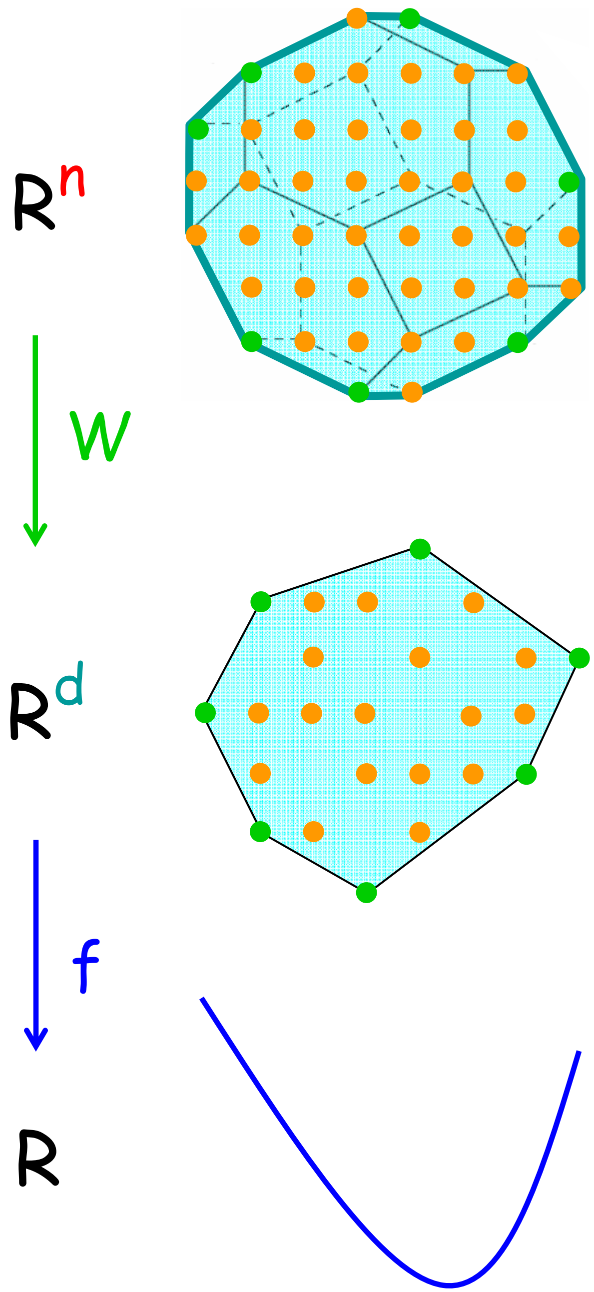


Repeat for all normals and Get vertices of conv(WS) and preimages in S

Since zone(WE) refines conv(WS) assured to get all vertices

Shmuel Onn

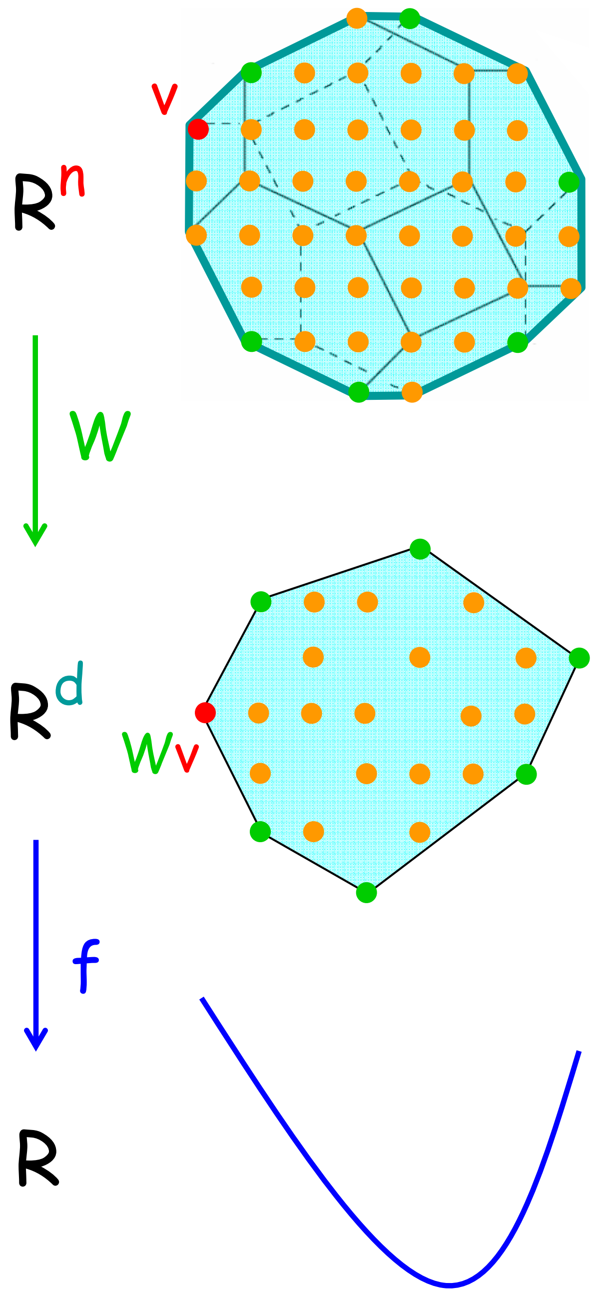
Proof of Theorem 1.1: Convex Discrete Maximization



- Set S in Z^n given by linear-optimization oracle
    - set E covering edge-directions of conv(S)
- Binary encoded $d \times n$ matrix W

- Convex function f on R^d to be maximized

Proof of Theorem 1.1: Convex Discrete Maximization



- Set S in Z^n given by linear-optimization oracle
  - set E covering edge-directions of conv(S)

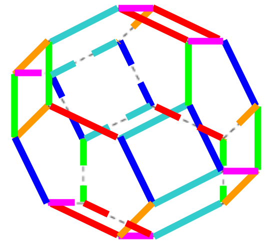
- Binary encoded $d \times n$ matrix W

- Convex function f on R^d to be maximized

Output any preimage v in S maximizing $f(Wv)$

Shmuel Onn

Convex Discrete Maximization - Some Applications



Some edge-behaved polytopes	And their applications
Cubes: unit vectors 1_i	e.g. 0-1 quadratic programming
Matroid polytopes: pairs $1_i - 1_j$ Also poly-matroids	e.g. spanning trees, experimental design
Multiway $m_1 \times \cdots \times m_k \times n$ transportation polytopes:	e.g. partitioning, clustering, multicommodity flows

Shmuel Onn

2. Nonlinear Integer Programming

Data:

- $S = \{x \text{ in } Z^N : A^{(n)}x = b,\ l \le x \le u\}$ given by inequalities
- W binary encoded
- f seperable (min), convex (max), given by comparison oracle

2. Nonlinear Integer Programming

Data:

- $S = \{x \text{ in } Z^N : A^{(n)}x = b,\ l \leq x \leq u\}$ given by inequalities
- W binary encoded
- f seperable (min), convex (max), given by comparison oracle

Methods: **mostly algebraic – Graver bases**

References: **De Loera, Hemmecke, Onn, Weismantel**

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

Shmuel Onn

N-Fold Systems

The n-fold product of $(r+s) \times t$ matrix A is the following $(r+ns) \times nt$ matrix:

