Operations Research Letters 44 (2016) 635–639

Contents lists available at ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

A note on non-degenerate integer programs with small sub-determinants

S. Artmann [a], F. Eisenbrand [b], C. Glanzer [a], T. Oertel [c,*], S. Vempala [d], R. Weismantel [a]

[a] Swiss Federal Institute of Technology, Zürich (ETH Zürich), Switzerland

[b] École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

[c] Cardiff University, United Kingdom

[d] Georgia Institute of Technology, United States

ARTICLE INFO

Article history:
Received 11 March 2016
Received in revised form
30 June 2016
Accepted 5 July 2016
Available online 18 July 2016

Keywords:
Integer programming
Restricted determinants
Linear programming

ABSTRACT

The intention of this note is two-fold. First, we study integer optimization problems in standard form defined by $A \in \mathbb{Z}^{m\times n}$ and find an algorithm to solve such problems in polynomial-time provided that both the largest absolute value of an entry in A and m are constant.

Then, this is applied to solve integer programs in inequality form in polynomial-time, where the absolute values of all maximal sub-determinants of A lie between 1 and a constant.

1. Introduction

Let $A \in \mathbb{Z}^{m\times n}$ be a matrix such that all of its entries are bounded in absolute value by an integer Δ. Assume that for each row index i, $\gcd(A_{i,\cdot}) = 1$. We call the determinant of an $(n \times n)$-sub-matrix of A an $(n \times n)$-sub-determinant of A. Let

$$\delta_{\max}(A) := \max\{|d| : d \text{ is an } (n \times n)\text{-sub-determinant of } A\}.$$

We study the complexity of an integer programming problem in terms of the parameter Δ when presented in standard form (1). Moreover, we study integer programming problems in inequality form (2) that are associated with the matrix A whose 'complexity' is measured by the parameter $\delta_{\max}(A)$.

$$\max\left\{c^T x : Ax = b,\ x \geqslant 0,\ x \in \mathbb{Z}^n\right\}, \tag{1}$$

$$\max\left\{c^T x : Ax \leqslant b, x \in \mathbb{Z}^n\right\}. \tag{2}$$

In what follows, we use the Turing model when we measure time complexity.

It is known that when the absolute value of all sub-determinants of A is bounded by one, A is totally unimodular and the integer programs (1) and (2) are polynomially solvable. This concept of total unimodularity was pioneered by the works of Hoffman, Kruskal, Veinott, Dantzig and many other researchers. It has led to a beautiful and fundamental theory that is so important that it is covered by all standard textbooks in combinatorial optimization nowadays. For instance, see [8] for a thorough treatment of the subject.

When $\delta_{\max}(A) > 1$, then surprisingly little is known.

Bonifas et al. showed in [1] that for a bounded polyhedron $P = \{x \in \mathbb{R}^n : Ax \leqslant b\}$ its (combinatorial) diameter is bounded in the order of $O(\delta_{\max}(A)^2 \cdot n^{3.5} \cdot \log(n \cdot \delta_{\max}(A)))$. This improves an important result of Dyer and Frieze [2] that applies to TU-matrices.

Veselov and Chirkov (2009) showed in [9], how (2) can be solved polynomially in m and n and the encoding size of the data when $\delta_{\max}(A) \leqslant 2$ and no $(n \times n)$-sub-matrices are singular.

There exists a dynamic programming approach to solve (1) by Papadimitriou [7], see also [8], Part IV, Section 18.6: Let $\Delta(A, b)$ be an upper bound on the absolute values of A and b. Then, if (1) is feasible and bounded, it has an optimal solution with components bounded by $U := n(n+1)\Delta(A, b)(m \cdot \Delta(A, b))^m$.

