From Approximate to Exact Integer Programming

Daniel Dadush[1], Friedrich Eisenbrand[2], and Thomas Rothvoss[3(✉)]

[1] CWI, Amsterdam, The Netherlands
dadush@cwi.nl
[2] EPFL, Lausanne, Switzerland
friedrich.eisenbrand@epfl.ch
[3] University of Washington, Seattle, USA
rothvoss@uw.edu

Abstract. Approximate integer programming is the following: For a given convex body $K \subseteq \mathbb{R}^n$, either determine whether $K \cap \mathbb{Z}^n$ is empty, or find an integer point in the convex body $2 \cdot (K - c) + c$ which is K, scaled by 2 from its center of gravity c. Approximate integer programming can be solved in time $2^{O(n)}$ while the fastest known methods for exact integer programming run in time $2^{O(n)} \cdot n^n$. So far, there are no efficient methods for integer programming known that are based on approximate integer programming. Our main contribution are two such methods, each yielding novel complexity results.

First, we show that an integer point $x^* \in (K \cap \mathbb{Z}^n)$ can be found in time $2^{O(n)}$, provided that the *remainders* of each component $x_i^* \mod \ell$ for some arbitrarily fixed $\ell \geq 5(n+1)$ of x^* are given. The algorithm is based on a *cutting-plane technique*, iteratively halving the volume of the feasible set. The cutting planes are determined via approximate integer programming. Enumeration of the possible remainders gives a $2^{O(n)} n^n$ algorithm for general integer programming. This matches the current best bound of an algorithm by Dadush (2012) that is considerably more involved. Our algorithm also relies on a new *asymmetric approximate Carathéodory theorem* that might be of interest on its own.

Our second method concerns integer programming problems in standard equation form $Ax = b, 0 \leq x \leq u, x \in \mathbb{Z}^n$. Such a problem can be reduced to the solution of $\prod_i O(\log u_i + 1)$ approximate integer programming problems. This implies, for example that *knapsack* or *subset-sum* problems with *polynomial variable range* $0 \leq x_i \leq p(n)$ can be solved in time $(\log n)^{O(n)}$. For these problems, the best running time so far was $n^n \cdot 2^{O(n)}$.

A full version of this paper can be found under https://arxiv.org/abs/2211.03859.

D. Dadush—Supported by ERC Starting Grant no. 805241-QIP.

F. Eisenbrand—Supported by the Swiss National Science Foundation (SNSF) grant 185030 and 207365.

T. Rothvoss—Supported by NSF CAREER grant 1651861 and a David & Lucile Packard Foundation Fellowship.

A. Del Pia and V. Kaibel (Eds.): IPCO 2023, LNCS 13904, pp. 100–114, 2023.
https://doi.org/10.1007/978-3-031-32726-1_8

1 Introduction

Many *combinatorial optimization problems* as well as many problems from the *algorithmic geometry of numbers* can be formulated as an *integer linear program*

$$\max\{\langle c, x\rangle \mid Ax \leq b, x \in \mathbb{Z}^n\} \tag{1}$$

where $A \in \mathbb{Z}^{m\times n}, b \in \mathbb{Z}^m$ and $c \in \mathbb{Z}^n$, see, e.g. [16,27,30]. Lenstra [23] has shown that integer programming can be solved in polynomial time, if the number of variables is fixed. A careful analysis of his algorithm yields a running time of $2^{O(n^2)}$ times a polynomial in the binary encoding length of the input of the integer program. Kannan [19] has improved this to $n^{O(n)}$, where, from now on we ignore the extra factor that depends polynomially on the input length. The current best algorithm is the one of Dadush [10] with a running time of $2^{O(n)} \cdot n^n$.

The question whether there exists a *singly exponential time*, i.e., a $2^{O(n)}$-algorithm for integer programming is one of the most prominent open problems in the area of algorithms and complexity. Integer programming can be described in the following more general form. Here, a *convex body* is synonymous for a full-dimensional compact and convex set.

> **Integer Programming (IP)**
> Given a *convex body* $K \subseteq \mathbb{R}^n$, find an integer solution $x^* \in K \cap \mathbb{Z}^n$ or assert that $K \cap \mathbb{Z}^n = \emptyset$.

The convex body K must be well described in the sense that there is access to a *separation oracle*, see [16]. Furthermore, one assumes that K contains a ball of radius $r > 0$ and that it is contained in some ball of radius R. In this setting, the current best running times hold as well. The additional polynomial factor in the input encoding length becomes a polynomial factor in $\log(R/r)$ and the dimension n. Central to this paper is *Approximate integer programming* which is as follows.

> **Approximate Integer Programming (Approx-IP)**
> Given a *convex body* $K \subseteq \mathbb{R}^n$, let $c \in \mathbb{R}^n$ be its center of gravity. Either find an integer vector $x^* \in (2\cdot(K-c)+c) \cap \mathbb{Z}^n$, or assert that $K \cap \mathbb{Z}^n = \emptyset$.

The convex body $2\cdot(K-c)+c$ is K scaled by a factor of 2 from its center of gravity. The algorithm of Dadush [11] solves approximate integer programming in singly exponential time $2^{O(n)}$. Despite its clear relation to exact integer programming, there is no reduction from exact to approximate known so far. Our guiding question is the following: Can approximate integer programming be used to solve the exact version of (specific) integer programming problems?

