Proximity results and faster algorithms for Integer Programming using the Steinitz Lemma

Friedrich Eisenbrand* Robert Weismantel†

Abstract

We consider integer programming problems in standard form $\max\{c^T x : Ax = b, x \geqslant 0, x \in \mathbb{Z}^n\}$ where $A \in \mathbb{Z}^{m\times n}$, $b \in \mathbb{Z}^m$ and $c \in \mathbb{Z}^n$. We show that such an integer program can be solved in time $(m\cdot\Delta)^{O(m)} \cdot \|b\|_\infty^2$, where Δ is an upper bound on each absolute value of an entry in A. This improves upon the longstanding best bound of Papadimitriou (1981) of $(m \cdot \Delta)^{O(m^2)}$, where in addition, the absolute values of the entries of b also need to be bounded by Δ. Our result relies on a lemma of Steinitz that states that a set of vectors in $\mathbb{R}^m$ that is contained in the unit ball of a norm and that sum up to zero can be ordered such that all partial sums are of norm bounded by m.

We also use the Steinitz lemma to show that the ℓ_1-distance of an optimal integer and fractional solution, also under the presence of upper bounds on the variables, is bounded by $m \cdot (2\,m \cdot \Delta + 1)^m$. Here Δ is again an upper bound on the absolute values of the entries of A. The novel strength of our bound is that it is independent of n.

We provide evidence for the significance of our bound by applying it to general knapsack problems where we obtain structural and algorithmic results that improve upon the recent literature.

1 Introduction

Many algorithmic problems, most notably problems from *combinatorial optimization* and the *geometry of numbers* can be formulated as an *integer linear program*. This is an optimization problem of the form

$$\max\{c^T x : Ax = b,\, x \geqslant 0,\, x \in \mathbb{Z}^n\} \tag{1.1}$$

where $A \in \mathbb{Z}^{m\times n}$, $b \in \mathbb{Z}^m$ and $c \in \mathbb{Z}^n$. An integer program as we describe it above is in *(equation) standard form.* Any integer program in *inequality form*, i.e., $\max\{c^T x : Ax \leqslant b,\, x \in \mathbb{Z}^n\}$ can be transformed into an integer program in standard form by duplicating variables and introducing *slack variables.* Unlike *linear programming*, integer programming is NP-complete [7].

Lenstra [19] has shown that an integer program in inequality form, with a fixed number of variables can be solved in polynomial time. A careful analysis of his algorithm shows a time bound of $2^{O(n^2)}$ times a polynomial in the length of the input that contains binary encodings of numbers. This has been improved by Kannan [18] to $2^{O(n \log n)}$ which is the best asymptotic upper bound on the exponent of 2 in 30 years. The question whether this can be improved to $2^{O(n)}$ belongs to one of the most prominent mysteries in the theory of algorithms. The current record on the constant hidden in the O-notation in the exponent is held by Dadush [11].

Papadimitriou [22] has provided an algorithm for integer programs in standard form that is, in some sense, complementary to the result of Lenstra and its improvement of Kannan. He considered the case of an integer program (1.1) in which the entries of A and b are bounded by Δ in absolute value. His algorithm is *pseudopolynomial* if m is fixed and is thus a natural generalization of pseudopolynomial time algorithms to solve unbounded knapsack problems [15].

The algorithm is based on dynamic programming and can be briefly described as follows. First, one shows that, if (1.1) is feasible and bounded, then (1.1) has an optimal solution with components bounded by $U = (n + 1)(m \cdot \Delta)^m$. The dynamic program is a maximum weight path problem on the (acyclic) graph with nodes

$$V = \{0, \dots, n\} \times \{-n \cdot \Delta \cdot U, \dots, n \cdot \Delta \cdot U\}^m$$

where one has an arc from (j, b') to $(j+1, b'')$ if $b'' - b' = k \cdot a^{(j+1)}$ for some $k \in \mathbb{N}_0$ and where $a^{(j+1)}$ is the $j+1$-st column of A. The weight of this arc is $k \cdot c_{j+1}$. The optimum solution corresponds to a longest path to the vertex (n, b). The running time of this algorithm is linear in the size of the graph. The number of nodes of this graph is bounded from below by $U^m \geqslant \Delta^{m^2}$. The upper bound on the running time in [22] is

$$O(n^{2m+2} \cdot (m\Delta)^{(m+1)(2m+1)}). \tag{1.2}$$

1.1 Contributions of this paper

We present new structural and algorithmic results concerning integer

*EPFL, Lausanne, Switzerland
†ETH Zürich, Switzerland

programs in standard form (1.1) using the Steinitz lemma, see Section 1.1 below.

a) We show that the integer program (1.1) can be solved in time
$$(m \cdot \Delta)^{O(m)} \cdot \|b\|_\infty^2$$
where Δ is an upper bound on the entries of A only. This improves upon the $(m \cdot \Delta)^{\Omega(m^2)}$ running time of the algorithm of Papadimitriou. Recall that in the setting of Papadimitriou the entries of b are bounded by Δ as well. This improvement addresses an open problem raised by Fomin et al. [13, 20].