The dynamic program is a maximum weight path problem on a properly defined (acyclic) graph. The number of vertices in the graph is

$$|V| = (n+1)(2U+1)^m,$$

and the number of edges is bounded by

$$|E| \leqslant (2U+1)^m |V|.$$

* Corresponding author.
E-mail address: oertelt@cardiff.ac.uk (T. Oertel).

http://dx.doi.org/10.1016/j.orl.2016.07.004
0167-6377/

Let $\lambda(m, n, \Delta(A, b))$ denote the running time of the dynamic program. As $\lambda(m, n, \Delta(A, b)) \in \mathcal{O}(|V| + |E|)$, one obtains

$$\lambda(m, n, \Delta(A, b)) \in \mathcal{O}\left(n(2U + 1)^{2m}\right).$$

We show how to avoid a dependence of the running time on the largest absolute value of an entry in b: For fixed m, an integer program can be solved in time polynomially bounded by n, the largest absolute value Δ of an entry in A, and the binary encoding size of b. This result is one important ingredient to solve the optimization problem (2) in polynomial-time for any constant values of $\delta_{\max}(A)$, provided that A has no singular $(n \times n)$-sub-matrices and $\mathrm{rank}(A) = n$. It turns out that the condition that all $(n \times n)$-sub-determinants shall be non-zero imposes very harsh restrictions on A. In particular, A can have at most $n + 1$ rows provided that n exceeds a certain constant.

2. Dynamic programming revisited

2.1. The pure integer case

We show that one can solve problem (1) in time polynomial in n, Δ and $\langle b\rangle$, where $\Delta = \max_{i,j}\{|A_{i,j}|\}$ and $\langle b\rangle = \log_2(\max_i\{|b_i|\})$. This is an improvement over Papadimitriou's approach [7], as we eliminate the unary dependency on b. For $S \subseteq \{1, \ldots, n\}$, let A_S denote the matrix stemming from A by the columns indexed by S.

Lemma 1. *If the integer program (1) is feasible and bounded, there exists an optimal solution $x^* \in \mathbb{Z}^n$ where at least $n - m$ components of x^* are bounded by $(m+2) \cdot (m \cdot \Delta)^m$. Furthermore, the columns of A corresponding to components of x^* that are larger than $(m + 2) \cdot (m \cdot \Delta)^m$ are linearly independent.*

The proof of this lemma is in the Appendix. Once this lemma is shown, we have the following result.

Theorem 2. *There exists an algorithm that solves the integer programming problem (1) in time bounded by*

$$\rho(m, n, \Delta, \langle b\rangle) \in \mathcal{O}\left(2^{\tau_1} \cdot \Delta^{\tau_2} \cdot n^{\tau_3} \cdot \tau\right),$$

where τ_1, τ_2 and τ_3 are polynomials in m and τ is a polynomial in m, n and $\langle b\rangle := \log_2(\max_i\{|b_i|\})$.

Proof. We assume that the problem is feasible and bounded. Let x^* be the optimal solution as defined in Lemma 1 and let $S \subseteq \{1, \ldots, n\}$ be the set of indices of the components of x^* that are bounded by $(m + 2) \cdot (m \cdot \Delta)^m$. By $\bar{S}$, we denote the complement of S. Now, let

$$b'' := \sum_{j\in S} x_j^* A_{\cdot,j} \quad \text{and} \quad b' := b - b''.$$

It follows that x_S^* is an optimal solution of the integer program

$$\max\left\{\sum_{j\in S} c_j x_j : \sum_{j\in S} x_j A_{\cdot,j} = b'', x \in \mathbb{Z}^S_{\geqslant 0}\right\}, \tag{3}$$

and $x_{\bar{S}}^*$ is an optimal solution of the integer program

$$\max\left\{\sum_{j\in \bar{S}} c_j x_j : \sum_{j\in \bar{S}} x_j A_{\cdot,j} = b', x \in \mathbb{Z}^{\bar{S}}_{\geqslant 0}\right\}. \tag{4}$$

Since $\|b''\|_\infty \leqslant \Delta \cdot n \cdot (m+2) \cdot (m \cdot \Delta)^m$, the integer programming problem (3) can be solved with Papadimitriou's algorithm [7] in time $\mathcal{O}(\lambda(m, n, \|b''\|_\infty))$.