1.1 Contributions of This Paper

We present two different algorithms to reduce the exact integer programming problem (IP) to the approximate version (APPROX-IP).

a) Our first method is a randomized cutting-plane algorithm that, in time $2^{O(n)}$ and for any $\ell \geq 5(n+1)$ finds a point in $K \cap (\mathbb{Z}^n/\ell)$ with high probability, if K contains an integer point. This algorithm uses an oracle for (APPROX-IP) on K intersected with one side of a hyperplane that is close to the center of gravity. Thereby, the algorithm collects ℓ integer points close to K. The collection is such that the convex combination with uniform weights $1/\ell$ of these points lies in K. If, during an iteration, no point is found, the volume of K is roughly halved and eventually K lies on a lower-dimensional subspace on which one can recurse.

b) If equipped with the component-wise remainders $v \equiv x^* \pmod{\ell}$ of a solution x^* of (IP), one can use the algorithm to find a point in $(K - v) \cap \mathbb{Z}^n$ and combine it with the remainders to a full solution of (IP), using that $(K - v) \cap \ell\mathbb{Z}^n \neq \emptyset$. This runs in singly exponential randomized time $2^{O(n)}$. Via enumeration of all remainders, one obtains an algorithm for (IP) that runs in time $2^{O(n)} \cdot n^n$. This matches the best-known running time for general integer programming [11], which is considerably involved.

c) Our analysis depends on a new *approximate Carathéodory theorem* that we develop in Sect. 4. While approximate Carathéodory theorems are known for centrally symmetric convex bodies [4,26,28], our version is for general convex sets and might be of interest on its own.

d) Our second method is for integer programming problems $Ax = b$, $x \in \mathbb{Z}^n$, $0 \leq x \leq u$ in equation standard form. We show that such a problem can be reduced to $2^{O(n)} \cdot (\prod_i \log(u_i + 1))$ instances of (APPROX-IP). This yields a running time of $(\log n)^{O(n)}$ for such IPs, in which the variables are bounded by a polynomial in the dimension. The so-far best running time for such instances $2^{O(n)} \cdot n^n$. Well known benchmark problems in this setting are *knapsack* and *subset-sum* with polynomial upper bounds on the variables, see Sect. 5.

1.2 Related Work

If the convex body K is an ellipsoid, then the integer programming problem (IP) is the well known *closest vector problem (CVP)* which can be solved in time $2^{O(n)}$ with an algorithm by Micciancio and Voulgaris [25]. Blömer and Naewe [7] previously observed that the sampling technique of Ajtai et al. [1] can be modified in such a way as to solve the closest vector approximately. More precisely, they showed that a $(1+\epsilon)$-approximation of the closest vector problem can be found in time $O(2+1/\epsilon)^n$ time. This was later generalized to arbitrary convex sets by Dadush [11]. This algorithm either asserts that the convex body K does not contain any integer points, or it finds an integer point in the body stemming from K is scaled by $(1+\epsilon)$ from its center of gravity. Also the running time of this randomized algorithm is $O(2+1/\epsilon)^n$. In our paper, we restrict to the case $\epsilon = 1$ which can be solved in singly exponential time. The technique of reflection sets was also used by Eisenbrand et al. [13] to solve (CVP) in the ℓ_∞-norm approximately in time $O(2+\log(1/\epsilon))^n$.

In the setting in which integer programming can be attacked with dynamic programming, tight upper and lower bounds on the complexity are known [14,17,20]. Our $n^n \cdot 2^{O(n)}$ algorithm could be made more efficient by constraining the possible remainders of a solution $\pmod{\ell}$ efficiently. This barrier is different than the one in classical integer-programming methods that are based on branching on flat directions [16,23] as they result in a branching tree of size $n^{O(n)}$.

The *subset-sum problem* is as follows. Given a set $Z \subseteq \mathbb{N}$ of n positive integers and a *target value* $t \in \mathbb{N}$, determine whether there exists a subset $S \subseteq Z$ with $\sum_{s \in S} s = t$. Subset sum is a classical NP-complete problem that serves as a benchmark in algorithm design. The problem can be solved in pseudopolynomial time [5] by dynamic programming. The current fastest pseudopolynomial-time algorithm is the one of Bringmann [8] that runs in time $O(n+t)$ up to polylogarithmic factors. There exist instances of subset-sum whose set of feasible solutions, interpreted as 0/1 incidence vectors, require numbers of value n^n in the input, see [2]. Lagarias and Odlyzko [21] have shown that instances of subset sum in which each number of the input Z is drawn uniformly at random from $\{1,\dots,2^{O(n^2)}\}$ can be solved in polynomial time with high probability. The algorithm of Lagarias and Odlyzko is based on the LLL-algorithm [22] for lattice basis reduction.

2 Preliminaries

A *lattice* Λ is the set of integer combinations of linearly independent vectors, i.e. $\Lambda := \Lambda(B) := \{Bx \mid x \in \mathbb{Z}^r\}$ where $B \in \mathbb{R}^{n\times r}$ has linearly independent columns. The *determinant* is the volume of the r-dimensional parallelepiped spanned by the columns of the basis B, i.e. $\det(\Lambda) := \sqrt{\det_r(B^T B)}$. We say that Λ has *full rank* if $n = r$. In that case the determinant is simply $\det(\Lambda) = |\det_n(B)|$. For a full rank lattice Λ, we denote the dual lattice by $\Lambda^* = \{y \in \mathbb{R}^n \mid \langle x, y\rangle \in \mathbb{Z}\ \forall x \in \Lambda\}$. Note that $\det(\Lambda^*)\cdot\det(\Lambda) = 1$. For an introduction to lattices, we refer to [24].