We then consider integer programs of the form

$$\max\{c^T x : Ax = b, \, 0 \leqslant x \leqslant u, \, x \in \mathbb{Z}^n\}$$

where $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$, $u \in \mathbb{N}^n$, and $c \in \mathbb{Z}^n$ and $|a_{ij}| \leqslant \Delta$ for each i, j. Thus we allow the variables of integer program (1.1) to be bounded from above by $0 \leqslant x \leqslant u$ for some $u \in \mathbb{N}^n$. In this setting, we show the following.

b) We provide new bounds on the distance of an optimal vertex x^* of the LP-relaxation and an optimal solution of the integer program itself. More precisely, we show that there exists an optimal solution z^* of the integer program such that
$$\|z^* - x^*\|_1 \leqslant m \cdot (2 \cdot m \cdot \Delta + 1)^m$$
holds. A classical bound of Cook et al. [9] implies, in the standard-form setting, $\|z^* - x^*\|_\infty \leqslant n \cdot (\sqrt{m} \cdot \Delta)^m$ and thus $\|z^* - x^*\|_1 \leqslant n^2 \cdot (\sqrt{m} \cdot \Delta)^m$. Thus our bound is an improvement by a factor of n^2 for integer programs in standard form and fixed m.

c) We use this to generalize a recent bound on the absolute integrality gap for the case $m = 1$ by Aliev et al. [2] that states that $c^T(x^* - z^*) \leqslant \|c\|_\infty \cdot 2 \cdot \Delta$. Our distance bound shows that the absolute integrality gap is bounded by $\|c\|_\infty \cdot O(m)^{m+1} \cdot O(\Delta)^m$.

d) Our new distance bound yields an algorithm for integer programs in standard form that runs in time
$$n \cdot (\log \Delta)^2 \cdot O(m)^{m+2} \cdot O(\Delta)^{m(m+1)}.$$
For the unbounded and bounded knapsack problems where all items are of weight Δ_a at most, we obtain algorithms that run in time $O(n \cdot \Delta_a^2)$ and $O(n^2 \cdot \Delta_a^2)$ respectively. This is an improvement by a factor of n to the so far best bounds for this problem by Tamir [29].

1.2 The Steinitz lemma

Our algorithms and structural results rely on a Lemma of Steinitz [28] that we now describe. Here $\|\cdot\|$ denotes an arbitrary norm of $\mathbb{R}^m$.

Theorem 1.1. (Steinitz (1913)) *Let $x_1, \dots, x_n \in \mathbb{R}^m$ such that*
$$\sum_{i=1}^n x_i = 0 \quad \text{and} \quad \|x_i\| \leqslant 1 \text{ for each } i.$$
There exists a permutation $\pi \in S_n$ such that all partial sums satisfy
$$\|\sum_{j=1}^k x_{\pi(j)}\| \leqslant c(m) \text{ for all } k = 1, \dots, n.$$
Here $c(m)$ is a constant depending on m only.

The first explicit bounds on $c(m)$ were exponential in m see [6]. It was later shown by Sevast'anov [26, 27] that the constant $c(m) = m$ and that this is asymptotically optimal, see also [14]. The proof of the Steinitz lemma with constant $c(m) = m$ is based on LP-techniques [14] and can be quickly summarized as follows. One constructs sets $A_n \supset A_{n-1} \supset \cdots \supset A_m$ where $A_n = \{1, \dots, n\}$ and $|A_k| = k$ for each k such that the following linear system which is described by A_k with variables λ_i, $i \in A_k$ is feasible for each k:

$$\begin{array}{rcl} \sum_{i \in A_k} \lambda_i x_i & = & 0 \\ \sum_{i \in A_k} \lambda_i & = & k - m \\ 0 \leqslant \lambda_i \leqslant 1, & & i \in A_k. \end{array} \tag{1.3}$$

For any permutation π with $\{\pi(i)\} = A_i \setminus A_{i-1}$ for $i = n, \dots, m+1$ one has then for any $k \geqslant m$
$$\begin{aligned} \|\sum_{i=1}^k x_{\pi(i)}\| &= \|\sum_{i \in A_k} x_i\| = \|\sum_{i \in A_k} (1 - \lambda_i) \, x_i\| \\ &\leqslant \sum_{i \in A_k} (1 - \lambda_i) = m. \end{aligned}$$

In the inequality, we used $\|x_i\| \leqslant 1$ for each i and in the first and second equation we used (1.3). The sets A_k are constructed inductively as follows. $A_n = \{1, \dots, n\}$. If A_k has been constructed, where $k > m$, one first notes that the system (1.3) is of course also solvable if the right-hand-side $k - m$ of the second constraint is replaced by $k - 1 - m$. Once this replacement has been done, one observes that (1.3) consists of $m + 1$ equations and the inequalities $0 \leqslant \lambda \leqslant 1$. A vertex solution of (1.3) has thus at most $m + 1$ fractional entries that sum up to a value less than $m + 1$. A vertex solution of (1.3) must

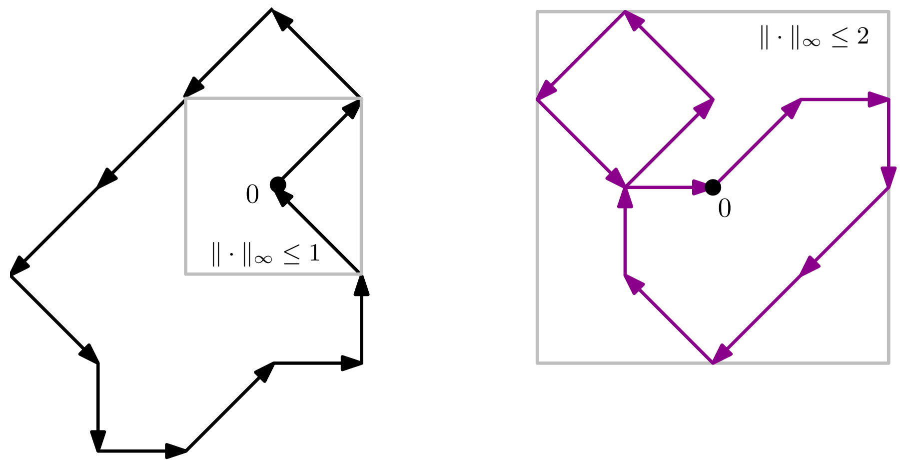

Figure 1: An example of a re-ordering satisfying the Steinitz bound for the ℓ_∞-norm. The vectors on the left have ℓ_∞-norm at most one and summ up to zero. These vectors are rearranged on the right such that the partial sums have ℓ_∞-norm bounded by 2.

therefore have one entry equal to zero. Otherwise the components of the vertex sum up to a value larger than $k-1-m$. The set A_{k-1} is now the set A_k from which the index corresponding to the zero in the vertex solution has been removed.