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

N-Fold Systems

The n-fold product of $(r+s) \times t$ matrix A is the following $(r+ns) \times nt$ matrix:

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

We have four theorems on n-fold integer programming:

Theorem 2.1: For fixed A, **linear integer programming** over n-fold products of A can be done in polynomial-time:

$$\max \{ wx : A^{(n)}x = b,\ l \le x \le u,\ x \text{ integer}\}$$

Shmuel Onn

N-Fold Systems

The n-fold product of $(r+s) \times t$ matrix A is the following $(r+ns) \times nt$ matrix:

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

We have four theorems on n-fold integer programming:

Theorem 2.2: For fixed A, **separable convex integer minimization** over n-fold products of A can be done in polynomial-time:

$$\min \{ \sum f_i(x_i) \; : \; A^{(n)}x = b, \; l \leq x \leq u, \; x \text{ integer} \}$$

N-Fold Systems

The n-fold product of $(r+s) \times t$ matrix A is the following $(r+ns) \times nt$ matrix:

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

We have four theorems on n-fold integer programming:

Theorem 2.3: For fixed A, **integer point l_p-nearest to a given x** over n-fold products of A can be found in polynomial-time:

$$\min \{ \, |x - x|_p \; : \; A^{(n)}x = b, \; l \le x \le u, \; x \text{ integer} \}$$

Shmuel Onn

N-Fold Systems

The n-fold product of $(r+s) \times t$ matrix A is the following $(r+ns) \times nt$ matrix:

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

We have four theorems on n-fold integer programming:

Theorem 2.4: For fixed d and A, **convex integer maximization** over n-fold products of A can be done in polynomial-time:

$$\max \{ f(Wx) \ : \ A^{(n)}x = b, \ l \leq x \leq u, \ x \text{ integer} \}$$

Preliminaries on Graver Bases

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

Preliminaries on Graver Bases

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

Lemma: For fixed A, can compute in polytime the Graver basis $G(A^{(n)})$ of

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

Preliminaries on Graver Bases

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

Lemma: For fixed A, can compute in polytime the Graver basis $G(A^{(n)})$ of

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

(Use stabilization of $G(A^{(n)})$ due to Hosten, Santos, Sturmfels, Sullivant)

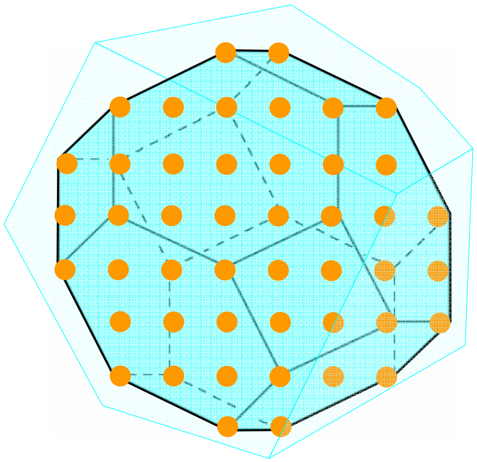
Shmuel Onn

Proof of Theorems 2.1 - 2.3
(linear and convex n-fold minimization)

Shmuel Onn

Proof of Theorems 2.1 - 2.3
(linear and convex n-fold minimization)

R^N

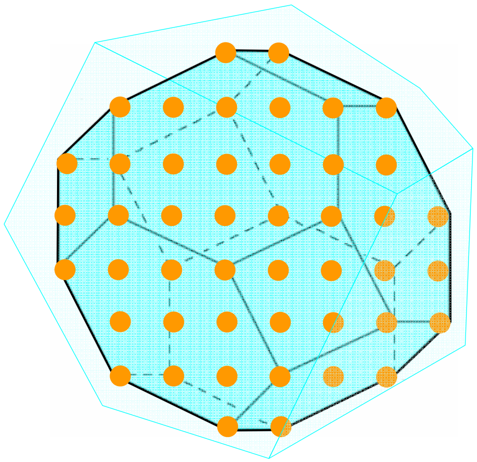


- Set $S = \{x \text{ in } Z^N : A^{(n)}x = b,\ l \le x \le u\}$, $N = nt$

Shmuel Onn

Proof of Theorems 2.1 - 2.3
(linear and convex n-fold minimization)

R^N



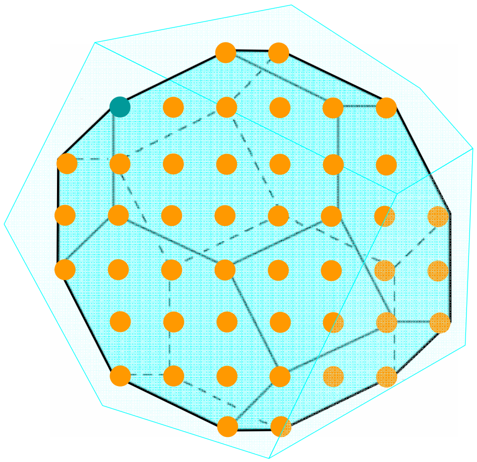
- Set $S = \{ x \text{ in } Z^N : A^{(n)} x = b,\ l \leq x \leq u \}$, $N = nt$

Construct the Graver basis $G(A^{(n)})$

Shmuel Onn

Proof of Theorems 2.1 - 2.3
(linear and convex n-fold minimization)

R^N

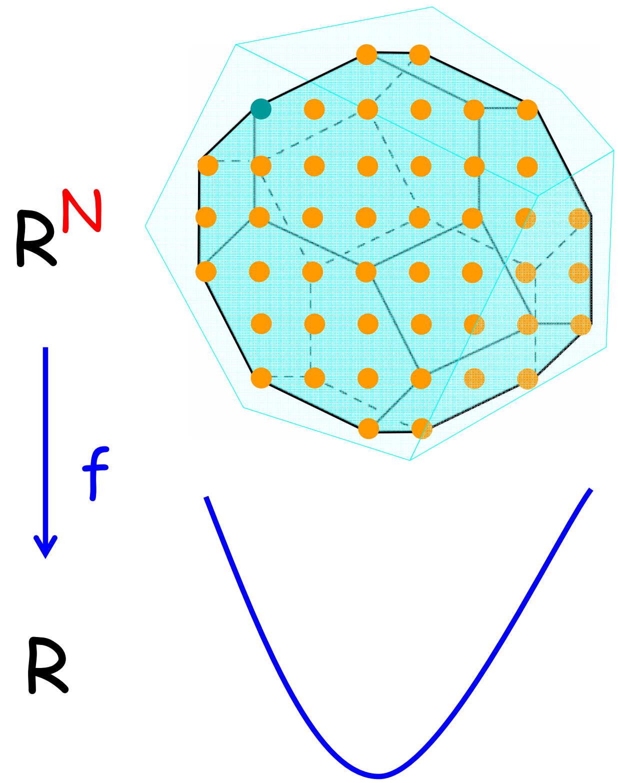


- Set $S = \{x \text{ in } Z^N : A^{(n)}x = b,\ l \le x \le u\}$, $N=nt$

  Construct the Graver basis $G(A^{(n)})$

  Find initial point by auxiliary n-fold program

Proof of Theorems 2.1 - 2.3
(linear and convex n-fold minimization)