A set $Q \subseteq \mathbb{R}^n$ is called a *convex body* if it is convex, compact and has a non-empty interior. A set Q is *symmetric* if $Q = -Q$. Recall that any symmetric convex body Q naturally induces a norm $\|\cdot\|_Q$ of the form $\|x\|_Q = \min\{s \geq 0 \mid x \in sQ\}$. For a full rank lattice $\Lambda \subseteq \mathbb{R}^n$ and a symmetric convex body $Q \subseteq \mathbb{R}^n$ we denote $\lambda_1(\Lambda, Q) := \min\{\|x\|_Q \mid x \in \Lambda \setminus \{\mathbf{0}\}\}$ as the length of the shortest vector with respect to the norm induced by Q. We denote the Euclidean ball by $B_2^n := \{x \in \mathbb{R}^n \mid \|x\|_2 \leq 1\}$ and the ℓ_∞-ball by $B_\infty^n := [-1,1]^n$. An (origin centered) *ellipsoid* is of the form $\mathcal{E} = A(B_2^n)$ where $A : \mathbb{R}^n \to \mathbb{R}^n$ is an invertible linear map. For any such ellipsoid $\mathcal{E}$ there is a unique positive definite matrix $M \in \mathbb{R}^{n\times n}$ so that $\|x\|_{\mathcal{E}} = \sqrt{x^T M x}$. The *barycenter* (or *centroid*) of a convex body Q is the point $\frac{1}{\mathrm{Vol}_n(Q)} \int_Q x\, dx$. We will use the following version of (APPROX-IP) that runs in time $2^{O(n)}$, provided that the symmetrizer for the used center c is large enough. This is the case for c being the center of gravity, see Theorem 3. Note that the center of gravity of a convex body can be (approximately) computed in randomized polynomial time [6,12].

Theorem 1 (Dadush [11]). *There is a $2^{O(n)}$-time algorithm* APXIP(K, c, Λ) *that takes as input a convex set $K \subseteq \mathbb{R}^n$, a point $c \in K$ and a lattice $\Lambda \subseteq \mathbb{R}^n$. Assuming that $Vol_n((K-c)\cap(c-K)) \geq 2^{-\Theta(n)} Vol_n(K)$ the algorithm either returns a point $x \in (c + 2(K-c)) \cap \Lambda$ or returns* EMPTY *if $K \cap \Lambda = \emptyset$.*

One of the classical results in the geometry of numbers is Minkowski's Theorem which we will use in the following form:

Theorem 2 (Minkowski's Theorem). *For a full rank lattice $\Lambda \subseteq \mathbb{R}^n$ and a symmetric convex body $Q \subseteq \mathbb{R}^n$ one has*

$$\lambda_1(\Lambda, Q) \leq 2 \cdot \left(\frac{\det(\Lambda)}{Vol_n(Q)}\right)^{1/n}$$

We will use the following bound on the density of sublattices which is an immediate consequence of Minkowski's Second Theorem. Here we abbreviate $\lambda_1(\Lambda) := \lambda_1(\Lambda, B_2^n)$.

Lemma 1. *Let $\Lambda \subseteq \mathbb{R}^n$ be a full rank lattice. Then for any k-dimensional sublattice $\tilde{\Lambda} \subseteq \Lambda$ one has* $\det(\tilde{\Lambda}) \geq (\frac{\lambda_1(\Lambda)}{\sqrt{k}})^k$.

Finally, we revisit a few facts from *convex geometry*. Details and proofs can be found in the excellent textbook by Artstein-Avidan, Giannopoulos and Milman [3].

Lemma 2 (Grünbaum's Lemma). *Let $K \subseteq \mathbb{R}^n$ be any convex body and let $\langle a, x\rangle = \beta$ be any hyperplane through the barycenter of K. Then $\frac{1}{e} Vol_n(K) \leq Vol_n(\{x \in K \mid \langle a, x\rangle \leq \beta\}) \leq (1 - \frac{1}{e}) Vol_n(K)$.*

For a convex body K, there are two natural symmetric convex bodies that approximate K in many ways: the "inner symmetrizer" $K \cap (-K)$ (provided $\mathbf{0} \in K$) and the "outer symmetrizer" in form of the *difference body* $K - K$. The following is a consequence of a more general inequality of Milman and Pajor.

Theorem 3. *Let $K \subseteq \mathbb{R}^n$ be any convex body with barycenter $\mathbf{0}$. Then $Vol_n(K \cap (-K)) \geq 2^{-n} Vol_n(K)$.*

In particular Theorem 3 implies that choosing c as the barycenter of K in Theorem 1 results in a $2^{O(n)}$ running time—however this will not be the choice that we will later make for c. Also the size of the difference body can be bounded:

Theorem 4 (Inequality of Rogers and Shephard). *For any convex body $K \subseteq \mathbb{R}^n$ one has $Vol_n(K - K) \leq 4^n Vol_n(K)$.*

Recall that for a convex body Q with $\mathbf{0} \in \text{int}(Q)$, the *polar* is $Q^\circ = \{y \in \mathbb{R}^n \mid \langle x, y\rangle \leq 1 \ \forall x \in Q\}$. We will use the following relation between volume of a symmetric convex body and the volume of the polar; to be precise we will use the lower bound (which is due to Bourgain and Milman).

Theorem 5 (Blaschke-Santaló-Bourgain-Milman). *For any symmetric convex body $Q \subseteq \mathbb{R}^n$ one has*

$$C^n \leq \frac{Vol_n(Q) \cdot Vol_n(Q^\circ)}{Vol_n(B_2^n)^2} \leq 1$$

where $C > 0$ is a universal constant.

We will also rely on the result of Frank and Tardos to reduce the bit complexity of constraints:

Theorem 6 (Frank, Tardos [15]). *There is a polynomial time algorithm that takes $(a, b) \in \mathbb{Q}^{n+1}$ and $\Delta \in \mathbb{N}_+$ as input and produces a pair $(\tilde{a}, \tilde{b}) \in \mathbb{Z}^{n+1}$ with $\|\tilde{a}\|_\infty, |\tilde{b}| \leq 2^{O(n^3)} \cdot \Delta^{O(n^2)}$ so that $\langle a, x\rangle = b \Leftrightarrow \langle \tilde{a}, x\rangle = \tilde{b}$ and $\langle a, x\rangle \leq b \Leftrightarrow \langle \tilde{a}, x\rangle \leq \tilde{b}$ for all $x \in \{-\Delta, \ldots, \Delta\}^n$.*

