Integer Programming in Strongly Polynomial Oracle Time

Shmuel Onn

Technion - Israel Institute of Technology

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

joint work with Martin Koutecký and Asaf Levin, coming ICALP

Integer Programming

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

Integer Programming

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

A: integer $m \times n$ matrix

b: right-hand side in Z^m

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

w: objective in Z^n

Integer Programming

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

A: integer $m \times n$ matrix

b: right-hand side in Z^m

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

w: objective in Z^n

The bit size of the data is denoted by [A,w,b,l,u]

Outline

The Strongly Polynomial Oracle Algorithm

Outline

The Strongly Polynomial Oracle Algorithm

Effective Realization of Oracles

Outline

The Strongly Polynomial Oracle Algorithm

Effective Realization of Oracles

Programs with Block Structure

Outline

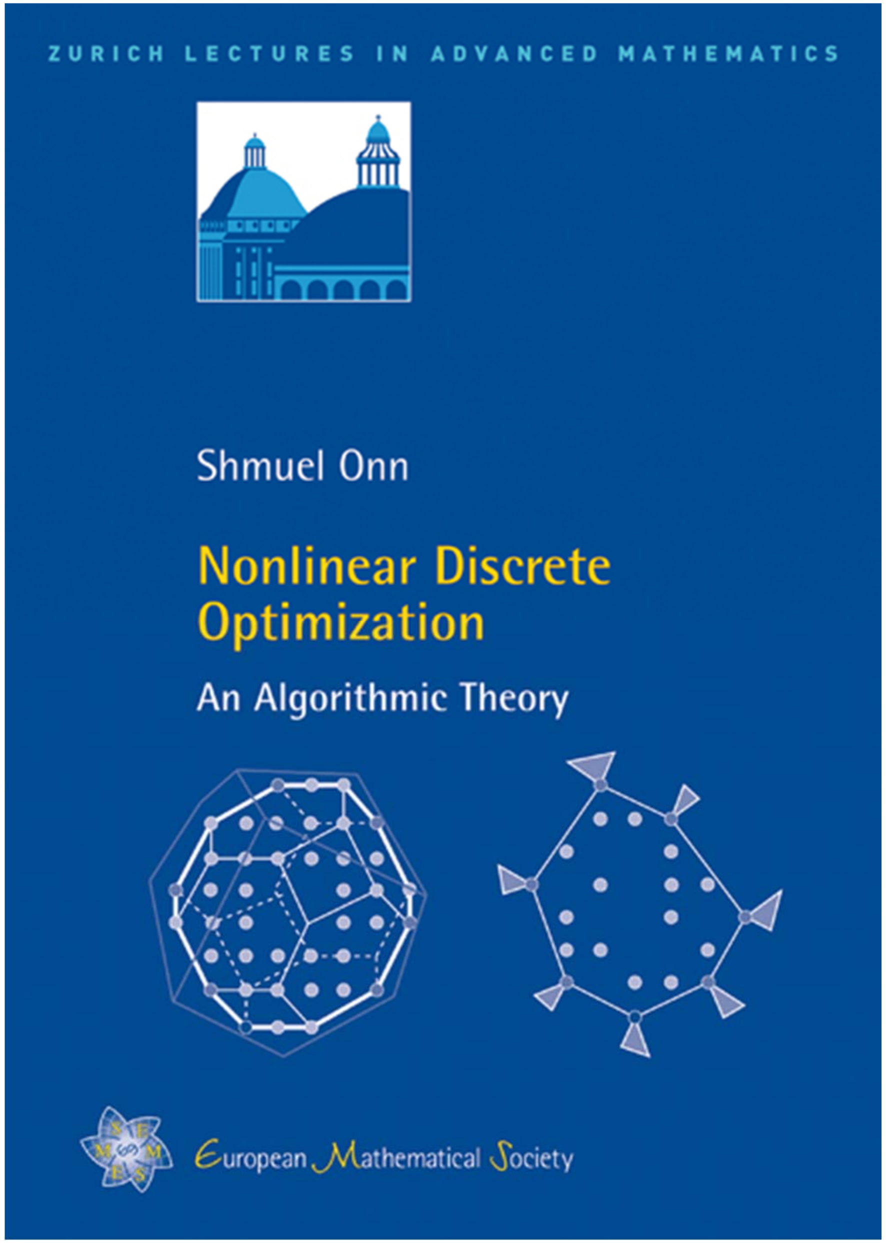
The Strongly Polynomial Oracle Algorithm

Effective Realization of Oracles

Programs with Block Structure

Programs with Bounded Tree-Depth

The Strongly Polynomial Oracle Algorithm



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

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-Best Oracle

Let x be a feasible point of the integer program

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

Graver-Best Oracle

Let x be a feasible point of the integer program

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

A Graver-best step for x is a vector h such that $x+h$ is feasible and at least as good as any feasible $x+cg$ with $c \in Z$ and $g \in G(A)$.