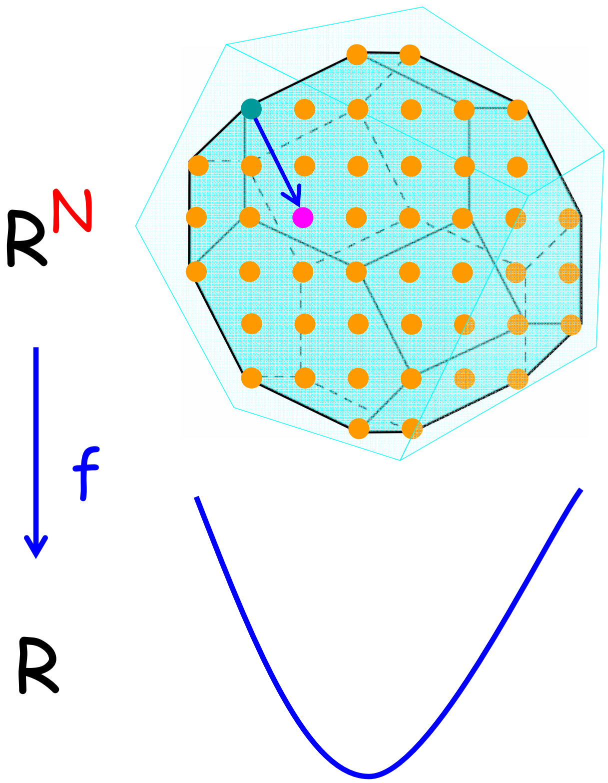
- Set $S = \{x \text{ in } Z^N : A^{(n)}x = b,\ l \le x \le u\}$, $N = nt$

  Construct the Graver basis $G(A^{(n)})$

  Find initial point by auxiliary n-fold program

- Separable convex function f to be minimized

Proof of Theorems 2.1 - 2.3
(linear and convex n-fold minimization)



- Set $S = \{x \text{ in } Z^N : A^{(n)}x = b,\ l \le x \le u\}$, $N=nt$

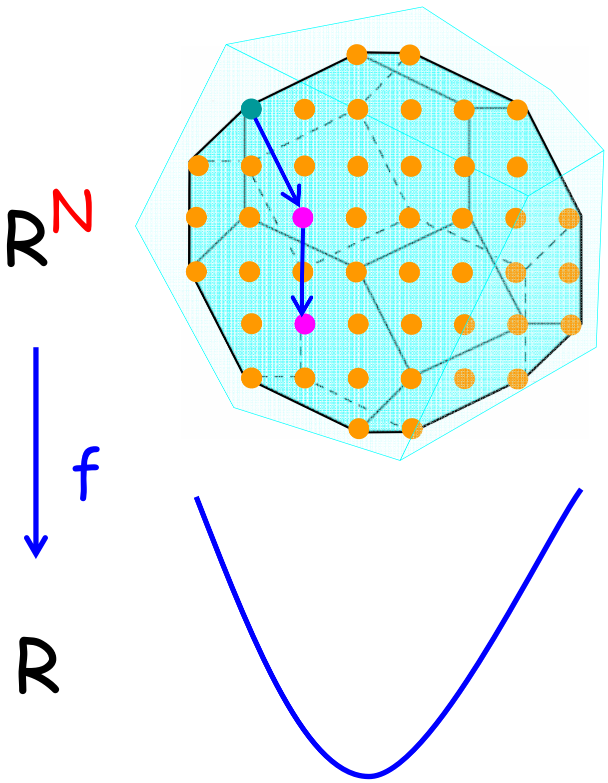
  Construct the Graver basis $G(A^{(n)})$

  Find initial point by auxiliary n-fold program

- Separable convex function f to be minimized

  Iteratively greedily augment initial point
  to optimal one using elements from $G(A^{(n)})$

Proof of Theorems 2.1 - 2.3
(linear and convex n-fold minimization)



- Set $S = \{x \text{ in } Z^N : A^{(n)}x = b,\ l \le x \le u\}$, $N=nt$

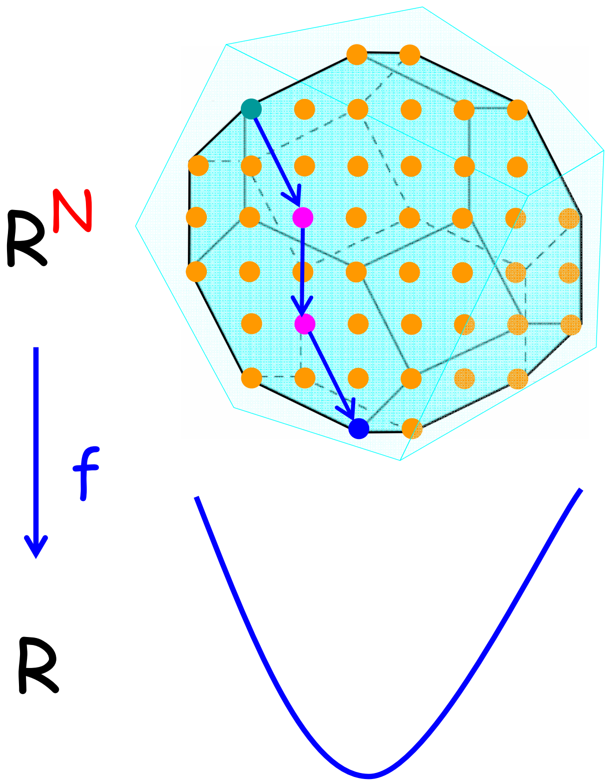
  Construct the Graver basis $G(A^{(n)})$

  Find initial point by auxiliary n-fold program

- Separable convex function f to be minimized

  Iteratively greedily augment initial point to optimal one using elements from $G(A^{(n)})$

Proof of Theorems 2.1 - 2.3
(linear and convex n-fold minimization)



- Set $S = \{x \text{ in } Z^N : A^{(n)}x = b,\ l \le x \le u\}$, $N = nt$

  Construct the Graver basis $G(A^{(n)})$

  Find initial point by auxiliary n-fold program

- Separable convex function f to be minimized

  Iteratively greedily augment initial point to optimal one using elements from $G(A^{(n)})$

Integer Caratheodory Theorem assures polynomial convergence

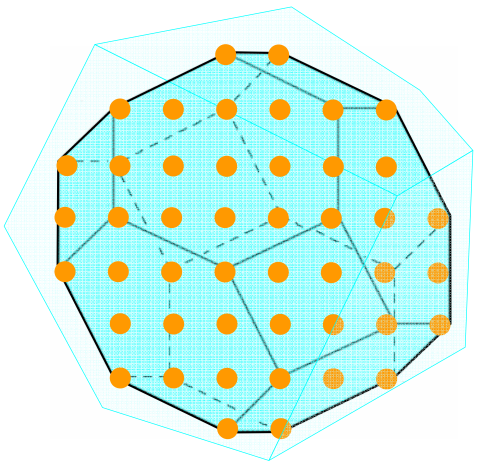
Shmuel Onn

Proof of Theorem 2.4
(convex n-fold maximization)

Shmuel Onn

Proof of Theorem 2.4
(convex n-fold maximization)

