Degree Sequence Optimization and Sparse Integer Programming

Shmuel Onn

Technion - Israel Institute of Technology

https://sites.google.com/view/shmuel-onn

Degree Sequence Optimization (DSO)

All our graphs and subgraphs are on vertex set $[n] := \{1, . . ., n\}$

Degree Sequence Optimization (DSO)

All our graphs and subgraphs are on vertex set $[n] := \{1, . . ., n\}$

The degree of vertex i in graph G is denoted $d_i(G)$

Degree Sequence Optimization (DSO)

All our graphs and subgraphs are on vertex set $[n] := \{1, . . ., n\}$

The degree of vertex i in graph G is denoted $d_i(G)$

The general Degree Sequence Optimization problem is:

Given a graph H and a function $f_i : \{0,1, . . ., d_i(H)\} \text{ ---> } Z$
for each vertex i, find a subgraph G of H which minimizes

$$f_1(d_1(G)) + . . . + f_n(d_n(G))$$

Degree Sequence Optimization (DSO)

All our graphs and subgraphs are on vertex set $[n] := \{1, . . ., n\}$

The degree of vertex i in graph G is denoted $d_i(G)$

The general Degree Sequence Optimization problem is:

Given a graph H and a function $f_i : \{0,1, . . ., d_i(H)\} \dashrightarrow Z$
for each vertex i, find a subgraph G of H which minimizes

$$f_1(d_1(G)) + . . . + f_n(d_n(G))$$

Example: with $f_i(x)=(x-1)^2$ for all i can decide if H has a perfect matching

Degree Sequence Optimization (DSO)

All our graphs and subgraphs are on vertex set $[n] := \{1, . . ., n\}$

The degree of vertex i in graph G is denoted $d_i(G)$

The general Degree Sequence Optimization problem is:

Given a graph H and a function $f_i : \{0,1, . . ., d_i(H)\}$ ---> Z
for each vertex i, find a subgraph G of H which minimizes

$$f_1(d_1(G)) + . . . + f_n(d_n(G))$$

NP-hard: H has a nonempty cubic subgraph if and only if for some i the optimal value of DSO with $f_i(x)=(x-3)^2$ and $f_j(x)=x(x-3)^2$ for all $j \neq i$ is zero

Some Special Cases

Complete Graphs

When $H = K_n$ is the complete graph, the problem is:

Complete Graphs

When $H = K_n$ is the complete graph, the problem is:

Given functions $f_i : \{0,1, . . ., n-1\} \rightarrow Z$ find a graph G minimizing

$$f_1(d_1(G)) + . . . + f_n(d_n(G))$$

Complete Graphs

When $H = K_n$ is the complete graph, the problem is:

Given functions $f_i : \{0,1, . . ., n-1\} \dashrightarrow Z$ find a graph G minimizing

$$f_1(d_1(G)) + . . . + f_n(d_n(G))$$

Complexity still open! But:

Complete Graphs

When $H = K_n$ is the complete graph, the problem is:

Given functions $f_i : \{0,1, . . ., n-1\} \dashrightarrow Z$ find a graph G minimizing

$$f_1(d_1(G)) + . . . + f_n(d_n(G))$$

Complexity still open! But:

Theorem (Deza-Levin-Meesum-Onn 2018)

Polynomial time solvable when all functions are the same $f_1 = . . . = f_n$

Complete Graphs

When $H = K_n$ is the complete graph, the problem is:

Given functions $f_i : \{0,1, . . ., n-1\} \dashrightarrow Z$ find a graph G minimizing

$$f_1(d_1(G)) + . . . + f_n(d_n(G))$$

Deciding if $(d_1, . . ., d_n)$ is a degree sequence: set $f_i(x)=(x-d_i)^2$ for all i

Complete Graphs

When $H = K_n$ is the complete graph, the problem is:

Given functions $f_i : \{0,1, . . ., n-1\} \dashrightarrow Z$ find a graph G minimizing

$$f_1(d_1(G)) + . . . + f_n(d_n(G))$$

Deciding if $(d_1, . . ., d_n)$ is a degree sequence: set $f_i(x)=(x-d_i)^2$ for all i

Polynomial time by the characterization of Erdos-Gallai, but in contrast:

Shmuel Onn

Complete Graphs

When $H = K_n$ is the complete graph, the problem is:

