Sparse Integer Programming is FPT

Martin Koutecký
Computer Science Institute, Charles University
koutecky@iuuk.mff.cuni.cz

Shmuel Onn
Technion – Israel Institute of Technology
onn@technion.ac.il

Abstract

We report on major progress in integer programming in variable dimension, asserting that the problem, with linear or separable-convex objective, is fixed-parameter tractable parameterized by the numeric measure and sparsity measure of the defining matrix.

Integer linear programming, with data $w, l, u \in \mathbb{Z}^n$, $A \in \mathbb{Z}^{m \times n}$, and $b \in \mathbb{Z}^m$, is the problem

$$\min\{wx \ : \ Ax = b, \ \ l \leq x \leq u, \ \ x \in \mathbb{Z}^n\} \, . \tag{1}$$

It has a very broad expressive power and numerous applications, but is generally NP-hard. A well known result [4] asserts that integer linear programming is fixed-parameter tractable (see [1]) when parameterized by the dimension (number of variables) n, but this does not help in typical situations where the dimension is large and forms a variable part of the input.

Here we report on a recent powerful result in integer programming in variable dimension, asserting that the problem is fixed-parameter tractable when parameterized by the *numeric measure* $a := \|A\|_\infty := \max_{i,j} |A_{i,j}|$ and the *sparsity measure* $d := \min\{\mathrm{td}(A), \mathrm{td}(A^T)\}$ of A. Here $\mathrm{td}(A)$ is the *tree-depth* of A, defined below, and A^T is the transpose. The result holds more generally for integer nonlinear programming where the objective function is *separable-convex*, that is, of the form $f(x) = \sum_{i=1}^n f_i(x_i)$ where each f_i is a univariate convex function which takes on integer values on integer arguments and which is given by an evaluation oracle. Below we denote by $L := \log(\|u - l\|_\infty + 1)$ the bit complexity of the lower and upper bounds, and the times are in terms of the number of arithmetic operations and oracle queries.

Theorem *The linear or separable-convex program* (2) *is fixed-parameter tractable on a, d; and if $d = \mathrm{td}(A^T)$ and is fixed, it is polynomial time solvable even if unary encoded a is variable:*

$$\min\{f(x) \ : \ Ax = b, \ \ l \leq x \leq u, \ \ x \in \mathbb{Z}^n\} \, . \tag{2}$$

More specifically, there exist computable functions h_1 and h_2 such that the following hold:

1. [3] *When $f(x) = wx$ is linear, the problem is solvable in fixed-parameter tractable time*

$$h_1(a,d)\text{poly}(n) \ \ \textit{if } d = \text{td}(A) \quad \textit{and} \quad (a+1)^{h_2(d)}\text{poly}(n) \ \ \textit{if } d = \text{td}(A^T)\ ;$$

2. [2] *When $f(x)$ is separable-convex, it is solvable in fixed-parameter tractable time*

$$h_1(a,d)\text{poly}(n)L \ \ \textit{if } d = \text{td}(A) \quad \textit{and} \quad (a+1)^{h_2(d)}\text{poly}(n)L \ \ \textit{if } d = \text{td}(A^T)\ .$$

The theorem concerns *sparse integer programming* in the sense that at least one of A and A^T has small tree-depth, a parameter which plays a central role in sparsity, see [5], and which is defined as follows. The *height* of a rooted tree is the maximum number of vertices on a path from the root to a leaf. Given a graph $G = (V, E)$, a rooted tree on V is *valid* for G if for each edge $\{j, k\} \in E$ one of j, k lies on the path from the root to the other. The *tree-depth* $\text{td}(G)$ of G is the smallest height of a rooted tree which is valid for G. The graph of an $m \times n$ matrix A is the graph $G(A)$ on $[n]$ where j, k is an edge if and only if there is an $i \in [m]$ such that $A_{i,j}A_{i,k} \neq 0$. The *tree-depth* of A is the tree-depth $\text{td}(A) := \text{td}(G(A))$ of its graph.

Here is a very rough outline of the proof. The complete details are in [2, 3].