The reader will notice some resemblance in spirit to the proof the Beck-Fiala theorem in Discrepancy Theory [5, 21]. While techniques beyond linear programming lead to better guarantees in the Beck-Fiala setting [3, 4], the best asymptotic constant $c(m) = m$ is revealed by LP-techniques in the Steinitz setting.

We are not the first to apply the Steinitz lemma in the context of integer programming. Dash et al. [12] have shown that an integer program (1.1) can be solved in pseudopolynomial time if a certain parameter of the number of rows τ is a function of m, i.e., $\tau = \tau(m)$. The interesting aspect of their algorithm is that it relies on linear programming techniques only. The number of inequalities in their linear program is bounded by an exponential in $\tau(m)$. Buchin et al. [8] have shown that $m^{m/2-o(m)} \leqslant \tau(m) \leqslant m^{m+o(m)}$ which then yields an algorithm for integer programming that is pseudopolynomial for fixed m but doubly exponential in m. Their upper bound on $\tau(m)$ is proved via the Steinitz lemma. We take a different path in applying the Steinitz lemma. We use it to derive more efficient dynamic programming formulations directly and indirectly via new proximity results between integer and linear programming optimal solutions.

2 A faster dynamic program

We now describe a dynamic programming approach to solve (1.1) that is based on the Steinitz-type-lemma (Theorem 1.1) and which is more efficient than the original algorithm of Papadimitriou [22]. Let us first consider the feasibility problem, i.e., we have to decide whether there exists a non-negative integer vector $z^* \in \mathbb{Z}^n_{\geqslant 0}$ such that $Az^* = b$ holds. The solution z^* gives rise to a sequence of vectors $v_1, \dots, v_t$ such that each v_i is a column of A and

$$v_1 + \cdots + v_t = b. \tag{2.4}$$

The i-th column of A appears z_i^* times on the left of equation (2.4) and $t = \|z^*\|_1$. This equation can be re-written as

$$(v_1 - b/t) + \cdots + (v_t - b/t) = 0. \tag{2.5}$$

Observe that the infinity norm of each $v_i - b/t$ is at most $2\,\Delta$. The Steinitz-type-lemma implies that there exists a permutation π of the numbers $1, \dots, t$ such that all partial sums of the sequence

$$v_{\pi(1)} - b/t, \dots, v_{\pi(t)} - b/t \tag{2.6}$$

have infinity norm at most $2\,m \cdot \Delta$. In other words, for each $j \in \{1, \dots, t\}$ one has

$$\|v_{\pi(1)} + \cdots + v_{\pi(j)} - (j/t) \cdot b\|_\infty \leqslant 2\,m \cdot \Delta. \tag{2.7}$$

This implies that each partial sum of the sequence

$$v_{\pi(1)}, \dots, v_{\pi(t)}$$

is contained in the set $\mathscr{S} \subseteq \mathbb{Z}^m$ that consists of all points $x \in \mathbb{Z}^m$ for which there exists a $j \in \{1, \dots, t\}$ with

$$\|x - (j/t) \cdot b\|_\infty \leqslant 2\, m \cdot \Delta. \tag{2.8}$$

This set $\mathscr{S}$ contains at most $(4\, m \cdot \Delta + 1)^m \cdot t$ elements. The partial sums of

$$v_{\pi(1)}, \dots, v_{\pi(t)} \tag{2.9}$$

correspond to the nodes of a directed walk from 0 to b in the digraph $D = (\mathscr{S}, A)$ where one has a directed arc $xy \in A$ from $x \in \mathscr{S}$ to $y \in \mathscr{S}$ if $y - x$ is a column of A. If there exists a path from 0 to b in this digraph D on the other hand, then the arcs of the path define a multiset of columns of A summing up to b.

How fast is this approach to solve the integer feasibility problem? The number of vertices $|\mathscr{S}|$ of the digraph is equal to $(4\, m \cdot \Delta + 1)^m \cdot \|b\|_1$. The number of arcs $|A|$ is bounded by $|\mathscr{S}| \cdot n$. The integer feasibility problem is an unweighted single-source shortest path problem that can be solved with breadth-first-search in linear time [1, 10]. Consequently, the integer feasibility problem in standard form (1.1) can be solved in time

$$|\mathscr{S}| \cdot n = O(m \cdot \Delta)^m \cdot \|b\|_1 \cdot n.$$

Theorem 2.1. *Let $A \in \mathbb{Z}^{m \times n}$ and $b \in \mathbb{Z}^m$ be given and suppose that each absolute value of an entry of A is bounded by Δ. In time $O(m \cdot \Delta)^m \cdot \|b\|_1 \cdot n$ one can compute a solution of*

$$Ax = b,\, x \in \mathbb{Z}^n_{\geqslant 0}$$

or assert that such a solution does not exist.