3 The Cut-Or-Average Algorithm

First, we discuss our CUT-OR-AVERAGE algorithm that on input of a convex set K, a lattice Λ and integer $\ell \geq 5(n+1)$, either finds a point $x \in \frac{\Lambda}{\ell} \cap K$ or decides that $K \cap \Lambda = \emptyset$ in time $2^{O(n)}$. Note that for any polyhedron $K = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ with rational A, b and lattice Λ with basis B one can compute a value of Δ so that $\log(\Delta)$ is polynomial in the encoding length of A, b and B and $K \cap \Lambda \neq \emptyset$ if and only if $K \cap [-\Delta, \Delta]^n \cap \Lambda \neq \emptyset$. See Schrijver [31] for details. In other words, w.l.o.g. we may assume that our convex set is bounded. The pseudo code of the algorithm can be found in Fig. 1. An intuitive description of the algorithm is as follows: we compute the barycenter c of K and an ellipsoid $\mathcal{E}$ that approximates K up to a factor of $R = n+1$. Then we iteratively use the oracle for approximate integer programming from Theorem 1 to find a convex combination z of lattice points in a 3-scaling of K until z is close to the barycenter c. If this succeeds, then we can directly use an asymmetric version of the *Approximate Carathéodory Theorem* (Lemma 9) to find an unweighted average of ℓ lattice points that lies in K; this would be a vector of the form $x \in \frac{\Lambda}{\ell} \cap K$. If the algorithm fails to approximately express c as a convex combination of lattice points, then we will have found a hyperplane H going almost through the barycenter c so that $K \cap H_{\geq}$ does not contain a lattice point. Then the algorithm continues searching in $K \cap H_{\leq}$ (Fig. 2). This case might happen repeatedly, but after polynomial number of times, the volume of K will have dropped below a threshold so that we may recurse on a *single* $(n-1)$-dimensional subproblem. We will now give the detailed analysis. Note that in order to obtain a clean exposition we did not aim to optimize any constant. However by merely tweaking the parameters one could make the choice of $\ell = (1+\varepsilon)n$ work for any constant $\varepsilon > 0$.

3.1 Bounding the Number of Iterations

We begin the analysis with a few estimates that will help us to bound the number of iterations.

Lemma 3. *Any point x found in line (7) lies in a 3-scaling of K around c, i.e. $x \in c + 3(K-c)$ assuming $0 < \rho \leq 1$.*

Proof. We verify that

$$x \in (c - \rho d) + 2(K - (c - \rho d)) = c + 2(K-c) + \rho d \subseteq c + 3(K-c)$$

using that $\|\rho d\|_{\mathcal{E}} = \rho \leq 1$.

Next we bound the distance of z to the barycenter:

Lemma 4. *At the beginning of the kth iterations of the WHILE loop on line (5), one has $\|c - z\|_{\mathcal{E}}^2 \leq \frac{9R^2}{k}$.*

Proof. We prove the statement by induction on k. At $k = 1$, by construction on line (4), $z \in c + 2(K-c) \subseteq c + 2R\mathcal{E}$. Thus $\|c - z\|_{\mathcal{E}}^2 \leq (2R)^2 \leq 9R^2$, as needed.

Input: Convex set $K \subseteq \mathbb{R}^n$, lattice Λ, parameter $\ell \geq 5(n+1)$
Output: Either a point $x \in K \cap \frac{\Lambda}{\ell}$ or conclusion that $K \cap \Lambda = \emptyset$

(1) WHILE $\lambda_1(\Lambda^*, (K-K)^\circ) > \frac{1}{2}$ DO
  (2) Compute barycenter c of K.
  (3) Let $\mathcal{E} := \{x \in \mathbb{R}^n \mid x^T M x \leq 1\}$ be $\mathbf{0}$-centered ellipsoid with $c + \mathcal{E} \subseteq K \subseteq c + R \cdot \mathcal{E}$ for $R := n+1$, let $\rho := \frac{1}{4n}$.
  (4) Let $z := \text{APXIP}(K, c, \Lambda)$, $X = \{z\}$. If $z = \text{EMPTY}$, Return EMPTY.
  (5) WHILE $\|c - z\|_{\mathcal{E}} > \frac{1}{4}$ DO
    (6) Let $d := \frac{-(z-c)}{\|z-c\|_{\mathcal{E}}}$, $a := -M(z-c)$.
    (7) Compute $x := \text{APXIP}(K \cap \{x \in \mathbb{R}^n \mid \langle a, x\rangle \geq \langle a, c + \rho d/2\rangle\}, c + \rho d, \Lambda)$.
    (8) IF $x = \text{EMPTY}$ THEN replace K by $K' := K \cap \{x \in \mathbb{R}^n \mid \langle a, x\rangle \leq \langle a, c + \rho d/2\rangle\}$ and GOTO (1).
    (9) ELSE $X := X \cup \{x\}$, $z := (1 - \frac{1}{|X|})z + \frac{x}{|X|}$.
  (10) Compute $\mu \in \frac{\mathbb{Z}^X_{\geq 0}}{\ell}$ with $\sum_{x \in X} \mu_x = 1$ and $\sum_{x\in X} \mu_x x \in K$ using Asymmetric Approximate Carathéodory.
  (11) Return $\sum_{x \in X} \mu_x x$.
(12) Compute $y \in \Lambda^* \setminus \{\mathbf{0}\}$ with $\|y\|_{(K-K)^\circ} \leq \frac{1}{2}$.
(13) Find $\beta \in \mathbb{Z}$ so that $K \cap \Lambda \subseteq U$ with $U = \{x \in \mathbb{R}^n \mid \langle y, x\rangle = \beta\}$.
(14) IF $n = 1$ THEN $U = \{x^*\}$; Return x^* if $x^* \in \Lambda$ and return "$K \cap \Lambda = \emptyset$" otherwise.
(15) Recurse on $(n-1)$-dim. instance CUT-OR-AVERAGE$(K \cap U, \Lambda \cap U, \ell)$.