Graver-Best Oracle

Let x be a feasible point of the integer program

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

A Graver-best step for x is a vector h such that $x+h$ is feasible and at least as good as any feasible $x+cg$ with $c \in Z$ and $g \in G(A)$.

A Graver-best oracle for an integer matrix A is one that queried on w,b,l,u and feasible x returns a Graver-best step h for x.

Weakly Polynomial Solution

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

Lemma: IP is solvable with Graver-best oracle for A in time $O(n\,[A,w,b,l,u])$

Weakly Polynomial Solution

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

Lemma: IP is solvable with Graver-best oracle for A in time $O(n\,[A,w,b,l,u])$

(Hemmecke, Onn, Weismantel and my book Theorem 3.12)

Weakly Polynomial Solution

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

Lemma: IP is solvable with Graver-best oracle for A in time $O(n\,[A,w,b,l,u])$

(Hemmecke, Onn, Weismantel and my book Theorem 3.12)

The algorithm iteratively adds Graver-best steps till optimum is reached

Strongly Polynomial Solution

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

Lemma: IP is solvable with Graver-best oracle for A in time $O(n[A,w,b,l,u])$

Theorem 1: IP is solvable with Graver-best oracle for A in time poly($n[A]$)

Strongly Polynomial Solution

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

Lemma: IP is solvable with Graver-best oracle for A in time $O(n[A,w,b,l,u])$

Theorem 1: IP is solvable with Graver-best oracle for A in time poly($n[A]$)

(our postdoc Koutecký, Levin, Onn, coming ICALP)

Strongly Polynomial Solution

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

Lemma: IP is solvable with Graver-best oracle for A in time $O(n[A,w,b,l,u])$

Theorem 1: IP is solvable with Graver-best oracle for A in time poly($n[A]$)

Talk Wednesday morning by Martin Koutecký:
breakthroughs in computational social choice

Shmuel Onn

Strongly Polynomial Solution

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

Lemma: IP is solvable with Graver-best oracle for A in time $O(n[A,w,b,l,u])$

Theorem 1: IP is solvable with Graver-best oracle for A in time poly($n[A]$)

Proof:

1. Solve the LP-relaxation and get real optimal y^* (Tardos)
2. By proximity lemma (HS,HKW) can search for integer optimal x^* close to y^* and reduce data to $\underline{b},\underline{l},\underline{u}$ with $[\underline{b},\underline{l},\underline{u}]$ polynomial in $[A]$
3. Reduce data to $\underline{w}$ with $[\underline{w}]$ polynomial in $[A]$ (Frank-Tardos)
4. Use lemma to solve the reduced program in $O(n[A,\underline{w},\underline{b},\underline{l},\underline{u}])=\text{poly}(n[A])$

Effective Realization

of

Graver-Best Oracles

Realization of Graver-Best Oracles

The primal graph of A has columns as vertices and two columns form an edge if some row has nonzero entries in both columns.

Realization of Graver-Best Oracles

The primal graph of A has columns as vertices and two columns form an edge if some row has nonzero entries in both columns.

The dual graph of A is the primal graph of its transpose A^T

Realization of Graver-Best Oracles

The primal graph of A has columns as vertices and two columns form an edge if some row has nonzero entries in both columns.

The dual graph of A is the primal graph of its transpose A^T

Primal Lemma (KLO): If the primal graph of A has bounded tree-width and

$$\max\{\, |g|_\infty \;:\; g \in G(A) \,\}$$

is bounded then get an effective Graver-best oracle for A by DP.

Realization of Graver-Best Oracles

The primal graph of A has columns as vertices and two columns form an edge if some row has nonzero entries in both columns.