We next describe how to tackle the optimization problem (1.1). We introduce weights on the arcs of the digraph $D = (\mathscr{S}, A)$. The weight of the arc xy is c_i if $y - x$ is the i-th column of A. Down below, we will argue that the longest path in the thereby weighted digraph from 0 to b corresponds to an optimal solution of (1.1). The longest path problem in D can be solved in time $O(|\mathscr{S}| \cdot |A|)$ with the Bellman-Ford algorithm [1]. Since

$$|A| \leqslant |\mathscr{S}| \cdot n$$

our discussion below implies that the integer program (1.1) can be solved in time $O(n \cdot |\mathscr{S}|^2)$ provided that there do not exist positive cycles reachable from 0. The next lemma clarifies that such a positive cycle exists if and only if the feasible integer program (1.1) is unbounded.

Lemma 2.1. *Suppose that* (1.1) *is feasible. The integer program* (1.1) *is unbounded if and only if D contains a cycle of strictly positive length that is reachable from* 0.

Proof. It follows from the theory of integer linear programming [25] that (1.1) is unbounded if and only if there exists an integer solution of $Ax = 0$, $x \geqslant 0$, $c^T x > 0$. Let $r^* \in \mathbb{Z}^n_{\geqslant 0}$ be such a solution. Using the Steinitz-type-lemma in the spirit of the rearrangement (2.6) but with $b = 0$, r^* corresponds to a (not necessarily simple) cycle in D of positive length starting at 0. This proves the lemma.

Remark 1. *The reader might have noticed that D contains a positive simple cycle that is reachable from* 0 *if and only if there exists a positive simple cycle in D containing* 0*. The two cycles however might not be a translation of each other.*

The algorithm to solve (1.1) is now as follows. We first check integer feasibility of (1.1). Then we run a single-source longest path algorithm from 0 to the other nodes of D, in particular to b. If the algorithm detects a cycle of positive weight, we assert that (1.1) is unbounded. Otherwise, the longest path form 0 to b corresponds to an optimal solution of (1.1). We therefore have proved the following theorem.

Theorem 2.2. *The integer program* (1.1) *can be solved in time*

$$n \cdot O(m \cdot \Delta)^{2 \cdot m} \cdot \|b\|_1^2 = (m \cdot \Delta)^{O(m)} \cdot \|b\|_\infty^2$$

where Δ is an upper bound on all absolute values of entries in A.

Remark 2. *The longest path problem runs in linear time if the digraph D does not have any cycles at all. This is for example the case when A has only non-negative entries. In this case one has a running time of $O(m \cdot \Delta)^m \cdot \|b\|_1 \cdot n$ for the integer program* (1.1)*. A well known example of such an integer program is the* configuration IP *for scheduling, see, e.g. [16, 17].*

Remark 3. *For the case in which Δ is an upper bound on the absolute values of the entries of both A and b the set $\mathscr{S}$ contains at most $(4\, m \cdot \Delta + 1)^m$ elements and the integer program* (1.1) *can be solved in time $n \cdot O(m \cdot \Delta)^{2\, m}$ and in time $n \cdot O(m \cdot \Delta)^m$ if the digraph does not have any cycles.*

3 Proximity in the ℓ_1-norm

In this section, we provide the results b) and c). From now on we consider integer programs in standard form with upper bounds on the variables, where the absolute values of A only need to be bounded by some integer Δ. In other words, we consider a problem of the form

$$\max\{c^T x : Ax = b,\, 0 \leqslant x \leqslant u,\, x \in \mathbb{Z}^n\} \tag{3.10}$$

where $A \in \mathbb{Z}^{m\times n}$, $b \in \mathbb{Z}^m$ and $c \in \mathbb{Z}^n$ and $u \in \mathbb{N}^n$ such that $|a_{ij}| \leqslant \Delta$ for each i, j. We are interested in the distance between an optimal vertex of the LP-relaxation of (3.10) and a closest integer optimum z^* in the ℓ_1-norm.

A previous bound that has been useful in many algorithmic applications, see for example [24] was shown by Cook et al. [9]. In its full generality, it is concerned with the distance in the ℓ_∞ norm in the setting of an integer program in inequality form

$$\max\{c^T x : Ax \leqslant b,\, x \in \mathbb{Z}^n\}. \tag{3.11}$$

We suppose that A and b are integral and that (3.11) is feasible and bounded. Cook et al. [9] show that for any optimal solution x^* of the linear programming relaxation there exists an optimal solution z^* of the integer program with

$$\|x^* - z^*\|_\infty \leqslant n \cdot \delta, \tag{3.12}$$

where δ is the largest absolute value of the determinant of any square submatrix of A. By the Hadamard bound, see, e.g.[25], δ is bounded by $n^{n/2} \cdot \Delta^n$, where Δ is, as before, an upper bound on the absolute values of the entries of A.

Applied to an integer program in standard form (1.1) this result implies that, for a given optimal linear solution x^* there exists an integer optimal solution z^* such that $\|z^* - x^*\|_1 \leqslant n^2\delta$. Since the Hadmard bound implies $\delta \leqslant m^{m/2}\Delta^m$

$$\|z^* - x^*\|_1 \leqslant n^2 \cdot m^{m/2}\Delta^m. \tag{3.13}$$

Using the Steinitz lemma, we show next that

$$\|z^* - x^*\|_1 \leqslant m \cdot (2 \cdot m \cdot \Delta + 1)^m.$$

We will see in a later section how this leads to algorithms for integer programs in standard form with upper bounds on the variables. In the following, let x^* and z^* be optimal solutions of the linear programming relaxation of (3.10) and of the integer program (3.10) respectively. A vector $y \in \mathbb{Z}^n$ is called a *cycle* of $(z^* - x^*)$ if $A\,y = 0$ and

$$|y_i| \leqslant |(z^* - x^*)_i| \text{ and } y_i \cdot (z^* - x^*)_i \geqslant 0 \text{ for each } i. \tag{3.14}$$

LEMMA 3.1. *Let y be a cycle of $(z^* - x^*)$, then the following assertions hold.*

i) $z^* - y$ is a feasible integer solution of (3.10).

ii) There exists an $\varepsilon > 0$ such that $x^* + \varepsilon y$ is a feasible solution of the linear programming relaxation of (3.10).

iii) One has $c^T y \leqslant 0$.