Fig. 1. The Cut-Or-Average algorithm.

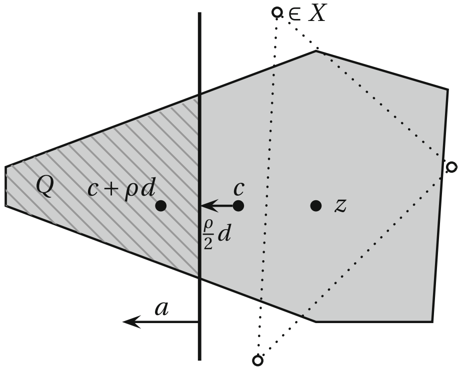

Fig. 2. Visualization of the inner WHILE loop where $Q := K \cap \{x \in \mathbb{R}^n \mid \langle a, x\rangle \geq \left\langle a, c + \frac{\rho}{2} d\right\rangle\}$.

Now assume $k \geq 2$. Let z, z' denote the values of z during iteration $k-1$ before and after the execution of line (9) respectively, and let x be the vector found on line (7) during iteration $k-1$. Note that $z' = (1-\frac{1}{k})z + \frac{1}{k}x$. By the induction hypothesis, we have that $\|z-c\|^2_{\mathcal{E}} \leq 9R^2/(k-1)$. Our goal is to show that $\|z'-c\|^2_{\mathcal{E}} \leq 9R^2/k$. Letting d denote the normalized version of $z-c$, we see that $\|d\|_{\mathcal{E}} = 1$ and hence $d \in K-c$. By construction $\langle a, x-c\rangle \geq 0$ and from Lemma 3 we have $x \in c + 3(K-c)$ which implies

$\|x-c\|_{\mathcal{E}} \leq 3R$. The desired bound on the $\mathcal{E}$-norm of $z'-c$ follows from the following calculation:

$$\begin{aligned}
\|z'-c\|_{\mathcal{E}}^2 &= \Big\|\Big(1-\frac{1}{k}\Big)(z-c)+\frac{1}{k}(x-c)\Big\|_{\mathcal{E}}^2 \\
&= \Big(1-\frac{1}{k}\Big)^2\|z-c\|_{\mathcal{E}}^2 - 2\Big(1-\frac{1}{k}\Big)\frac{1}{k}\langle a, x-c\rangle + \frac{1}{k^2}\|x-c\|_{\mathcal{E}}^2 \\
&\leq \Big(1-\frac{1}{k}\Big)^2\|z-c\|_{\mathcal{E}}^2 + \frac{1}{k^2}\|x-c\|_{\mathcal{E}}^2 \\
&\leq \Big(\Big(1-\frac{1}{k}\Big)^2\frac{1}{k-1}+\frac{1}{k^2}\Big)\cdot 9R^2 = \frac{9R^2}{k}.
\end{aligned}$$

In particular Lemma 4 implies an upper bound on the number of iterations of the inner WHILE loop:

Corollary 1. *The WHILE loop on line (5) never takes more than* $36R^2$ *iterations.*

Proof. By Lemma 4, for $k := 36R^2$ one has $\|c-z\|_{\mathcal{E}}^2 \leq \frac{9R^2}{k} \leq \frac{1}{4}$.

Next, we prove that every time we replace K by $K' \subset K$ in line (8), its volume drops by a constant factor.

Lemma 5. *In step (8) one has* $Vol_n(K') \leq (1-\frac{1}{e})\cdot(1+\frac{\rho}{2})^n \cdot Vol_n(K)$ *for any* $\rho \geq 0$. *In particular for* $0 \leq \rho \leq \frac{1}{4n}$ *one has* $Vol_n(K') \leq \frac{3}{4} Vol_n(K)$.

Proof. The claim is invariant under affine linear transformations, hence we may assume w.l.o.g. that $\mathcal{E} = B_2^n$, $M = I_n$ and $c = \mathbf{0}$. Note that then $B_2^n \subseteq K \subseteq RB_2^n$. Let us abbreviate $K_{\leq t} := \{x \in K \mid \langle d, x\rangle \leq t\}$. In this notation $K' = K_{\leq \rho/2}$. Recall that Grünbaum's Lemma (Lemma 2) guarantees that $\frac{1}{e} \leq \frac{\mathrm{Vol}_n(K_{\leq 0})}{\mathrm{Vol}_n(K)} \leq 1-\frac{1}{e}$. Moreover, it is well known that the function $t \mapsto \mathrm{Vol}_n(K_{\leq t})^{1/n}$ is concave on its support, see again [3]. Then

$$\begin{aligned}
\mathrm{Vol}_n(K_{\leq 0})^{1/n} &\geq \Big(\frac{1}{1+\rho/2}\Big)\cdot \mathrm{Vol}_n(K_{\leq \rho/2})^{1/n} + \Big(\frac{\rho/2}{1+\rho/2}\Big)\cdot \underbrace{\mathrm{Vol}_n(K_{\leq -1})^{1/n}}_{\geq 0} \\
&\geq \Big(\frac{1}{1+\rho/2}\Big)\cdot \mathrm{Vol}_n(K_{\leq \rho/2})^{1/n}
\end{aligned}$$

and so

$$\Big(1-\frac{1}{e}\Big)\cdot \mathrm{Vol}_n(K) \geq \mathrm{Vol}_n(K_{\leq 0}) \geq \Big(\frac{1}{1+\rho/2}\Big)^n \cdot \mathrm{Vol}_n(K_{\leq \rho/2})$$

Rearranging gives the first claim in the form $\mathrm{Vol}_n(K_{\leq \rho/2}) \leq (1-\frac{1}{e})\cdot(1+\frac{\rho}{2})^n \cdot \mathrm{Vol}_n(K)$. For the 2nd part we verify that for $\rho \leq \frac{1}{4n}$ one has $(1-\frac{1}{e})\cdot(1+\frac{\rho}{2})^n \leq (1-\frac{1}{e})\cdot\exp(\frac{\rho}{2}) \leq \frac{3}{4}$.