The dual graph of A is the primal graph of its transpose A^T

Primal Lemma (KLO): If the primal graph of A has bounded tree-width and

$$\max \{ \|g\|_\infty \ : \ g \in G(A) \}$$

is bounded then get an effective Graver-best oracle for A by DP.

Dual Lemma (KLO): If the dual graph of A has bounded tree-width and

$$\max \{ \|g\|_1 \ : \ g \in G(A) \}$$

is bounded then get an effective Graver-best oracle for A by DP.

Shmuel Onn

Programs with Block Structure

N-Fold Integer Programming

The n-fold product of $r \times t$ block A_1 and $s \times t$ block A_2 is

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

Parameterization of N-Fold IP

Consider n-fold integer programming over $r \times t$ block A_1 and $s \times t$ block A_2

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

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

Parameterization of N-Fold IP

Consider n-fold integer programming over $r \times t$ block A_1 and $s \times t$ block A_2

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

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

The parameters are r,s,t,a=max A

Parameterization of N-Fold IP

Consider n-fold integer programming over $r \times t$ block A_1 and $s \times t$ block A_2

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

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

The parameters are r,s,t,a=max A

The input is n and the bit size [w,b,l,u]

Complexity of N-Fold Integer Programming

-- Polynomial time $n^{f(r,s,t,a)}[w,b,l,u]$ (De Loera, Hemmecke, Onn, Weismantel)

Complexity of N-Fold Integer Programming

-- Polynomial time $n^{f(r,s,t,a)}[w,b,l,u]$ (De Loera, Hemmecke, Onn, Weismantel)

-- Parameter tractable $f(r,s,t,a)n^3[w,b,l,u]$ (Hemmecke, Onn, Romanchuk)

Led to several recent breakthroughs in the theory of parameterized complexity

Complexity of N-Fold Integer Programming

-- Polynomial time $n^{f(r,s,t,a)}[w,b,l,u]$ (De Loera, Hemmecke, Onn, Weismantel)

-- Parameter tractable $f(r,s,t,a)n^3[w,b,l,u]$ (Hemmecke, Onn, Romanchuk)

-- Strongly polynomial $n^{f(r,s,t,a)}$ (De Loera, Hemmecke, Lee)

Complexity of N-Fold Integer Programming

-- Polynomial time $n^{f(r,s,t,a)}[w,b,l,u]$ (De Loera, Hemmecke, Onn, Weismantel)

-- Parameter tractable $f(r,s,t,a)n^3[w,b,l,u]$ (Hemmecke, Onn, Romanchuk)

-- Strongly polynomial $n^{f(r,s,t,a)}$ (De Loera, Hemmecke, Lee)

Theorem 2 (Koutecký, Levin, Onn) :

Parameter tractable and strongly polynomial $f(r,s,a)\,\mathrm{poly}(n,t)$

Complexity of N-Fold Integer Programming

-- Polynomial time $n^{f(r,s,t,a)}[w,b,l,u]$ (De Loera, Hemmecke, Onn, Weismantel)

-- Parameter tractable $f(r,s,t,a)n^3[w,b,l,u]$ (Hemmecke, Onn, Romanchuk)

-- Strongly polynomial $n^{f(r,s,t,a)}$ (De Loera, Hemmecke, Lee)

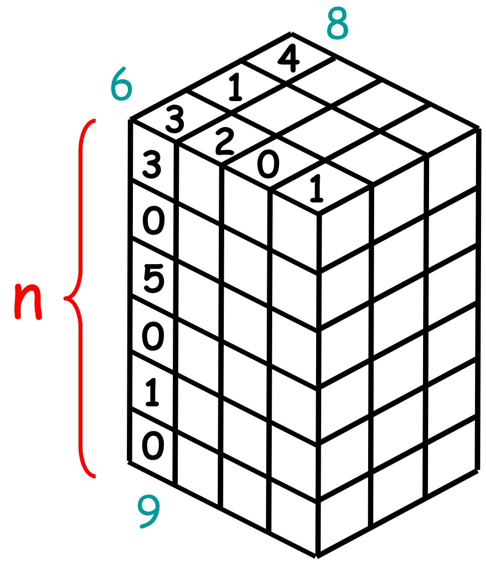
Theorem 2 (Koutecký, Levin, Onn) :

Parameter tractable and strongly polynomial $f(r,s,a)\,\mathrm{poly}(n,t)$

Proof: Use dual lemma to get parameter tractable Graver-best oracle and use Theorem 1 to get parameter tractable strongly polynomial time.