Proof. We first show i). Since $A\,y = 0$ we only need to verify that the bounds on the variables $0 \leqslant z^* - y \leqslant u$ are satisfied. If $(z^* - x^*)_i < 0$, then $y_i \leqslant 0$ and we only need to verify that the upper bound $z_i^* - y_i \leqslant u_i$ is not violated. But one has $y_i \geqslant (z^* - x^*)_i$ which is equivalent to $x_i^* \geqslant (z^* - y)_i$. Since $x^* \leqslant u$ one has $z_i^* - y_i \leqslant u_i$. The case where $(z^* - x^*)_i \geqslant 0$ follows by a similar line of argument.

To see ii) note that $y_i > 0$ implies that $z_i^* > x_i^*$ and thus x_i^* is not at the upper bound u_i. If $y_i < 0$ then $z_i^* < x_i^*$ which means that the lower bound $0 \leqslant x_i$ is not tight at x^*. Therefore, there exists an $\varepsilon > 0$ such that $x^* + \varepsilon y$ is a feasible solution of the linear program.

The assertion iii) follows from the optimality of x^* and ii).

LEMMA 3.2. *Let x^* be an optimal solution of the linear programming relaxation of* (3.10) *and let z^* be an optimal integer solution of* (3.10) *such that $\|z^* - x^*\|_1$ is minimal. There does not exist a cycle of $z^* - x^*$.*

Proof. Suppose that y is a cycle of $z^* - x^*$. By i) and iii) of Lemma 3.1, $z^* - y$ is also an optimal solution of the integer program (3.10). But $\|z^* - y - x^*\|_1 < \|z^* - x^*\|_1$ contradicting the minimality of $\|z^* - x^*\|_1$.

We are now ready to apply the Steinitz-type lemma to derive a new bound on the ℓ_1-distance between x^* and z^*.

THEOREM 3.1. *Let x^* be an optimal vertex solution of the linear programming relaxation of* (3.10)*. There exists an optimal solution z^* of the integer program* (3.10) *such that*

$$\|z^* - x^*\|_1 \leqslant m \cdot (2\,m \cdot \Delta + 1)^m.$$

Here, Δ is an upper bound on the absolute values of the entries in A.

Proof. Let z^* be an optimal integer solution such that $\|z^* - x^*\|_1$ is minimal. In the following we use the notation $\lfloor x^* \rceil$ for the vector that one obtains from x^* by rounding each component towards the corresponding component of z^*. More precisely, the i-th component of $\lfloor x^* \rceil$ is set to

$$\lfloor x^* \rceil_i = \begin{cases} \lceil x^* \rceil_i & \text{if } z_i^* > x_i^* \text{ and} \\ \lfloor x^* \rfloor_i & \text{if } z_i^* \leqslant x_i^* \end{cases}$$

and we denote the rest by $\{x^*\} = x^* - \lfloor x^* \rceil$. Clearly, one has

$$A(z^* - \lfloor x^* \rceil) - A\{x^*\} = 0. \tag{3.15}$$

We are now again in the setting of the Steinitz-lemma where we have a sequence of vectors

$$v_1, \ldots, v_t, -A\{x^*\} \tag{3.16}$$

that sum up to zero. More precisely this sequence is constructed as follows. Start with the empty sequence. For each column index i append $|(z^* - \lfloor x^* \rceil)_i|$ copies of $\mathrm{sign}((z^* - \lfloor x^* \rceil)_i) \cdot a_i$ to the list, where a_i is the i-th column of A. Finally append $-A\{x^*\}$ to the list. Since x^* has at most m positive entries, we conclude that $\| - A\{x^*\}\|_\infty \leqslant \Delta \cdot m$ and that there are vectors $w_1, \ldots, w_m$ of ℓ_∞-norm at most Δ with

$$-A\{x^*\} = w_1 + \cdots + w_m.$$

This means that the sequence of vectors (3.16) can be expanded to a sequence

$$v_1, \ldots, v_t, w_1, \ldots, w_m \tag{3.17}$$

where each vector is at most of ℓ_∞-norm Δ and that sum up to the zero vector. Observe that $t = \|z^* - \lfloor x^* \rceil\|_1$ and that $t + m \geqslant \|z^* - x^*\|_1$. The Steinitz Lemma implies that the sequence (3.17) can be re-arranged in such a way

$$u_1, \ldots, u_{t+m} \tag{3.18}$$

that for each $1 \leqslant k \leqslant t + m$ the partial sum $p_k = \sum_{i=1}^k u_i$ satisfies

$$\|p_k\|_\infty \leqslant m\Delta. \tag{3.19}$$

We will now argue that there cannot be indices $1 \leqslant k_1 < \cdots < k_{m+1} \leqslant t + m$ with

$$p_{k_1} = \cdots = p_{k_{m+1}}, \tag{3.20}$$

which implies that $t + m$ is bounded by m times the number of integer points of norm at most $m \cdot \Delta$ and therefore

$$\|z^* - x^*\|_1 \leqslant m \cdot (2 \cdot m \cdot \Delta + 1)^m.$$

Assume to this end that there exist $m + 1$ indices $1 \leqslant k_1 < \cdots < k_{m+1} \leqslant t + m$ satisfying (3.20). If there exists one index k_i such that all the vectors

$$u_{k_i+1}, \ldots, u_{k_{i+1}}$$

from the rearrangement (3.18) are columns of A or negatives thereof, then this corresponds to a cycle y of $z^* - x^*$ which, by the minimality of $\|z^* - x^*\|_1$ and Lemma 3.2 is impossible.