Given functions $f_i : \{0,1, . . ., n-1\} \dashrightarrow Z$ find a graph G minimizing

$$f_1(d_1(G)) + . . . + f_n(d_n(G))$$

Deciding if $(d_1, . . ., d_n)$ is a degree sequence: set $f_i(x)=(x-d_i)^2$ for all i

Polynomial time by the characterization of Erdos-Gallai, but in contrast:

Theorem (Deza-Levin-Meesum-Onn 2018)
NP-complete to decide if $(d_1, . . ., d_n)$ is degree sequence of 3-hypergraph

Complete Graphs

When $H = K_n$ is the complete graph, the problem is:

Given functions $f_i : \{0,1, . . ., n-1\} \dashrightarrow Z$ find a graph G minimizing

$$f_1(d_1(G)) + . . . + f_n(d_n(G))$$

Deciding if $(d_1, . . ., d_n)$ is a degree sequence: set $f_i(x)=(x-d_i)^2$ for all i

Polynomial time by the characterization of Erdos-Gallai, but in contrast:

Theorem (Deza-Levin-Meesum-Onn 2018)
NP-complete to decide if $(d_1, . . ., d_n)$ is degree sequence of 3-hypergraph

This answers a 30 year long open question (Colbourn et al. 1986)

Complete Bipartite Graphs

When $H = K_{m,n}$ is the complete bipartite graph, the problem is equivalent to the line-sum optimization problem over matrices:

Complete Bipartite Graphs

When $H = K_{m,n}$ is the complete bipartite graph, the problem is equivalent to the line-sum optimization problem over matrices:

Given functions f_j, g_i find an mxn 0-1 matrix A minimizing

$$f_1(c_1(A)) + \ldots + f_n(c_n(A)) + g_1(r_1(A)) + \ldots + g_m(r_m(A))$$

Complete Bipartite Graphs

When $H = K_{m,n}$ is the complete bipartite graph, the problem is equivalent to the line-sum optimization problem over matrices:

Given functions f_j, g_i find an mxn 0-1 matrix A minimizing

$$f_1(c_1(A)) + \ldots + f_n(c_n(A)) + g_1(r_1(A)) + \ldots + g_m(r_m(A))$$

Complexity still open! But:

Complete Bipartite Graphs

When $H = K_{m,n}$ is the complete bipartite graph, the problem is equivalent to the line-sum optimization problem over matrices:

Given functions f_j, g_i find an mxn 0-1 matrix A minimizing

$$f_1(c_1(A)) + \ldots + f_n(c_n(A)) + g_1(r_1(A)) + \ldots + g_m(r_m(A))$$

Complexity still open! But:

Theorem (Koutecky-Onn 2021)
Polynomial time solvable over monotone matrices A, that is, having nonincreasing row sums $r_i(A)$ and column sums $c_j(A)$.

Complete Bipartite Graphs

When $H = K_{m,n}$ is the complete bipartite graph, the problem is equivalent to the line-sum optimization problem over matrices:

Given functions f_j, g_i find an mxn 0-1 matrix A minimizing

$$f_1(c_1(A)) + \ldots + f_n(c_n(A)) + g_1(r_1(A)) + \ldots + g_m(r_m(A))$$

Complexity still open! But:

Theorem (Koutecky-Onn 2021)
Polynomial time solvable over monotone matrices A, that is, having nonincreasing row sums $r_i(A)$ and column sums $c_j(A)$.
In particular, solvable when $f_1 = \ldots = f_n$ and $g_1 = \ldots = g_m$

Complete Bipartite Graphs

When $H = K_{m,n}$ is the complete bipartite graph, the problem is equivalent to the line-sum optimization problem over matrices:

Given functions f_j, g_i find an mxn 0-1 matrix A minimizing

$$f_1(c_1(A)) + \ldots + f_n(c_n(A)) + g_1(r_1(A)) + \ldots + g_m(r_m(A))$$

Complexity still open! But:

Theorem (Koutecky-Onn 2021)
Polynomial time solvable over monotone matrices A, that is, having nonincreasing row sums $r_i(A)$ and column sums $c_j(A)$.
In particular, solvable when $f_1 = \ldots = f_n$ and $g_1 = \ldots = g_m$

Proof: Involved dynamic programming with states being 7-tuples

Convex Functions and General Factors

The general Factor problem is:

Given a graph H and a subset B_i of $\{0,1, . . ., d_i(H)\}$ for each i, decide if H has a subgraph G with $d_i(G)$ in B_i for each i