Example - Multiway Tables

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



Corollary: Parameter tractable and strongly polynomial $f(l,m)\,\mathrm{poly}(n)$

Tree-Fold Integer Programming

(Chen, Marx, SODA 2018): Tree-fold integer programs have a matrix with several blocks in tree structure, parameterized by s_i,t,a=max A

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

Tree-Fold Integer Programming

Theorem 3 (Koutecký, Levin, Onn) :

Parameter tractable and strongly polynomial $f(r,s_i,a)\,\mathrm{poly}(n,t)$

Tree-Fold Integer Programming

Theorem 3 (Koutecký, Levin, Onn) :

Parameter tractable and strongly polynomial $f(r,s_i,a)\,\mathrm{poly}(n,t)$

Proof: Use dual lemma to get parameter tractable Graver-best oracle and use Theorem 1 to get parameter tractable strongly polynomial time.

Tree-Fold Integer Programming

Theorem 3 (Koutecký, Levin, Onn) :

Parameter tractable and strongly polynomial $f(r,s_i,a)\,\mathrm{poly}(n,t)$

Proof: Use dual lemma to get parameter tractable Graver-best oracle and use Theorem 1 to get parameter tractable strongly polynomial time.

Improves on Chen, Marx, SODA 2018 - strongly polynomial and t variable

Multistage Stochastic Integer Programming

These programs have a matrix the transpose A^T of a tree-fold matrix with several blocks in tree structure, parameterized by s_i,t,a=max A_i

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

Multistage Stochastic Integer Programming

Theorem 4 (Koutecký, Levin, Onn) :

Parameter tractable and strongly polynomial $f(r,s_i,a)\,\mathrm{poly}(n,t)$

Multistage Stochastic Integer Programming

Theorem 4 (Koutecký, Levin, Onn) :

Parameter tractable and strongly polynomial $f(r,s_i,a)\,poly(n,t)$

Proof: Use primal lemma to get parameter tractable Graver-best oracle and use Theorem 1 to get parameter tractable strongly polynomial time.

Programs with Bounded Tree-Depth

Programs with Bounded Tree-Depth

Theorem 5 (Koutecký, Levin, Onn) :
Integer programs with matrix A parameterized by a=max A and by the tree-depth d of the dual graph of A can be solved in parameter tractable and strongly polynomial time $f(a,d)$ poly(n)

Programs with Bounded Tree-Depth

Theorem 5 (Koutecký, Levin, Onn) :
Integer programs with matrix A parameterized by $a=\max A$ and by the tree-depth d of the dual graph of A can be solved in parameter tractable and strongly polynomial time $f(a,d)\,\mathrm{poly}(n)$

Proof: Embed such programs in tree-fold programs and use Theorem 3

Programs with Bounded Tree-Depth

Theorem 5 (Koutecký, Levin, Onn) :
Integer programs with matrix A parameterized by $a=\max A$ and by the tree-depth d of the dual graph of A can be solved in parameter tractable and strongly polynomial time $f(a,d)\,\mathrm{poly}(n)$

Proof: Embed such programs in tree-fold programs and use Theorem 3

Theorem 6 (Koutecký, Levin, Onn) :
Integer programs with matrix A parameterized by $a=\max A$ and by the tree-depth d of the primal graph of A can be solved in parameter tractable and strongly polynomial time $f(a,d)\,\mathrm{poly}(n)$

Programs with Bounded Tree-Depth

Theorem 5 (Koutecký, Levin, Onn) :
Integer programs with matrix A parameterized by a=max A and by the tree-depth d of the dual graph of A can be solved in parameter tractable and strongly polynomial time $f(a,d)$ poly(n)

Proof: Embed such programs in tree-fold programs and use Theorem 3

Theorem 6 (Koutecký, Levin, Onn) :
Integer programs with matrix A parameterized by a=max A and by the tree-depth d of the primal graph of A can be solved in parameter tractable and strongly polynomial time $f(a,d)$ poly(n)

Proof: Embed them in multistage stochastic programs and use Theorem 4

Reference:

A Parameterized Strongly Polynomial Algorithm
for Block Structured Integer Programs

Martin Koutecký, Asaf Levin, Shmuel Onn, coming ICALP

also available at http://ie.technion.ac.il/~onn