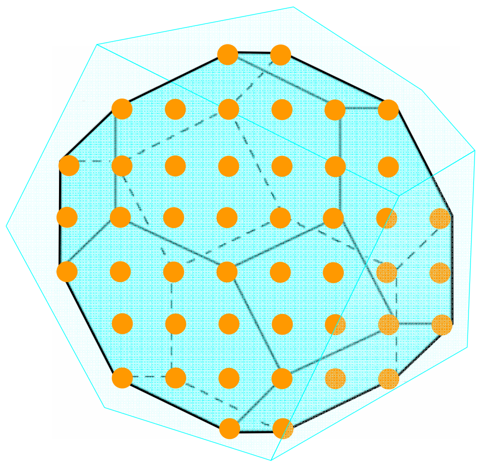
R^N



- Set $S = \{x \text{ in } Z^N : A^{(n)}x = b,\ l \leq x \leq u\}$, $N = nt$

Proof of Theorem 2.4
(convex n-fold maximization)

R^N

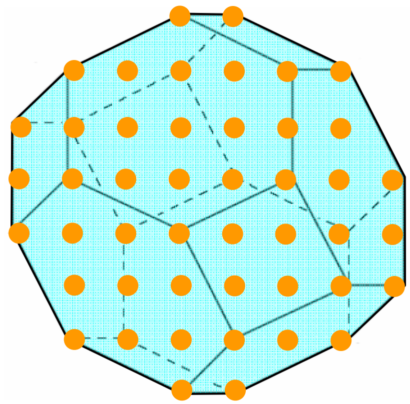


- Set $S = \{x \text{ in } Z^N : A^{(n)}x = b,\ l \leq x \leq u\}$, $N = nt$

  Construct the Graver basis $G(A^{(n)})$

Proof of Theorem 2.4
(convex n-fold maximization)

R^N



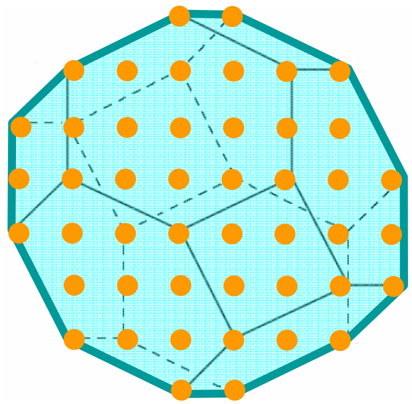
- Set $S = \{ x \text{ in } Z^N : A^{(n)} x = b,\ l \le x \le u \}$, $N = nt$

Construct the Graver basis $G(A^{(n)})$

Simulate linear-optimization oracle over S using Theorem 2.1

Proof of Theorem 2.4
(convex n-fold maximization)

R^N



- Set $S = \{ x \text{ in } Z^N : A^{(n)}x = b,\ l \le x \le u \}$, $N = nt$

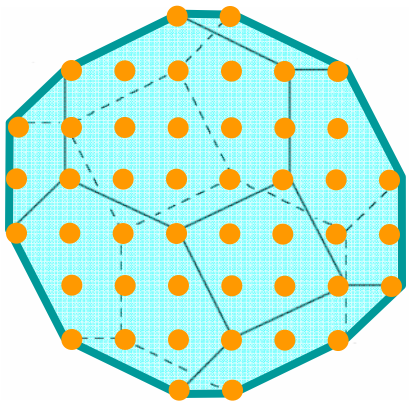
Construct the Graver basis $G(A^{(n)})$

Simulate linear-optimization oracle over S using Theorem 2.1

The Graver basis covers all edge-directions of conv(S)

Proof of Theorem 2.4
(convex n-fold maximization)

R^N



- Set $S = \{x \text{ in } Z^N : A^{(n)}x = b,\ l \le x \le u\}$, $N = nt$

Construct the Graver basis $G(A^{(n)})$

Simulate linear-optimization oracle over S using Theorem 2.1

The Graver basis covers all edge-directions of conv(S)

Apply Theorem 1.1 on convex discrete maximization

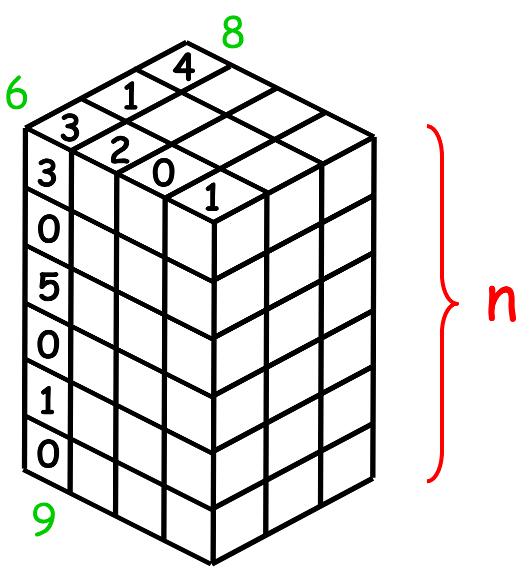
Applications & Universality of N-Folds: Multiway Tables

Shmuel Onn

Applications & Universality of N-Folds: Multiway Tables

The $m_1 \times \cdots \times m_k \times n$ tables with given line-sums form an n-fold system

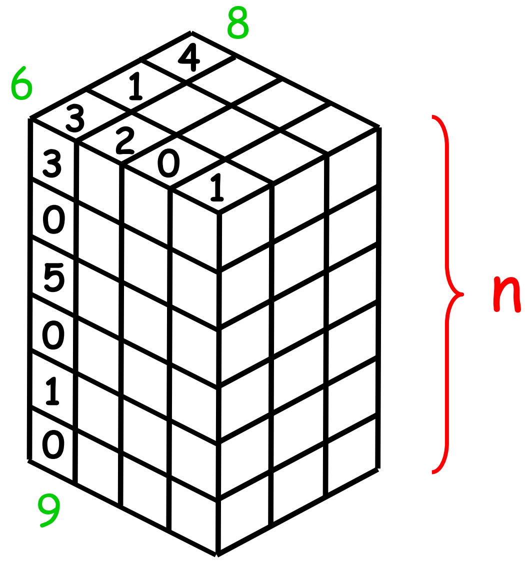
$S = \{x : A^{(n)}x = b,\ x \geq 0,\ x \text{ integer}\}$ for suitable A depending on $m_1, \ldots, m_k$



Shmuel Onn

Applications & Universality of N-Folds: Multiway Tables

The $m_1 \times \cdots \times m_k \times n$ tables with given line-sums form an n-fold system $S = \{x : A^{(n)}x = b,\ x \geq 0,\ x \text{ integer}\}$ for suitable A depending on $m_1, \ldots, m_k$

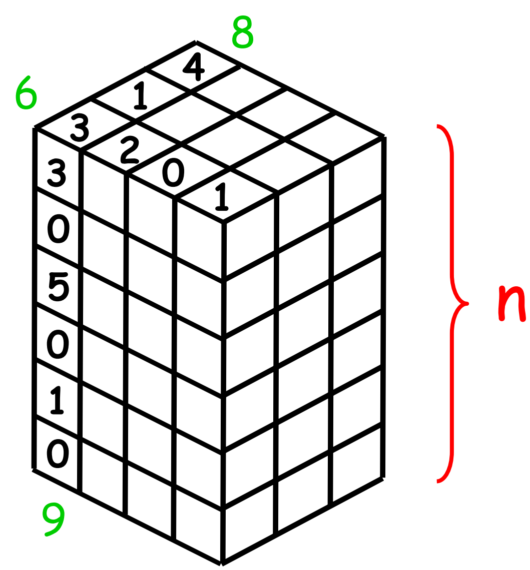


Corollary of Theorems 2.1 – 2.4: nonlinear discrete optimization over $m_1 \times \cdots \times m_k \times n$ tables with given line-sums can be done in polynomial-time