Convex Functions and General Factors

The general Factor problem is:

Given a graph H and a subset B_i of $\{0,1, . . ., d_i(H)\}$ for each i,
decide if H has a subgraph G with $d_i(G)$ in B_i for each i

Polytime if each B_i interval (Lovasz). Stronger, if no 2-gaps (Cornuejols)

Convex Functions and General Factors

The general Factor problem is:

Given a graph H and a subset B_i of $\{0,1, . . ., d_i(H)\}$ for each i,
decide if H has a subgraph G with $d_i(G)$ in B_i for each i

Polytime if each B_i interval (Lovasz). Stronger, if no 2-gaps (Cornuejols)

Reduces to DSO with $f_i(x):=0$ for x in B_i and $f_i(x):=1$ for x not in B_i

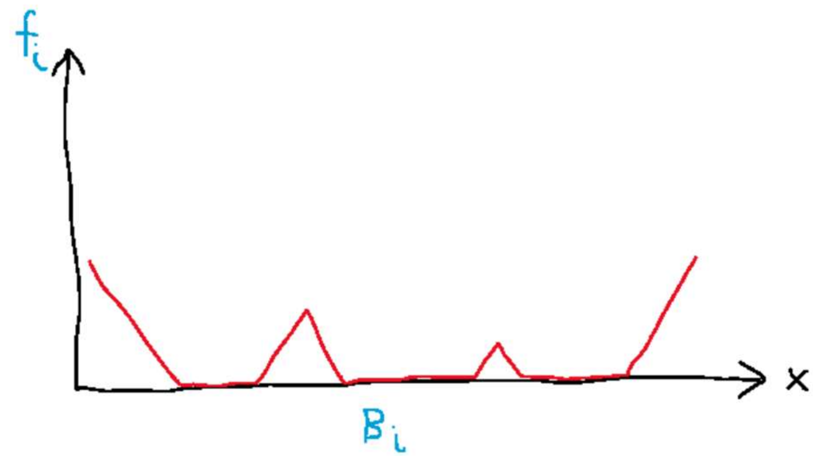
Convex Functions and General Factors

The general Factor problem is:

Given a graph H and a subset B_i of $\{0,1, \ldots, d_i(H)\}$ for each i, decide if H has a subgraph G with $d_i(G)$ in B_i for each i

Polytime if each B_i interval (Lovasz). Stronger, if no 2-gaps (Cornuejols)

Also reduces to DSO with the following which is convex for B_i interval:



Shmuel Onn

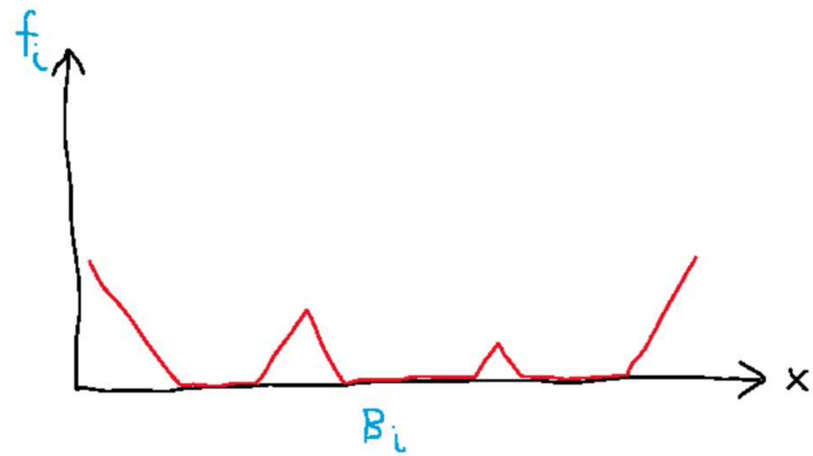
Convex Functions and General Factors

The general Factor problem is:

Given a graph H and a subset B_i of $\{0,1, . . ., d_i(H)\}$ for each i,
decide if H has a subgraph G with $d_i(G)$ in B_i for each i

Polytime if each B_i interval (Lovasz). Stronger, if no 2-gaps (Cornuejols)

Also reduces to DSO with the following which is convex for B_i interval:



Theorem (Deza-Onn, Apollonio-Sebo)