1. Few Graver-best steps suffice. Define a partial order $\sqsubseteq$ on $\mathbb{R}^n$ by $x \sqsubseteq y$ if $x_i y_i \geq 0$ and $|x_i| \leq |y_i|$ for all i. The *Graver basis* of the integer $m \times n$ matrix A is defined to be the finite set $\mathcal{G}(A) \subset \mathbb{Z}^n$ of $\sqsubseteq$-minimal elements in $\{z \in \mathbb{Z}^n \ : \ Az = 0,\ z \neq 0\}$. Given a feasible point x in (2), a *Graver-best step* at x is a step $s \in \mathbb{Z}^n$ such that $y := x + s$ is again feasible and has objective value at least as good as any feasible $x + cz$ with $c \in \mathbb{Z}_+$ and $z \in \mathcal{G}(A)$.

It can be shown that, starting from any feasible point, an optimal point can be reached using a suitably bounded number of Graver-best steps. And, an initial feasible point can be found, or infeasibility detected, by a suitable auxiliary integer program. See [6] for details.

2. Graver norm bounds. The parametrization by $a = \|A\|_\infty$ and $d = \min\{\text{td}(A), \text{td}(A^T)\}$ of the matrix A enables to bound the norm of elements in its Graver basis $\mathcal{G}(A)$ as follows. It can be shown that there exist functions g_1 and g_2 such that, if $d = \text{td}(A)$ then $\|x\|_\infty \leq g_1(a, d)$ for all $x \in \mathcal{G}(A)$, whereas if $d = \text{td}(A^T)$ then $\|x\|_1 \leq (a + 1)^{g_2(d)}$ for all $x \in \mathcal{G}(A)$.

3. Finding Graver-best steps. Let x be a feasible point in (2) and let $c \in \mathbb{Z}_+$ be a given step size. Then a best step with step size c is a solution of one of the following auxiliary integer programs in variables z, for each of the cases $d = \text{td}(A)$ and $d = \text{td}(A^T)$ respectively,

$$\min\{f(x + cz) \ : \ Ax = 0,\ \ l \leq x + cz \leq u,\ \ \|z\|_\infty \leq g_1(a, d),\ \ z \in \mathbb{Z}^n\}\ , \tag{3}$$

$$\min\{f(x + cz) \ : \ Ax = 0,\ \ l \leq x + cz \leq u,\ \ \|z\|_1 \leq (a + 1)^{g_2(d)},\ \ z \in \mathbb{Z}^n\}\ . \tag{4}$$

Since the variables in these programs are bounded by functions of the parameters only, it can be shown that each of these programs can be solved efficiently by recursion on a suitable tree of small height, which certifies that either $d = \text{td}(A)$ or $d = \text{td}(A^T)$ respectively, is small. It can also be shown that a small list of potential step sizes $c \in \mathbb{Z}_+$ can be produced, and then the suitable program (3) or (4) is repeatedly solved for each step size in the list. Finally, the Graver-best step at x is taken to be that $s := cz$ which gives the best improvement over all.

References

[1] Cygan, M., Fomin, F.V., Kowalik, Ł., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer (2015)

[2] Eisenbrand, F., Hunkenschröder, C., Klein, K.M., Koutecký, M., Levin, A., Onn, S.: An algorithmic theory of integer programming. ArXiv:1904.01361 1–63 (2019)

[3] Koutecký, M., Levin, A., Onn, S.: A parameterized strongly polynomial algorithm for block structured integer programs. Proceedings of ICALP 2018, Leibniz International Proceedings in Informatics, 107-85:1–14 (2018)

[4] Lenstra, H.W., Jr.: Integer programming with a fixed number of variables. Mathematics of Operations Research 8:538–548 (1983)

[5] Nešetřil, J., Ossona de Mendez, P.: Sparsity: Graphs, Structures, and Algorithms. Algorithms and Combinatorics, Springer (2012)

[6] Onn, S.: Nonlinear Discrete Optimization. Zurich Lectures in Advanced Mathematics, European Mathematical Society (2010). Available at: `http://ie.technion.ac.il/~onn/Book/NDO.pdf`