Lemma 6. *Consider a call of* CUT-OR-AVERAGE *on* (K,Λ) *where* $K \subseteq rB_2^n$ *for some* $r > 0$. *Then the total number of iterations of the outer WHILE loop over all recursion levels is bounded by* $O(n^2 \log(\frac{nr}{\lambda_1(\Lambda)}))$.

Proof. Consider any recursive run of the algorithm. The convex set will be of the form $\tilde{K} := K \cap U$ and the lattice will be of the form $\tilde{\Lambda} := \Lambda \cap U$ where U is a subspace and we denote $\tilde{n} := \dim(U)$. We think of $\tilde{K}$ and $\tilde{\Lambda}$ as $\tilde{n}$-dimensional objects. Let $\tilde{K}_t \subseteq \tilde{K}$ be the convex body after t iterations of the outer WHILE loop. Recall that $\mathrm{Vol}_{\tilde{n}}(\tilde{K}_t) \leq (\frac{3}{4})^t \cdot \mathrm{Vol}_{\tilde{n}}(\tilde{K})$ by Lemma 5 and $\mathrm{Vol}_{\tilde{n}}(\tilde{K}) \leq r^{\tilde{n}} \mathrm{Vol}_{\tilde{n}}(B_2^{\tilde{n}})$. Our goal is to show that for t large enough, there is a non-zero lattice vector $y \in \tilde{\Lambda}^*$ with $\|y\|_{(\tilde{K}_t - \tilde{K}_t)^\circ} \leq \frac{1}{2}$ which then causes the algorithm to recurse. To prove existence of such a vector y, we use Minkowski's Theorem (Theorem 2) followed by the Blaschke-Santaló-Bourgain-Milman Theorem (Theorem 5) to obtain

$$\begin{aligned}
\lambda_1(\tilde{\Lambda}^*, (\tilde{K}_t - \tilde{K}_t)^\circ) &\overset{\text{Thm 2}}{\leq} 2 \cdot \Big(\frac{\det(\tilde{\Lambda}^*)}{\mathrm{Vol}_{\tilde{n}}((\tilde{K}_t - \tilde{K}_t)^\circ)} \Big)^{1/\tilde{n}} \\
&\overset{\text{Thm 5}}{\leq} 2C \cdot \Big(\frac{\mathrm{Vol}_{\tilde{n}}(\tilde{K}_t - \tilde{K}_t)}{\det(\tilde{\Lambda}) \cdot \mathrm{Vol}_{\tilde{n}}(B_2^{\tilde{n}})^2} \Big)^{1/\tilde{n}} \\
&\overset{\text{Thm 4}}{\leq} 2 \cdot 4 \cdot \frac{\sqrt{\tilde{n}}}{2} \cdot C \Big(\frac{\mathrm{Vol}_{\tilde{n}}(\tilde{K}_t)}{\det(\tilde{\Lambda}) \cdot \mathrm{Vol}_{\tilde{n}}(B_2^{\tilde{n}})} \Big)^{1/\tilde{n}} \\
&\leq 4C\sqrt{\tilde{n}} \cdot r \cdot \frac{(3/4)^{t/\tilde{n}}}{\det(\tilde{\Lambda})^{1/\tilde{n}}} \leq 4C \cdot \frac{\tilde{n} \cdot r}{\lambda_1(\Lambda)} \cdot (3/4)^{t/\tilde{n}}
\end{aligned}$$

Here we use the convenient estimate of $\mathrm{Vol}_{\tilde{n}}(B_2^{\tilde{n}}) \geq \mathrm{Vol}_{\tilde{n}}(\frac{1}{\sqrt{\tilde{n}}} B_\infty^{\tilde{n}}) = (\frac{2}{\sqrt{\tilde{n}}})^{\tilde{n}}$. Moreover, we have used that by Lemma 1 one has $\det(\tilde{\Lambda}) \geq (\frac{\lambda_1(\Lambda)}{\sqrt{\tilde{n}}})^{\tilde{n}}$. Then $t = \Theta(\tilde{n} \log(\frac{\tilde{n} r}{\lambda_1(\Lambda)}))$ iterations suffice until $\lambda_1(\tilde{\Lambda}^*, (\tilde{K}_t - \tilde{K}_t)^\circ) \leq \frac{1}{2}$ and the algorithm recurses. Hence the total number of iterations of the outer WHILE loop over all recursion levels can be bounded by $O(n^2 \log(\frac{nr}{\lambda_1(\Lambda)}))$.

The iteration bound of Lemma 6 can be improved by amortizing the volume reduction over the different recursion levels following the approach of Jiang [18]. We refrain from that to keep our approach simple.

3.2 Correctness and Efficiency of Subroutines

Next, we verify that the subroutines are used correctly. The proofs in this section are deferred to the full version of this paper.

Lemma 7. *For any convex body $K \subseteq \mathbb{R}^n$ one can compute the barycenter c and a $\mathbf{0}$-centered ellipsoid $\mathcal{E}$ in randomized polynomial time so that $c + \mathcal{E} \subseteq K \subseteq c + (n+1)\mathcal{E}$.*

In order for the call of ApxIP in step (7) to be efficient, we need that the symmetrizer of the set is large enough volume-wise, see Theorem 1. In particular for any parameters $2^{-\Theta(n)} \leq \rho \leq 0.99$ and $R \leq 2^{O(n)}$ we will have $\mathrm{Vol}_n((Q - \tilde{c}) \cap (\tilde{c} - Q)) \geq 2^{-\Theta(n)} \mathrm{Vol}_n(Q)$ which suffices for our purpose.