DSO is polynomial time solvable for any convex f_i and any graph H

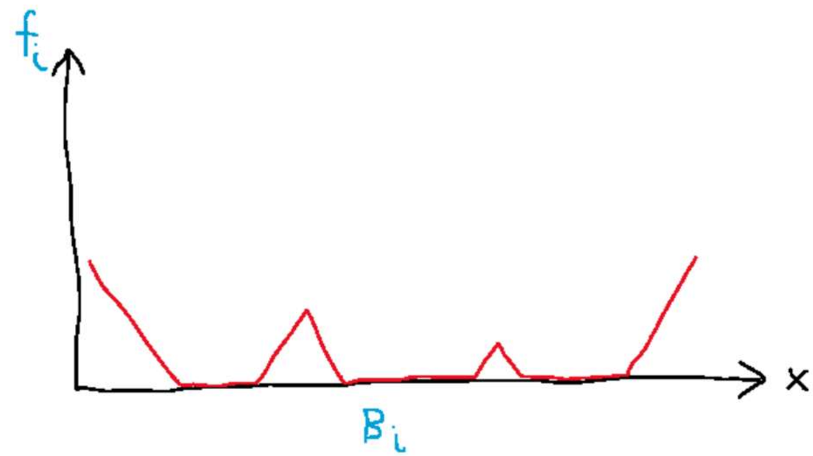
Convex Functions and General Factors

The general Factor problem is:

Given a graph H and a subset B_i of $\{0,1, . . ., d_i(H)\}$ for each i,
decide if H has a subgraph G with $d_i(G)$ in B_i for each i

Polytime if each B_i interval (Lovasz). Stronger, if no 2-gaps (Cornuejols)

Also reduces to DSO with the following which is convex for B_i interval:



Theorem (Deza-Onn, Apollonio-Sebo)

DSO is polynomial time solvable for any convex f_i and any graph H

Proof: Reduces to weighted matching on a suitable larger graph

Bounded Tree Width or Depth

Theorem (Onn 2022) For any fixed k, can solve DSO in polynomial time for any functions f_i and any graph H with tw(H) ≤ k

Bounded Tree Width or Depth

Theorem (Onn 2022) For any fixed k, can solve DSO in polynomial time for any functions f_i and any graph H with tw(H) ≤ k

I will show an earlier result which follows easily from recent results on Sparse Integer Programming forming a useful tool that I will show next

Bounded Tree Width or Depth

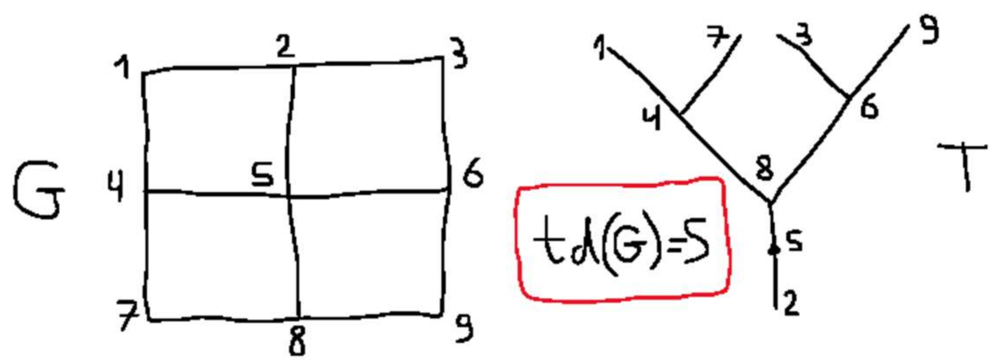
Theorem (Onn 2020) For any fixed k, can solve DSO in polynomial time for any functions f_i and any graph H with td(H) ≤ k

I will show an earlier result which follows easily from recent results on Sparse Integer Programming forming a useful tool that I will show next

Bounded Tree Width or Depth

Theorem (Onn 2020) For any fixed k, can solve DSO in polynomial time for any functions f_i and any graph H with td(H) ≤ k