If such an index k_i does not exist, then all the vectors $w_1, \ldots, w_m$ appear in the sequence

$$u_{k_1+1}, \ldots, u_{k_{m+1}}.$$

This corresponds to an integer vector $y \in \mathbb{Z}^n$ such that $A(y - \{x\}) = 0$, $|(y - \{x\})_i| \leqslant |(z^* - x^*)_i|$ and $\mathrm{sign}(y_i - \{x\}_i) = \mathrm{sign}({z^*}_i - {x^*}_i)$ for each i.

This implies that $z^* - x^* - (y - \{x\})$ is a cycle of $z^* - x^*$ which is again impossible.

3.1 Integrality gaps of integer programs

Our bound of Theorem 3.1 directly leads to a bound on the *(absolute) integrality gap* of integer programs. This gap is $c^T(x^* - z^*)$ and can, via Theorem 3.1, be bounded by

$$c^T(x^* - z^*) \leqslant \|c\|_\infty \|z^* - x^*\|_1 \leqslant \|c\|_\infty m \cdot (2 \cdot m \cdot \Delta + 1)^m. \tag{3.21}$$

An integer program (1.1) is called an *unbounded knapsack problem* if m =1. In this case, Aliev et al. [2] show that one has

$$c^T(x^* - z^*) \leqslant 2 \cdot \|c\|_\infty \cdot \Delta \tag{3.22}$$

which is asymptotically our bound for $m = 1$. They derived their bound using methods from the geometry of numbers. A careful analysis of our proof in the case $m = 1$ also yields the bound (3.22) exactly. More precisely, this follows since a partial sum cannot be equal to zero. Otherwise one would have found a cycle.

4 Algorithmic implications

We now devote our attention to dynamic programming algorithms for integer programs in standard form with upper bounds on the variables and where $|a_{ij}| \leqslant \Delta$ for each i, j. This setting has received considerable attention in the approximation algorithm community, especially for scheduling problems and the respective *configuration LPs*, see for example [24, 16, 17].

Our proximity result can now be used in a dynamic programming approach to solve an integer program in standard form with upper bounds on the variables (3.10). We first compute an optimal basic solution x^* of the LP-relaxation of (3.10). In the following we denote our bound on $\|z^* - x^*\|_1$ by $L_1 = m \cdot (2 \cdot m \cdot \Delta + 1)^m$. The proof of Theorem 3.1 reveals that there exists an optimal integer solution z^* with

$$\|z^* - \lfloor x^* \rfloor\|_1 \leqslant L_1,$$

after the variable transformation $y = z - \lfloor x^* \rfloor$ one has to solve an integer program of the form

$$\begin{array}{ll} \max c^T y & \text{s.t.} \\ & A\, y = A \cdot \{x^*\} \\ & -l^* \leqslant y \leqslant u^* \\ & \|y\|_1 \leqslant L_1 \\ & y \in \mathbb{Z}^n \end{array} \tag{4.23}$$

where $l^* = \min\{L_1, \lfloor x^* \rfloor\}$ and $u^* = \min\{L_1, u - \lfloor x^* \rfloor\}$. Notice that $\|l^*\|_\infty \leqslant L_1$ and $\|u^*\|_\infty \leqslant L_1$. The potential of the new proximity bound lies in the constraint on the ℓ_1-norm in (4.23) since one has

$$\|A \cdot y\|_\infty \leqslant \Delta \cdot L_1 \tag{4.24}$$

for each $y \in \mathbb{Z}^n$ that satisfies $\|y\|_1 \leqslant L_1$. Let $U \subseteq \mathbb{Z}^m$ be the set of integer vectors of infinity norm at most $\Delta \cdot L_1$. The cardinality of U is equal to

$$|U| = (\Delta \cdot L_1)^m = O(m)^{m+1} \cdot O(\Delta)^{m\cdot(m+1)} \tag{4.25}$$

To find the optimal y^* we build the following acyclic directed graph, see Figure 4. The nodes of the graph consist of a starting node $s = 0$ and a target node $t = A \cdot \{x\}$. Furthermore, we have $n-1$ copies of the set U that we denote by $U_1, ..., U_{n-1}$. The arcs are as follows.

There is an arc from s to a node $v \in U_1$ if there exists an integer y_1 such that

$$v = y_1 \cdot a_1 \text{ and } -l_1^* \leqslant y_1 \leqslant u_1^*$$

holds. Again, a_1 denotes the first column of A. The weight of the arc is $c_1 \cdot y_1$. There is an arc from a a node $u \in U_{i-1}$ to a node $v \in U_i$ if there exists an integer y_i such that

$$v - u = y_i \cdot a_i \text{ and } -l_i^* \leqslant y_i \leqslant u_i^*$$

holds. The weight of this arc is $c_i \cdot y_i$. Finally, there is an arc from $u \in U_{n-1}$ to t of weight $y_n \cdot c_n$ if

$$A\{x^*\} - u = y_n \cdot a_n \text{ and } -l_n^* \leqslant y_n \leqslant u_n^*$$

holds for some integer y_n. Clearly, a longest path in this graph corresponds to an optimal solution y^* of the integer program (3.10). The out-degree of each node is bounded by $u_i^* + l_i^* \leqslant 2 \cdot L_1 + 1$. Therefore, the number of arcs is bounded by