Shmuel Onn

Applications & Universality of N-Folds: Multiway Tables

The $m_1 \times \cdots \times m_k \times n$ tables with given line-sums form an n-fold system $S = \{x : A^{(n)}x = b,\ x \geq 0,\ x \text{ integer}\}$ for suitable A depending on $m_1, \ldots, m_k$

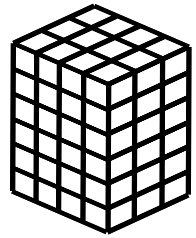


Corollary of Theorems 2.1 – 2.4: nonlinear discrete optimization over $m_1 \times \cdots \times m_k \times n$ tables with given line-sums can be done in polynomial-time

Universality Theorem for N-fold Systems (De Loera - Onn, SIAM Opt.)**:**
Every integer program is one over $3 \times m \times n$ tables with given line-sums

Shmuel Onn

N-fold Systems and Multiway Tables – Some Applications



Some Applications	References
Privacy in statistical data-bases	Onn (Lect. Notes Comp. Sci.)
Multicommodity flows	Berstein - Onn (Annals Combin.)
Congestion-avoiding transportation	Hemmecke - Onn - Weismantel
Error-correcting codes	- || -
Universality - scheme for arbitrary nonlinear integer programming	De Loera - Onn (SIAM Opt.)

Shmuel Onn

3. Nonlinear Combinatorial Optimization

Data:

- S {0,1}-valued with combinatorial structure
  - e.g. matchings, matroid intersections, independence systems
- W unary encoded
- f arbitrary given by comparison oracle

3. Nonlinear Combinatorial Optimization

Data:

- S {0,1}-valued with combinatorial structure
  e.g. matchings, matroid intersections, independence systems
- W unary encoded
- f arbitrary given by comparison oracle

Methods: mostly algebraic - polynomial identities and interpolation, number theory (Frobenius)

References: **Berstein, Lee, Onn, Weismantel, Wynn**

- Nonlinear bipartite matching (Disc. Opt.)
- Nonlinear matroid optimization and experimental design (SIAM Disc. Math.)
- Nonlinear optimization for matroid intersection and extensions (submitted)
- Nonlinear optimization over a weighted independence system (submitted)

Nonlinear Combinatorial Optimization

Theorem 3.1: Fix any d. Then can solve in randomized polynomial-time

$$\min/\max \{ f(Wx) \; : \; x \text{ in } S \}$$

for S in $\{0,1\}^n$ bipartite matchings or matroid intersection, unary W, any f.

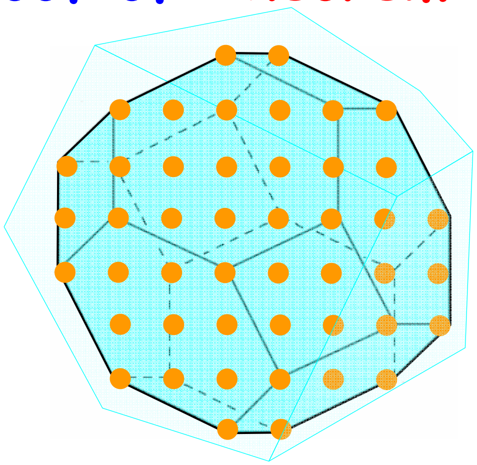
Proof of Theorem 3.1: Nonlinear Combinatorial Optimization

(very rough outline)

Shmuel Onn

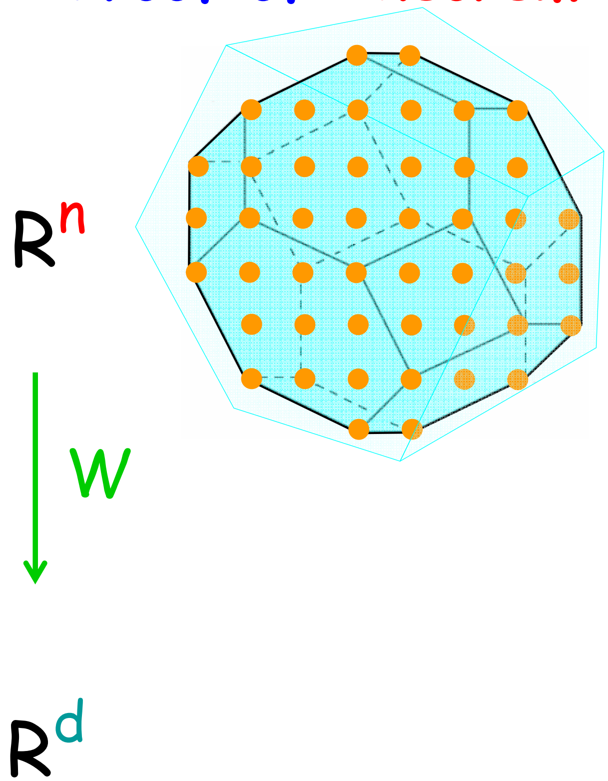
Proof of Theorem 3.1: Nonlinear Combinatorial Optimization

R^n



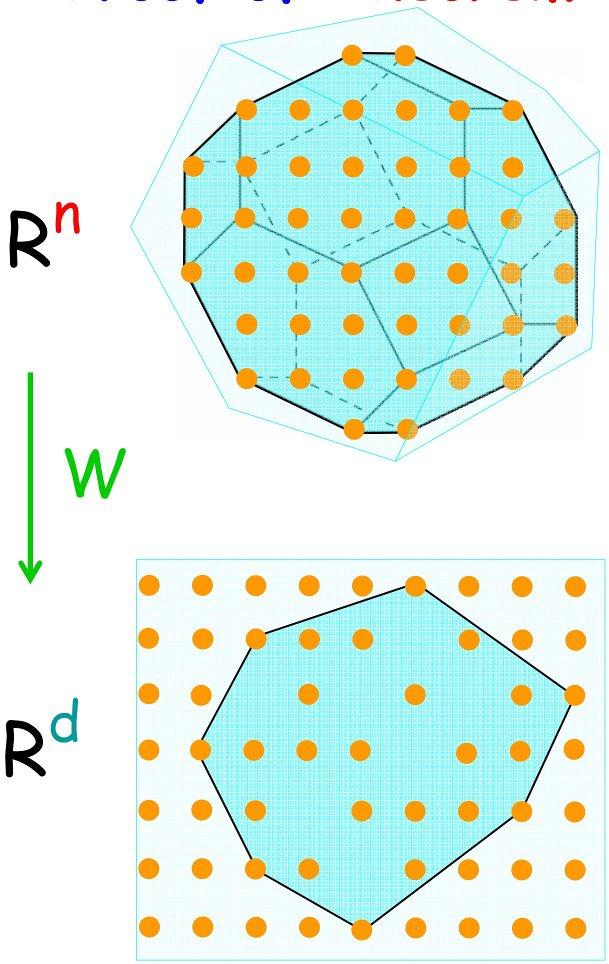
- Set S in $\{0,1\}^n$ - bipartite matchings in graph or intersection of matroids of two matrices