The height of a rooted tree is max number of vertices on a root-leaf path

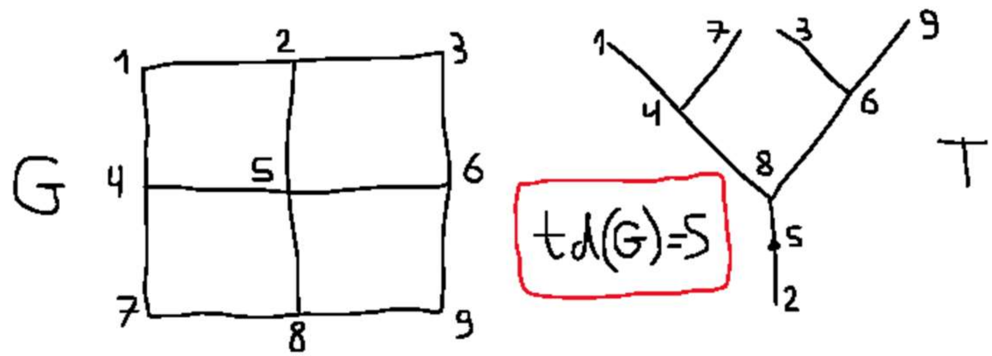


Bounded Tree Width or Depth

Theorem (Onn 2020) For any fixed k, can solve DSO in polynomial time for any functions f_i and any graph H with td(H) ≤ k

The height of a rooted tree is max number of vertices on a root-leaf path

Given a graph G on [n], a rooted tree on [n] is valid for G if for each edge ij of G, one of i and j is on the path from the root to the other.



Shmuel Onn

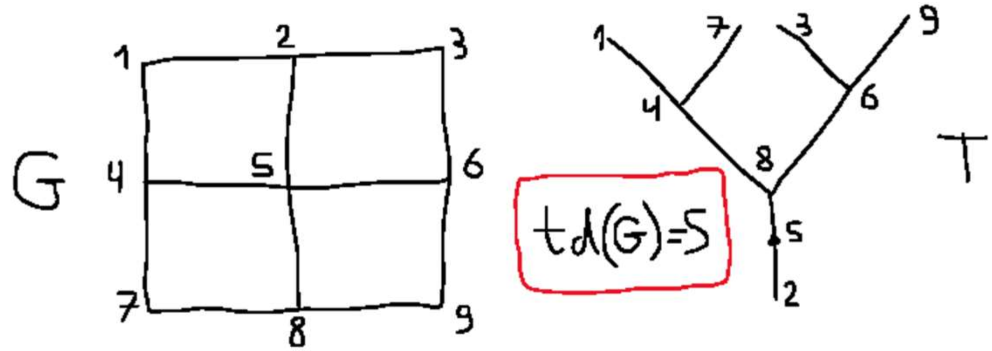
Bounded Tree Width or Depth

Theorem (Onn 2020) For any fixed k, can solve DSO in polynomial time for any functions f_i and any graph H with td(H) ≤ k

The height of a rooted tree is max number of vertices on a root-leaf path

Given a graph G on [n], a rooted tree on [n] is valid for G if for each edge ij of G, one of i and j is on the path from the root to the other.

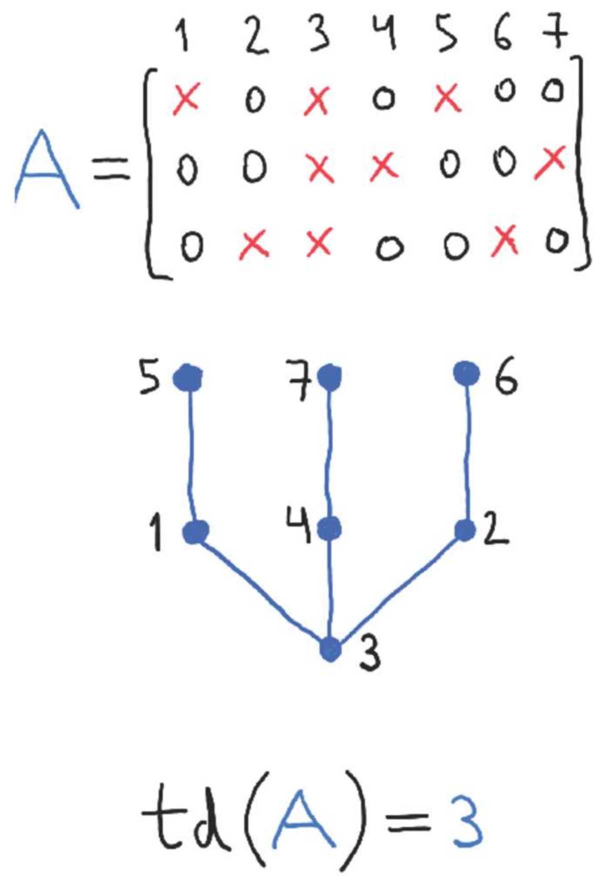
The tree-depth of G is the smallest height of a rooted tree valid for G