$$n \cdot |U| \cdot (2 \cdot L_1 + 1) = n \cdot O(\Delta)^m \cdot O(L_1)^{m+1} \tag{4.26}$$

which would yield a running time of $n \cdot O(m)^{2\cdot(m+1)} \cdot O(\Delta)^{(m+2)m}$. However, a standard technique is to provide $M_i = O((\log L_1)^2)$ binary variables b_j^i for each variable y_i and coefficients d_j^i that are powers of two such that each integer in the interval $[-l_i^*, u_i^*]$ can be written as

$$\sum_{j=1}^{M_i} d_j^i \cdot b_j^i, \quad b_j^i \in \{0, 1\} \tag{4.27}$$

and each choice for the variables $b_j^i \in \{0,1\}$ in (4.27) represents an integer in $[-l_i^*, u_i^*]$. We now repeat the above construction but reserve $O(n \cdot (\log L_1)^2)$ copies of the set U instead of $n-1$ only. Each copy is associated to a binary variable b_j^i. We order them arbitrarily and have an arc from a node u from one copy of U to the node v of its successor of weight zero, if $u = v$ and of weight $c_i \cdot d_j^i$ if the successor copy is associated to the variable b_j^i and $v = u + a_i \cdot d_j^i$. This means that the out-degree of each node is at most two. As a result we obtain a graph with

$$n \cdot O((\log L_1)^2) \cdot |U| = nO(m)^{m+3} O(\Delta)^{m\cdot(m+1)} \log(m \cdot \Delta)$$

arcs, where we assume $\Delta \geqslant 2$. We therefore have the following result.

THEOREM 4.1. *An integer program of the form* (3.10) *can be solved in time*

$$n \cdot O(m)^{m+3} \cdot O(\Delta)^{m\cdot(m+1)} \cdot \log(m \cdot \Delta)^2$$

if each component of A is bounded by Δ in absolute value.

4.1 Faster algorithms for integer knapsack

The *bounded knapsack problem* is of the following kind.

$$\max\{c^T x : a^T x = \beta, \, 0 \leqslant x \leqslant u, \, x \in \mathbb{Z}^n\} \tag{4.28}$$

where $c, a, u \in \mathbb{Z}^n_{>0}$ and $\beta \in \mathbb{Z}_{>0}$. If the upper bound is $u = \beta \cdot \mathbf{1}$, then the knapsack problem is called *unbounded*. We let Δ_a be an upper bound on the entries of a.

Tamir [29] has shown that the unbounded and bounded knapsack problem can be solved in time $O(n^2 \Delta_a^2)$ and in time $O(n^3 \Delta_a^2)$ respectively. These running times were obtained by applying the proximity result of Cook et al. [9]. We now use our proximity bound to save a factor of n in each case.

Unbounded knapsack We begin with the unbounded knapsack problem. An optimal fractional vertex x^* has only one positive entry, x_1^* lets say and by Theorem 3.1 there exists an optimal integer solution z^* with $\|z^* - x^*\| \leqslant 2 \cdot \Delta_a + 1$. We can assume that $x_1^* \geqslant 2 \cdot \Delta_a + 1$ since otherwise $\beta = O(\Delta_a^2)$ and an $O(n \cdot \Delta_a^2)$ algorithm is obvious, see Remark 3. If y^* is an optimal solution of

$$\max\{c^T y : a^T y = (2 \cdot \Delta_a + 1) a_1, \, y \geqslant 0, \, y \in \mathbb{Z}^n\}, \tag{4.29}$$

then $(y_1^* + x_1^* - (2 \cdot \Delta_a + 1), y_2^*, \ldots, y_n^*)$ is an optimal solution of the unbounded knapsack problem. Since all entries of a and $(2 \cdot \Delta_a + 1)a_1$ are positive and bounded by $O(\Delta_a^2)$ one can solve the knapsack problem (4.29) in time $O(n \cdot \Delta_a^2)$, see again Remark 3. Consequently we have the following theorem.

THEOREM 4.2. *An unbounded knapsack problem* (4.28) *can be solved in time $O(n \cdot \Delta_a^2)$.*

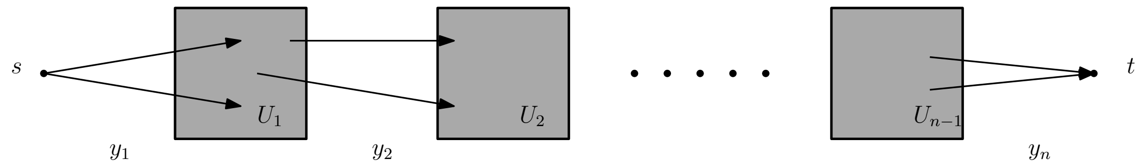

Figure 2: An illustration of the directed acyclic graph to solve the integer program (3.10).

Bounded knapsack Setting $m = 1$ in Theorem 4.1 we obtain a running time of

$$O(n \cdot (\log \Delta)^2 \cdot \Delta^2).$$

which is already an improvement over the running time of Tamir's algorithm if $\log \Delta \leqslant n$. A running time of $O(n^2 \cdot \Delta^2)$ can be obtained as follows. Again, we solve the linear programming relaxation of (4.28) and obtain an optimal vertex solution x^*. Following the notation from Section 4 we now have to solve an integer program of the form

$$\max\{c^T x : a^T x = \beta', \, -l^* \leqslant x \leqslant u^*, \, x \in \mathbb{Z}^n\}, \tag{4.30}$$

where β' is an integer with $0 \leqslant \beta' \leqslant \Delta_a$ and $\|l^*\|_\infty, \|u^*\|_\infty \leqslant 2 \cdot \Delta_a + 1$. This is equivalent to the bounded knapsack problem

$$\max\{c^T x : a^T x = \beta' + \sum_i a_i \cdot l_i^*, \, 0 \leqslant y \leqslant l_i^* + u^*, \, x \in \mathbb{Z}^n\}. \tag{4.31}$$

The new right-hand-side of this problem is $O(n \cdot \Delta_a^2)$. The bounded knapsack problem can thus be solved in time $O(n^2 \cdot \Delta_a^2)$ with an algorithm of Pferschy [23].