Proof of Theorem 3.1: Nonlinear Combinatorial Optimization



- Set S in $\{0,1\}^n$ - bipartite matchings in graph or intersection of matroids of two matrices
- Unary encoded $d \times n$ matrix W

Proof of Theorem 3.1: Nonlinear Combinatorial Optimization

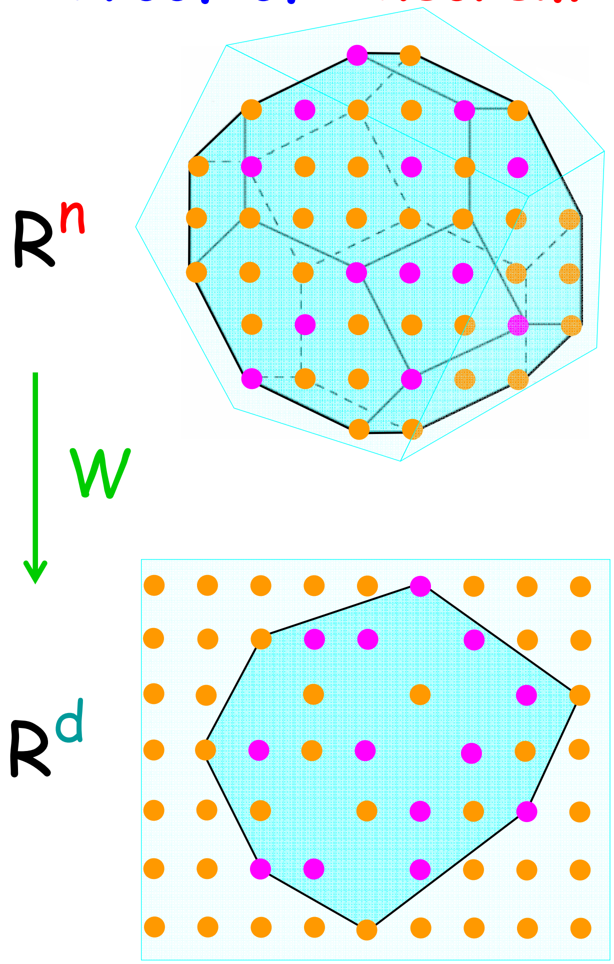


- Set S in $\{0,1\}^n$ - bipartite matchings in graph or intersection of matroids of two matrices

- Unary encoded $d \times n$ matrix W

Difficulty :
cannot check locally if any point is in WS

Proof of Theorem 3.1: Nonlinear Combinatorial Optimization

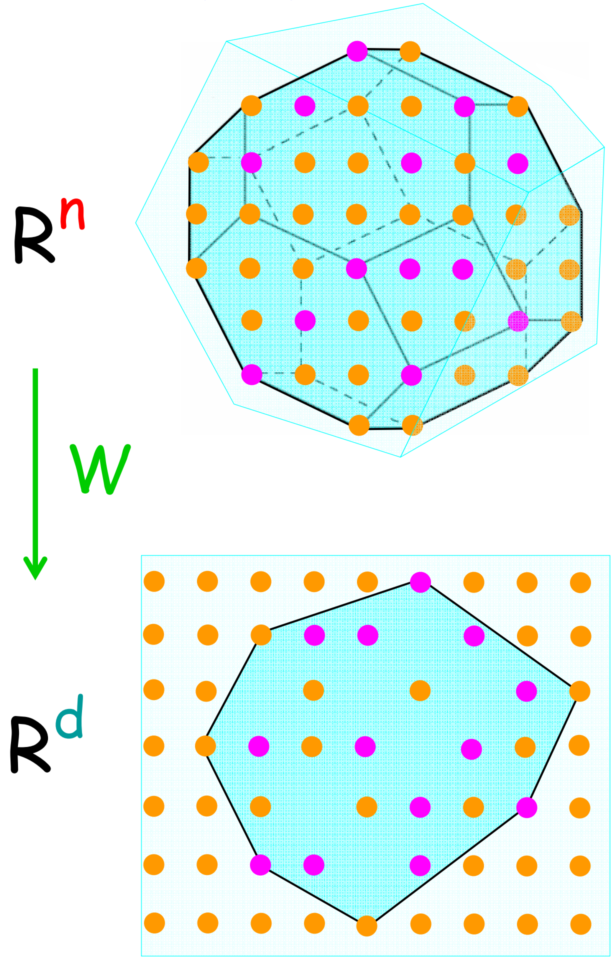


- Set S in $\{0,1\}^n$ - bipartite matchings in graph or intersection of matroids of two matrices

- Unary encoded $d \times n$ matrix W

Instead:
construct globally random subset of WS containing each point with high probability along with suitable random preimages in S

Proof of Theorem 3.1: Nonlinear Combinatorial Optimization



- Set S in $\{0,1\}^n$ - bipartite matchings in graph or intersection of matroids of two matrices

- Unary encoded $d \times n$ matrix W

Instead:

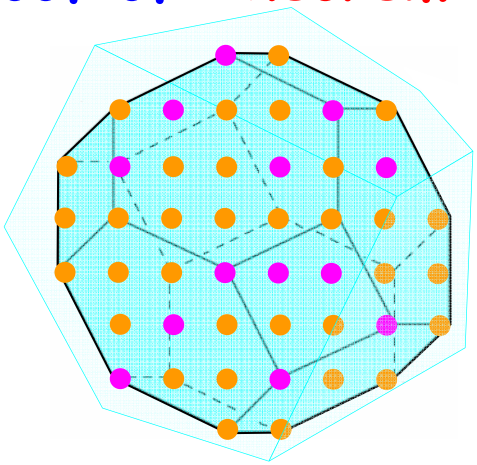
construct globally random subset of WS containing each point with high probability along with suitable random preimages in S

Use:

- Schwartz polynomial-identity testing
- Binet-Cauchy identity
- interpolation

Proof of Theorem 3.1: Nonlinear Combinatorial Optimization

R^n

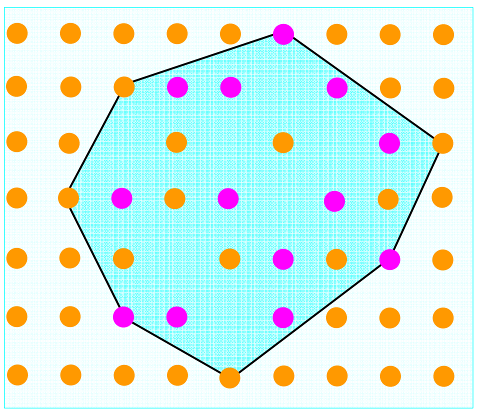


- Set S in $\{0,1\}^n$ - bipartite matchings in graph or intersection of matroids of two matrices

- Unary encoded $d \times n$ matrix W

$\downarrow W$

R^d

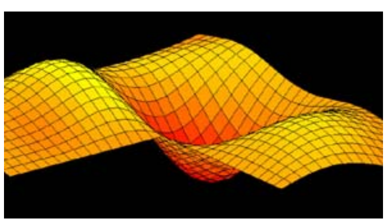


Instead:
construct globally random subset of WS containing each point with high probability along with suitable random preimages in S

Use:
- Schwartz polynomial-identity testing
- Binet-Cauchy identity
- interpolation