Sparse Integer Programming

Tree-Depth of a Matrix

The **graph** of **m×n** matrix A is the graph $G(A)$ on [n] with j,k an edge iff $A_{i,j}$, $A_{i,k}$ are nonzero for some i. The **tree-depth** of A is $td(A) := td(G(A))$



Shmuel Onn

Sparse Integer Programming

IP: $\min\{\, cx \;:\; Ax = b,\;\; l \le x \le u,\;\; x \in Z^n \,\}$

A integer $m\times n$ matrix, l,u,c in Z^n, b in Z^m

Shmuel Onn

Sparse Integer Programming

IP: $\min\{ cx \ : \ Ax = b, \ \ l \le x \le u, \ \ x \in Z^n \}$

A integer mxn matrix, l,u,c in Z^n, b in Z^m

The matrix A is parameterized by:

Numeric measure: $a:=|A|_\infty$ Sparsity measure: $d:=\min\{ td(A), td(A^T) \}$

Sparse Integer Programming

IP: $\min\{ cx : Ax = b,\ l \le x \le u,\ x \in Z^n \}$

A integer mxn matrix, l,u,c in Z^n, b in Z^m

The matrix A is parameterized by:

Numeric measure: $a := |A|_\infty$ Sparsity measure: $d := \min\{ td(A), td(A^T) \}$

Theorem (Koutecky-Levin-Onn) IP parameterized by a and d is FPT.
Specifically, there are functions $g(a,d)$, $h(d)$ so IP is solvable in time:

$g(a,d)\mathrm{poly}(n)$ when $d = td(A)$

$(a+1)^{h(d)}\mathrm{poly}(n)$ when $d = td(A^T)$

Sparse Integer Programming

IP: $\min\{\, cx \;:\; Ax = b,\;\; l \leq x \leq u,\;\; x \in Z^n \,\}$

A integer mxn matrix, l,u,c in Z^n, b in Z^m

The matrix A is parameterized by:

Numeric measure: $a := |A|_\infty$ Sparsity measure: $d := \min\{\, td(A),\, td(A^T)\,\}$

Theorem (Koutecky-Levin-Onn) IP parameterized by a and d is FPT.
Specifically, there are functions $g(a,d)$, $h(d)$ so IP is solvable in time:

$g(a,d)\text{poly}(n)$ when $d = td(A)$

$(a+1)^{h(d)}\text{poly}(n)$ when $d = td(A^T)$

Koutecky-Levin-Onn 2018, Eisenbrand-Hunkenschroder-Klein-Koutecky-Levin-Onn 2019

Koutecky-Onn 2021, "Sparse Integer Programming is FPT", Bulletin EATCS

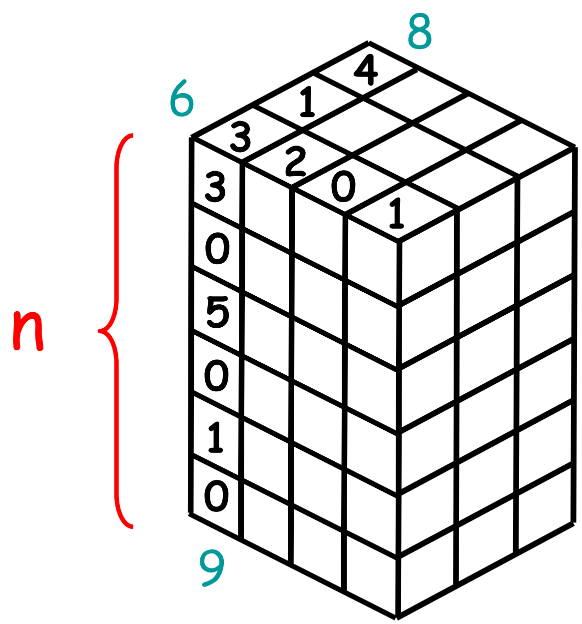
One Application: N-Fold and Block Shaped IP

$$A = \begin{pmatrix}
A_{1,1} & A_{1,2} & A_{1,3} & A_{1,4} & A_{1,5} & A_{1,6} & A_{1,7} & A_{1,8} \\
A_{2,1} & A_{2,2} & A_{2,3} & A_{2,4} & A_{2,5} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & A_{2,6} & A_{2,7} & A_{2,8} \\
A_{3,1} & A_{3,2} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & A_{3,3} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & A_{3,4} & A_{3,5} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & A_{3,6} & A_{3,7} & A_{3,8} \\
A_{4,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & A_{4,2} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & A_{4,3} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & A_{4,4} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & A_{4,5} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & A_{4,6} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & A_{4,7} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & A_{4,8}
\end{pmatrix}$$