Acknowledgments We would like to thank Janos Pach for pointing us to the papers [27, 26]. We also thank Daniel Dadush for his very helpful comments, in particular for his hint on a variation of the Steinitz theorem that lead to stronger running-time bounds. The authors thank Martin Skutella for hosting them both at TU Berlin and EPFL for hosting the second author. The first author acknowledges support from the Swiss National Science Foundation (SNSF) within the project *Convexity, geometry of numbers, and the complexity of integer programming (Nr. 163071)*. The second author acknowledges support from the *Alexander von Humboldt Foundation*.

References

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, Reading, 1974.

[2] I. Aliev, M. Henk, and T. Oertel. Integrality gaps of integer knapsack problems. In F. Eisenbrand and J. Koenemann, editors, *Proceedings of the 19th Conference on Integer Programming and Combinatorial Optimization (IPCO 2017)*, Lecture Notes in Computer Science, pages 25 – 38. Springer, 2017.

[3] W. Banaszczyk. Balancing vectors and gaussian measures of n-dimensional convex bodies. *Random Structures & Algorithms*, 12(4):351–360, 1998.

[4] N. Bansal, D. Dadush, and S. Garg. An algorithm for Komlós conjecture matching Banaszczyk's bound. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 788–799. IEEE, 2016.

[5] J. Beck and T. Fiala. Integer-Making theorems. *Discrete Applied Mathematics*, 3(1):1–8, 1981.

[6] V. Bergström. Ein neuer Beweis eines Satzes von e. steinitz. In *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, volume 8(1), pages 148–152. Springer, 1931.

[7] I. Borosh and L. B. Treybig. Bounds on positive integral solutions of linear diophantine equations. *Proceedings of the American Mathematical Society*, 55:299–304, 1976.

[8] K. Buchin, J. Matoušek, R. A. Moser, and D. Pálvölgyi. Vectors in a box. *Mathematical programming*, 135(1-2):323–335, 2012.

[9] W. Cook, A. M. H. Gerards, A. Schrijver, and E. Tardos. Sensitivity theorems in integer linear programming. *Mathematical Programming*, 34:251 – 264, 1986.

[10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* MIT Press, Cambridge, MA, 2nd edition, 2001.

[11] D. N. Dadush. *Integer programming, lattice algorithms, and deterministic volume estimation.* PhD thesis, Georgia Institute of Technology, 2012.

[12] S. Dash, R. Fukasawa, and O. Günlük. The master equality polyhedron with multiple rows. *Mathematical programming*, 132(1-2):125–151, 2012.

[13] F. Fomin, F. Panolan, MS. Ramanujan, and S. Saurabh. Fine-grained complexity of integer programming: The case of bounded branch-width and rank. *arXiv preprint* arXiv:1607.05342, 2016.

[14] V. S. Grinberg and S. V. Sevast'yanov. Value of the Steinitz constant. *Functional Analysis and Its Applications*, 14(2):125–126, 1980.

[15] O. H. Ibarra and C. E. Kim. Fast approximation algorithms for the knapsack and sum of subset problems.

J. Assoc. Comput. Mach., 22(4):463–468, 1975.

[16] K. Jansen. An EPTAS for scheduling jobs on uniform processors: Using an milp relaxation with a constant number of integral variables. *SIAM Journal on Discrete Mathematics*, 24(2):457–485, 2010.

[17] K. Jansen, K.-M. Klein, and J. Verschae. Closing the Gap for Makespan Scheduling via Sparsification Techniques. In I. Chatzigiannakis, M. Mitzenmacher, Y. Rabani, and D. Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *LIPIcs*, pages 72:1–72:13. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016.

[18] R. Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, 1987.

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

[20] M. Lewenstein, S. Pettie, and V. V. Williams. Structure and Hardness in P (Dagstuhl Seminar 16451). *Dagstuhl Reports*, 6(11):1–34, 2017.

[21] J. Matousek. *Geometric discrepancy: An illustrated guide*, volume 18. Springer Science & Business Media, 2009.

[22] C. H. Papadimitriou. On the complexity of integer programming. *J. Assoc. Comput. Mach.*, 28(4):765–768, 1981.

[23] U. Pferschy. Dynamic programming revisited: Improving knapsack algorithms. *Computing*, 63(4):419–430, 1999.

[24] P. Sanders, N. Sivadasan, and M. Skutella. Online scheduling with bounded migration. *Mathematics of Operations Research*, 34(2):481–498, 2009.

[25] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley, 1986.

[26] S. Sevast'janov. Approximate solution of some problems of scheduling theory. *Metody Diskret. Analiz*, 32:66–75, 1978.

[27] S. Sevast'janov and W. Banaszczyk. To the Steinitz lemma in coordinate form. *Discrete Mathematics*, 169(1-3):145–152, 1997.

[28] E. Steinitz. Bedingt konvergente Reihen und konvexe Systeme. *Journal für die reine und angewandte Mathematik*, 143:128–176, 1913.

[29] A. Tamir. New pseudopolynomial complexity bounds for the bounded and other integer knapsack related problems. *Operations Research Letters*, 37(5):303–306, 2009.