Lemma 8. *In step (7), the set $Q := \{x \in K \mid \langle a, x \rangle \geq \langle a, c + \frac{\rho}{2} d \rangle\}$ and the point $\tilde{c} := c + \rho d$ satisfy $Vol_n((Q - \tilde{c}) \cap (\tilde{c} - Q)) \geq (1-\rho)^n \cdot \frac{\rho}{2R} \cdot 2^{-n} \cdot Vol_n(Q)$.*

3.3 Conclusion on the Cut-Or-Average Algorithm

From the discussion above, we can summarize the performance of the algorithm in Fig. 1 as follows:

Theorem 7. *Given a full rank matrix $B \in \mathbb{Q}^{n\times n}$ and parameters $r > 0$ and $\ell \geq 5(n+1)$ with $\ell \in \mathbb{N}$ and a separation oracle for a closed convex set $K \subseteq rB_2^n$, there is a randomized algorithm that with high probability finds a point $x \in K \cap \frac{1}{\ell}\Lambda(B)$ or decides that $K \cap \Lambda(B) = \emptyset$. Here the running time is $2^{O(n)}$ times a polynomial in $\log(r)$ and the encoding length of B.*

This can be easily turned into an algorithm to solve integer linear programming:

Theorem 8. *Given a full rank matrix $B \in \mathbb{Q}^{n\times n}$, a parameter $r > 0$ and a separation oracle for a closed convex set $K \subseteq rB_2^n$, there is a randomized algorithm that with high probability finds a point $x \in K \cap \Lambda(B)$ or decides that there is none. The running time is $2^{O(n)} n^n$ times a polynomial in $\log(r)$ and the encoding length of B.*

Proof. Suppose that $K \cap \Lambda \neq \emptyset$ and fix an (unknown) solution $x^* \in K \cap \Lambda$. We set $\ell := \lceil 5(n+1) \rceil$. We iterate through all $v \in \{0, \ldots, \ell - 1\}^n$ and run Theorem 7 on the set K and the shifted lattice $v + \ell\Lambda$. For the outcome of v with $x^* \equiv v \mod \ell$ one has $K \cap (v + \ell\Lambda) \neq \emptyset$ and so the algorithm will discover a point $x \in K \cap (v + \Lambda)$.

4 An Asymmetric Approximate Carathéodory Theorem

The *Approximate Carathéodory Theorem* states the following.

> Given any point-set $X \subseteq B_2^n$ in the unit ball with $\mathbf{0} \in \text{conv}(X)$ and a parameter $k \in \mathbb{N}$, there exist $u_1, \ldots, u_k \in X$ (possibly with repetition) such that
>
> $$\Big\| \frac{1}{k} \sum_{i=1}^{k} u_i \Big\|_2 \leq O\big(1/\sqrt{k}\big).$$

The theorem is proved, for example, by Novikoff [28] in the context of the *perceptron algorithm*. An ℓ_p-version was provided by Barman [4] to find Nash equilibria. Deterministic and nearly-linear time methods to find the convex combination were recently described in [26]. In the following, we provide a generalization to asymmetric convex bodies and the dependence on k will be weaker but sufficient for our analysis of our Cut-or-Average algorithm from Sect. 3.

Recall that with a symmetric convex body K, we one can associate the *Minkowski norm* $\|\cdot\|_K$ with $\|x\|_K = \inf\{s \geq 0 \mid x \in sK\}$. In the following we will use the same definition also for an arbitrary convex set K with $\mathbf{0} \in K$. Symmetry is not given but one still has $\|x + y\|_K \leq \|x\|_K + \|y\|_K$ for all $x, y \in \mathbb{R}^n$ and $\|\alpha x\|_K = \alpha \|x\|_K$ for $\alpha \in \mathbb{R}_{\geq 0}$. Using this notation we can prove the main result of this section.

Lemma 9. *Given a point-set $X \subseteq K$ contained in a convex set $K \subseteq \mathbb{R}^n$ with $\mathbf{0} \in$ conv(X) and a parameter $k \in \mathbb{N}$, there exist $u_1, \ldots, u_k \in X$ (possibly with repetition) so that*

$$\Big\| \frac{1}{k} \sum_{i=1}^{k} u_i \Big\|_K \leq \min\{|X|, n+1\}/k.$$

Moreover, given X as input, the points $u_1, \ldots, u_k$ can be found in time polynomial in $|X|$, k and n.

Proof. Let $\ell = \min\{|X|, n+1\}$. The claim is true whenever $k \le \ell$ since then we may simply pick an arbitrary point in X. Hence from now on we assume $k > \ell$.

By Carathéodory's theorem, there exists a convex combination of zero, using ℓ elements of X. We write $\mathbf{0} = \sum_{i=1}^{\ell} \lambda_i v_i$ where $v_i \in X$, $\lambda_i \ge 0$ for $i \in [\ell]$ and $\sum_{i=1}^{\ell} \lambda_i = 1$. Consider the numbers $L_i = (k-\ell)\lambda_i + 1$. Clearly, $\sum_{i=1}^{\ell} L_i = k$. This implies that there exists an integer vector $\mu \in \mathbb{N}^{\ell}$ with $\mu \ge (k-\ell)\lambda$ and $\sum_{i=1}^{\ell} \mu_i = k$. It remains to show that we have

$$\left\| \frac{1}{k} \sum_{i=1}^{\ell} \mu_i v_i \right\|_K \le \ell/k.$$

In fact, one has

$$\begin{aligned} \Big\| \sum_{i=1}^{\ell} \mu_i v_i \Big\|_K &= \Big\| \sum_{i=1}^{\ell} \underbrace{(\mu_i - (k-\ell)\lambda_i)}_{\ge 0} v_i + \underbrace{(k-\ell)}_{\ge 0} \sum_{i=1}^{\ell} \lambda_i v_i \Big\|_K \\ &\le \sum_{i=1}^{\ell} (\mu_i - (k-\ell)\lambda_i) \underbrace{\|v_i\|_K}_{\le 1} + (k-\ell) \underbrace{\Big\| \sum_{i=1}^{\ell} \lambda_i v_i \Big\|_K}_{=0} \le \ell. \end{aligned}$$

For the moreover part, note that the coefficients $\lambda_1, \ldots, \lambda_\ell$ are the extreme points of a linear program which can be found in polynomial time. Finally, the linear system $\mu \ge \lceil (k-\ell)\lambda \rceil, \sum_{i=1}^{\ell} \mu_i = k$ has a totally unimodular constraint matrix and the right hand side is integral, hence any extreme point solution is integral as well, see e.g. [31].