$\downarrow f$

R

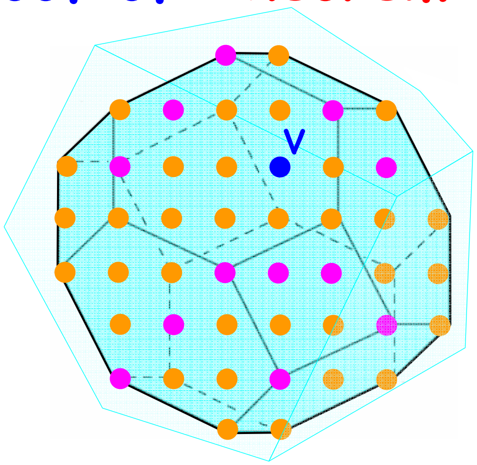


- function f on R^d to be minimized

Shmuel Onn

Proof of Theorem 3.1: Nonlinear Combinatorial Optimization

R^n



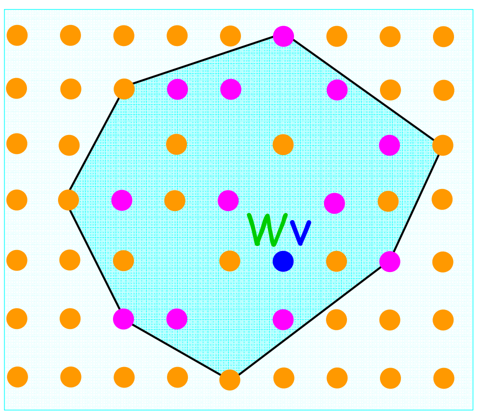
- Set S in $\{0,1\}^n$ - bipartite matchings in graph
  or intersection of matroids of two matrices

- Unary encoded $d \times n$ matrix W

$\downarrow W$

Instead:
construct globally random subset of WS
containing each point with high probability
along with suitable random preimages in S

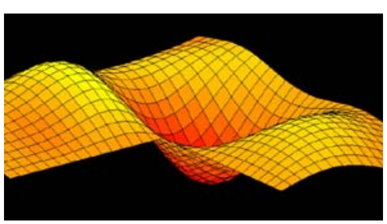
R^d



Use:
- Schwartz polynomial-identity testing
- Binet-Cauchy identity
- interpolation

$\downarrow f$

R



- function f on R^d to be minimized

Output v in S minimizing $f(Wv)$ on random subset

Shmuel Onn

Arbitrary Independence Systems: finding r-best solution

Shmuel Onn

Arbitrary Independence Systems: finding r-best solution

Theorem 3.2: For $d=1$ we can find in polynomial-time r-best solution of

$$\min/\max \{ f(Wx) \ : \ x \text{ in } S \}$$

for S independence system given by lin-opt oracle, W in $\{a_1, \ldots, a_p\}^n$, any f

Arbitrary Independence Systems: finding r-best solution

Theorem 3.2: For $d=1$ we can find in polynomial-time r-best solution of

$$\min/\max \{ f(Wx) \ : \ x \text{ in } S \}$$

for S independence system given by lin-opt oracle, W in $\{a_1, \ldots, a_p\}^n$, any f

Note:

A solution is r-best if there are at most r better objective values.

$r = r(a_1, \ldots, a_p)$ is constant, with $r(a_1, a_2) = F(a_1, a_2)$ the Frobenius number.

Arbitrary Independence Systems: finding r-best solution

Theorem 3.2: For $d=1$ we can find in polynomial-time r-best solution of

$$\min/\max \{ f(Wx) \ : \ x \text{ in } S \}$$

for S independence system given by lin-opt oracle, W in $\{a_1, \ldots, a_p\}^n$, any f

Note:

A solution is r-best if there are at most r better objective values.

$r = r(a_1, \ldots, a_p)$ is constant, with $r(a_1, a_2) = F(a_1, a_2)$ the Frobenius number.

So for W in $\{2,3\}^n$ can efficiently find a 1-best solution.

Arbitrary Independence Systems: finding r-best solution

Theorem 3.2: For $d=1$ we can find in polynomial-time r-best solution of

$$\min/\max \{ f(Wx) : x \text{ in } S \}$$

for S independence system given by lin-opt oracle, W in $\{a_1, \ldots, a_p\}^n$, any f

Note:

A solution is r-best if there are at most r better objective values.

$r = r(a_1, \ldots, a_p)$ is constant, with $r(a_1, a_2) = F(a_1, a_2)$ the Frobenius number.

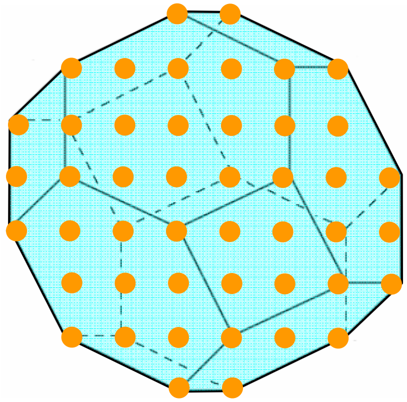
So for W in $\{2,3\}^n$ can efficiently find a 1-best solution. But amazingly:

Theorem 3.3: With data as above and W in $\{2,3\}^n$, finding an optimal (0-best) solution requires exponential-time.