Corollary: Block shaped IP and N-fold IP are solvable in FPT time

In Particular: Multiway Tables

Optimization in FPT time over $m_1 \times \cdots \times m_k \times n$ tables with given margins (and multicommodity flows), where this line started by De Loera – Onn:



Proof Sketch

Theorem (Koutecky-Levin-Onn) IP is FPT parameterized by a,d, where:

Numeric measure: $a := |A|_\infty$ Sparsity measure: $d := \min\{\, td(A),\ td(A^T)\,\}$

Proof Sketch

Theorem (Koutecky-Levin-Onn) IP is FPT parameterized by a,d, where:

Numeric measure: $a:=|A|_\infty$ Sparsity measure: $d:=\min\{\,td(A),\, td(A^T)\}$

1. It can be shown that the norm of vectors in the Graver basis of A (which is a subset of its integer kernel) are bounded in terms of a,d

Proof Sketch

Theorem (Koutecky-Levin-Onn) IP is FPT parameterized by a,d, where:

Numeric measure: $a:=|A|_\infty$ Sparsity measure: $d:=\min\{td(A), td(A^T)\}$

1. It can be shown that the norm of vectors in the Graver basis of A (which is a subset of its integer kernel) are bounded in terms of a,d

2. Using these bounds it can be shown that suitable auxiliary integer programs can be used to efficiently find Graver-best steps recursively on a small height tree validating small tree-depth $d=td(A)$ or $d=td(A^T)$

Proof Sketch

Theorem (Koutecky-Levin-Onn) IP is FPT parameterized by a,d, where:

Numeric measure: $a:=|A|_\infty$ Sparsity measure: $d:=\min\{td(A), td(A^T)\}$

1. It can be shown that the norm of vectors in the Graver basis of A (which is a subset of its integer kernel) are bounded in terms of a,d

2. Using these bounds it can be shown that suitable auxiliary integer programs can be used to efficiently find Graver-best steps recursively on a small height tree validating small tree-depth $d=td(A)$ or $d=td(A^T)$

3. It can be shown that few Graver-best steps suffice to reach optimum

Back to Degree Sequence Optimization

Theorem (Onn 2020) For any fixed k, can solve DSO in polynomial time for any functions f_i and any graph H with $\text{td}(H) \le k$

Back to Degree Sequence Optimization

Theorem (Onn 2020) For any fixed k, can solve DSO in polynomial time for any functions f_i and any graph H with td(H) ≤ k

Proof: The matrix A of the following IP has parameters a=n-1 and d=td(A^T)=td(H)+1 so solvable in polynomial time $n^{h(k+1)}$poly(n):

$$\min \sum_{i=1}^{n} \sum_{j=0}^{d_i(H)} f_i(j)\, y_{i,j}$$

$$\sum_{e \in \delta_i(H)} x_e - \sum_{j=0}^{d_i(H)} j\, y_{i,j} = 0$$

$$\sum_{j=0}^{d_i(H)} y_{i,j} = 1$$

$$0 \leq x_e,\ y_{i,j} \leq 1, \quad \text{integer}$$

Colored Degree Sequence Optimization

Adding suitable constraints to this IP, we can even solve the colored version of DSO, where the edges are colored by p colors and we need to find a subgraph having prescribed number of edges of each color:

Colored Degree Sequence Optimization

Adding suitable constraints to this IP, we can even solve the colored version of DSO, where the edges are colored by p colors and we need to find a subgraph having prescribed number of edges of each color:

Theorem (Onn 2020) For any fixed k,p, can solve the colored DSO in polynomial time for any functions f_i and any graph H with $td(H) \leq k$

Colored Degree Sequence Optimization

Adding suitable constraints to this IP, we can even solve the colored version of DSO, where the edges are colored by p colors and we need to find a subgraph having prescribed number of edges of each color:

Theorem (Onn 2020) For any fixed k,p, can solve the colored DSO in polynomial time for any functions f_i and any graph H with $td(H) \leq k$

A special case of this problem is the notorious exact matching problem for which a randomized algorithm is known but not a deterministic one