Since the columns of $A_{\bar{S}}$ are linearly independent, $x_{\bar{S}}^*$ is the unique solution of the system of equations

$$\sum_{j\in \bar{S}} x_j A_{\cdot,j} = b',$$

which can be found by using Gaussian elimination.

The algorithm starts by enumerating all possible

$$\mathcal{O}\left(2^m \cdot \Delta^m \cdot n^m \cdot (m + 2)^m \cdot (m \cdot \Delta)^{m^2}\right)$$

vectors b'' and then proceeds by enumerating all $\binom{n}{m} = O(n^m)$ components of x^* whose absolute value might be larger than $(m + 2) \cdot (m \cdot \Delta)^m$ in the optimal solution x^*. Then, one solves the integer program (3) with Papadimitriou's algorithm and the integer program (4) using Gaussian elimination.

Altogether this yields a running time of

$$\mathcal{O}\left(2^m \cdot \Delta^m \cdot n^m \cdot (m + 2)^m \cdot (m \cdot \Delta)^{m^2} \cdot n^m \cdot \lambda(m, n, \|b''\|_\infty) \cdot \tau\right),$$

where τ is a polynomial in m, n and $\log(\max_i\{|b_i|\})$ which corresponds to the running time of the Gaussian elimination algorithm (cf. [4, Section 1.4]).

We can assume that $m \leqslant n$, thus eliminating all terms of the form m^{τ_*}, for τ_* a polynomial in m. This gives the desired running time. $\square$

2.2. Extensions to the mixed integer setting

This section is devoted to generalizations of Lemma 1 and Theorem 2 in order to apply the idea from the previous section to mixed-integer optimization problems of the form

$$\max\left\{c^T x + d^T y : Ax + By = b, x, y \geqslant 0, x \in \mathbb{Z}^n, y \in \mathbb{R}^l\right\}, \tag{5}$$

where, as before, $A \in \mathbb{Z}^{mn}$ with upper bound Δ on the absolute values of A, $b \in \mathbb{Z}^m$, $c \in \mathbb{Z}^n$ and $d \in \mathbb{Q}^l$.

If we view problem (5) as a parametric integer problem in variables x only, then Lemma 1 is applicable. This observation directly leads us to a mixed-integer version of Lemma 1.

Lemma 3. *If the mixed-integer program (5) has an optimal solution, then it has an optimal solution (x^*, y^*) such that $x^* \in \mathbb{Z}^n$, where at least $n - m$ components of x^* are bounded by $(m + 2) \cdot (m \cdot \Delta)^m$. Furthermore, the columns of A corresponding to components of x^* that are larger than $(m + 2) \cdot (m \cdot \Delta)^m$ are linearly independent.*

With this lemma, we are prepared to prove a mixed-integer version of Theorem 2. In the special case when m is a constant, this result gives rise to a polynomial-time algorithm for solving the mixed-integer optimization problem (5).

Theorem 4. *There exists an algorithm that solves the mixed-integer programming problem (5) in time bounded by*

$$\mathcal{O}\left(2^{\tau_1} \cdot \Delta^{\tau_2} \cdot n^{\tau_3}\right) \cdot \kappa(m, l, \Delta, \langle b\rangle),$$

where τ_1, τ_2 and τ_3 are polynomials in m, $\kappa(m, l, \Delta, \langle b\rangle)$ is the worst case running time for solving a mixed-integer optimization problem of the type (5) with at most m integer variables and l continuous variables and $\langle b\rangle := \log(\max_i\{|b_i|\})$.

Proof. Let (x^*, y^*) be an optimal solution of problem (5) satisfying Lemma 3. By $S \subseteq \{1, \ldots, n\}$ we denote the indices of the components of x^* that are bounded by $(m + 2) \cdot (m \cdot \Delta)^m$. Furthermore let

$$b'' := \sum_{j\in S} x_j^* A_{\cdot,j} \quad \text{and} \quad b' := b - b''.$$