Finding 1-best versus Optimal (0-best) Solution

Finding 1-best versus Optimal (0-best) Solution

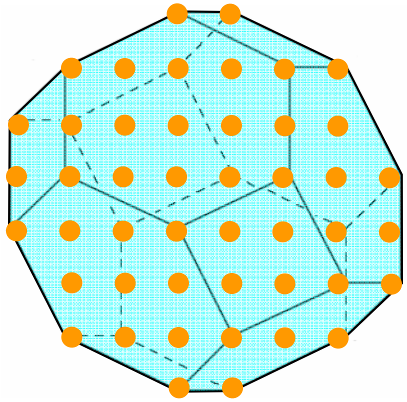
R^n



- Set S in $\{0,1\}^n$ independence system
  given by linear-optimization oracle

Finding 1-best versus Optimal (0-best) Solution

R^n



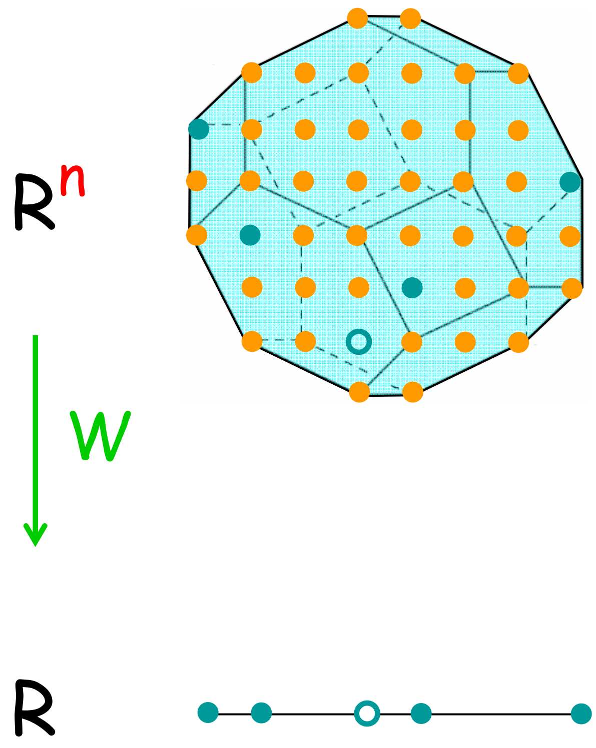
$\downarrow W$

R

- Set S in $\{0,1\}^n$ independence system
  given by linear-optimization oracle

- Weight vector W in $\{2,3\}^n$

Finding 1-best versus Optimal (0-best) Solution

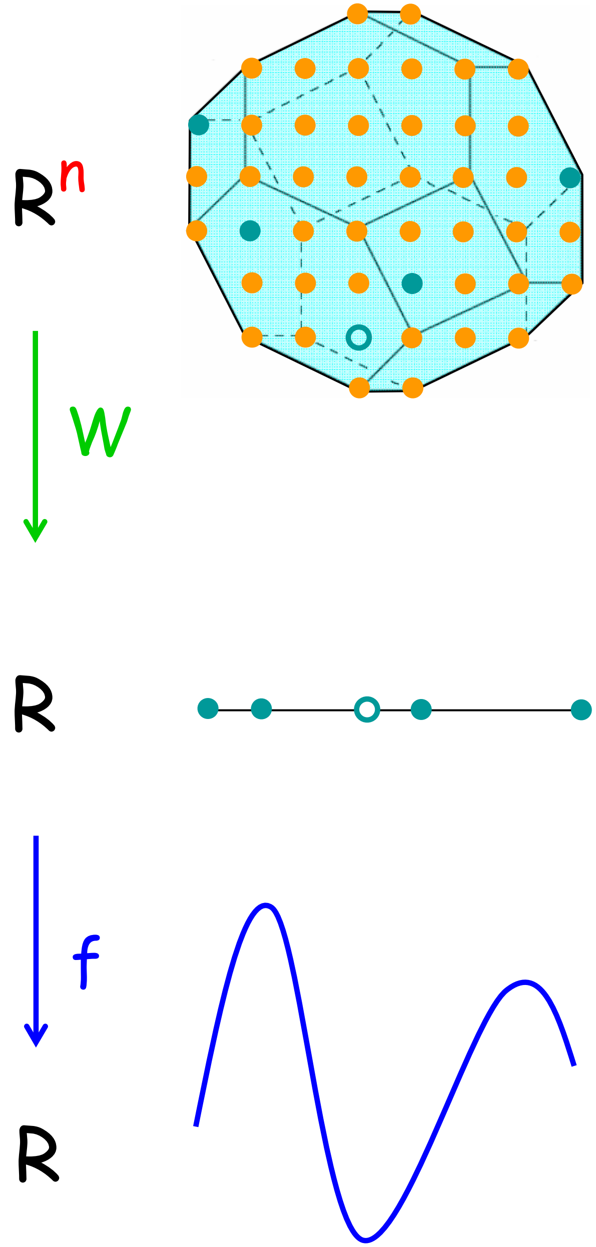


- Set S in $\{0,1\}^n$ independence system
  given by linear-optimization oracle

- Weight vector W in $\{2,3\}^n$

Can compute in polynomial-time
the image WS in R up to a point
But need exponential-time
to compute WS exactly

Finding 1-best versus Optimal (0-best) Solution



- Set S in $\{0,1\}^n$ independence system given by linear-optimization oracle

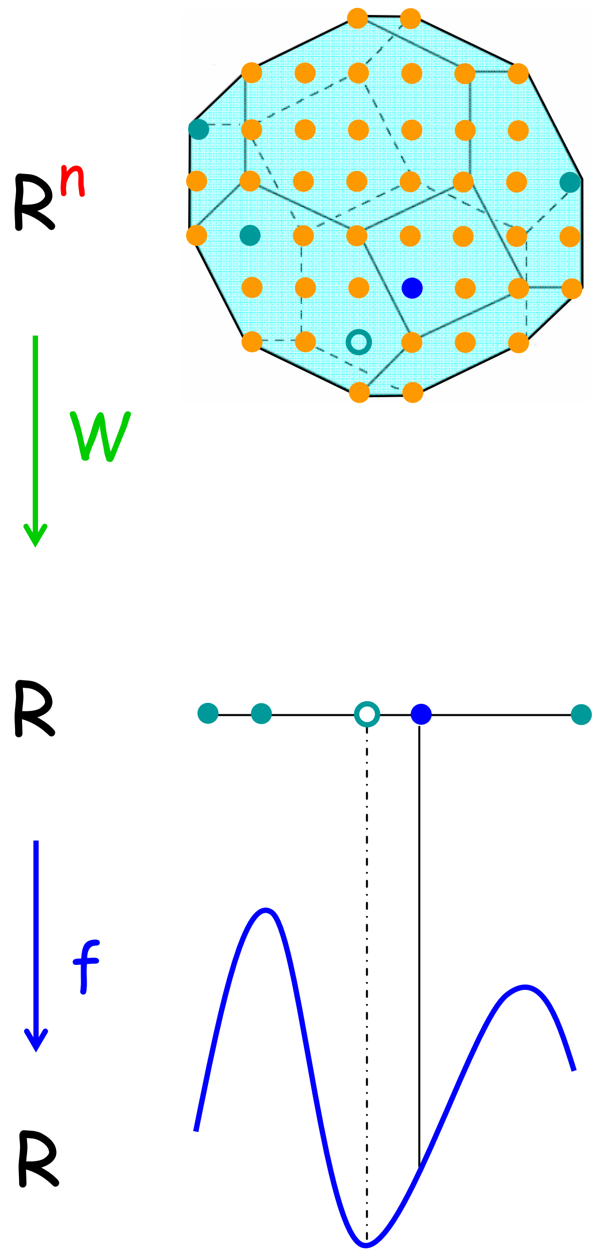
- Weight vector W in $\{2,3\}^n$

Can compute in polynomial-time the image WS in R up to a point
But need exponential-time to compute WS exactly

- Function f on R to be minimized

Shmuel Onn

Finding 1-best versus Optimal (0-best) Solution



- Set S in $\{0,1\}^n$ independence system given by linear-optimization oracle

- Weight vector W in $\{2,3\}^n$

  Can compute in polynomial-time the image WS in R up to a point
  But need exponential-time to compute WS exactly

- Function f on R to be minimized

Can get 1-best but not optimal solution

Shmuel Onn

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

Onn: Convex Discrete Optimization (SMS Lecture Notes, CRM Montréal)

Lee, Onn, Weismantel: Nonlinear Discrete Optimization (Monograph in preparation)

- Partition problems with convex objectives (Math. OR)
- Convex matroid optimization (SIAM Disc. Math.)
- The complexity of 3-way tables (SIAM Comp.)
- Convex combinatorial optimization (Disc. Comp. Geom.)
- Markov bases of 3-way tables (J. Symb. Comp.)
- All linear and integer programs are slim 3-way programs (SIAM Opt.)
- N-fold integer programming (Disc. Opt. in memory of Dantzig)
- Entry Uniqueness in margined tables (Lect. Notes Comp. Sci.)
- Graver complexity of integer programming (Annals Combin.)
- Nonlinear bipartite matching (Disc. Opt.)
- Convex integer maximization via Graver bases (J. Pure App. Algebra)
- Nonlinear matroid optimization and experimental design (SIAM Disc. Math.)
- Convex integer minimization (submitted)
- Nonlinear optimization for matroid intersection and extensions (submitted)
- Nonlinear optimization over a weighted independence system (submitted)