Lemma 10. *For any integer $\ell \ge 5(n+1)$, the convex combination μ computed in line (10) satisfies $\sum_{x \in X} \mu_x x \in K$.*

Proof. We may translate the sets X and K so that $c = \mathbf{0}$ without affecting the claim. Recall that $z \in \mathrm{conv}(X)$. By Carathéodory's Theorem there are $v_1, \ldots, v_m \in X$ with $m \le n+1$ so that $z \in \mathrm{conv}\{v_1, \ldots, v_m\}$ and so $\mathbf{0} \in \mathrm{conv}\{v_1 - z, \ldots, v_m - z\}$. We have $v_i \in 3K$ by Lemma 3 and $-z \in \frac{1}{4}\mathcal{E} \subseteq \frac{1}{4}K$ as well as $z \in \frac{1}{4}K$. Hence $\|v_i - z\|_K \le \|v_i\|_K + \|-z\|_K \le \frac{13}{4}$. We apply Lemma 9 and obtain a convex combination $\mu \in \frac{\mathbb{Z}_{\ge 0}^m}{\ell}$ with $\|\sum_{i=1}^{m} \mu_i (v_i - z)\|_{\frac{13}{4}K} \le \frac{m}{\ell}$. Then

$$\Big\| \sum_{i=1}^{m} \mu_i v_i \Big\|_K \le \Big\| \sum_{i=1}^{m} \mu_i (v_i - z) \Big\|_K + \underbrace{\|z\|_K}_{\le 1/4} \le \frac{13}{4} \frac{m}{\ell} + \frac{1}{4} \le 1$$

if $\ell \ge \frac{13}{3} m$. This is satisfies if $\ell \ge 5(n+1)$.

5 IPs with Polynomial Variable Range

Now we come to our second method that reduces (IP) to (APPROX-IP) that applies to integer programming in *standard equation form*

$$Ax = b,\ x \in \mathbb{Z}^n,\ 0 \le x_i \le u_i,\ i = 1, \ldots, n, \tag{2}$$

Here, $A \in \mathbb{Z}^{m\times n}$, $b \in \mathbb{Z}^m$, and the $u_i \in \mathbb{N}_+$ are positive integers that bound the variables from above. Our main goal is to prove the following theorem.

Theorem 9. *The integer feasibility problem in standard equation form (see (2)) can be solved in time* $2^{O(n)} \prod_{i=1}^n \log_2(u_i+1)$.

We now describe the algorithm. It is again based on the approximate integer programming technique of Dadush [11]. We exploit it to solve integer programming exactly via the technique of *reflection sets* developed by Cook et al. [9]. For each $i = 1,\dots,n$ we consider the two families of hyperplanes that slice the feasible region with the shifted lower and upper bounds respectively

$$x_i = 2^{j-1} \text{ and } x_i = u_i - 2^{j-1},\ 0 \le j \le \lceil \log_2(u_i)\rceil. \tag{3}$$

Following [9], we consider two points w, v that lie in the region between two consecutive planes $x_i = 2^{j-1}$ and $x_i = 2^j$ for some j. Suppose that $w_i \le v_i$ holds. Let s be the point such that $w = 1/2(s+v)$. The line-segment s, v is the line segment w, v scaled by a factor of 2 from v. Let us consider what can be said about the i-th component of s. Clearly $s_i \ge 2^{j-1} - (2^j - 2^{j-1}) = 0$. Similarly, if w and v lie in the region in-between $x_i = 0$ and $x_i = 1/2$, then $s_i \ge -1/2$. We conclude with the following observation.

Lemma 11. *Consider the hyperplane arrangement defined by the equations (3) as well as by $x_i = 0$ and $x_i = u_i$ for $1 \le i \le n$. Let $K \subseteq \mathbb{R}^n$ a cell of this hyperplane arrangement and $v \in K$. If K' is the result of scaling K by a factor of 2 from v, i.e.*

$$K' = \{v + 2(w - v) \mid w \in K\},$$

then K' satisfies the inequalities $-1/2 \le x_i \le u_i + 1/2$ for all $1 \le i \le n$.

We use this observation to prove Theorem 9:

Proof (Proof of Theorem 9). The task of (2) is to find an integer point in the affine subspace defined by the system of equations $Ax = b$ that satisfies the bound constraints $0 \le x_i \le u_i$. We first partition the feasible region with the hyperplanes (3) as well as $x_i = 0$ and $x_i = u_i$ for each i. We then apply the approximate integer programming algorithm with approximation factor 2 on each convex set $P_K = \{x \in \mathbb{R}^n \mid Ax = b\} \cap K$ where K ranges over all cells of the arrangement. In $2^{O(n)}$ time, the algorithm either finds an integer point in the convex set C_K that results from P_K by scaling it with a factor of 2 from its center of gravity, or it asserts that P_K does not contain an integer point. Clearly, $C_K \subseteq \{x \in \mathbb{R}^n \mid Ax = b\}$ and if the algorithm returns an integer point x^*, then, by Lemma 11, this integer point also satisfies the bounds $0 \le x_i \le u_i$. The running time of the algorithm is equal to the number of cells times $2^{O(n)}$ which is $2^{O(n)} \prod_{i=1}^n \log_2(u_i+1)$.

IPs in Inequality Form

We can also use Theorem 9 to solve integer linear programs in *inequality form*. Here the efficiency is strongly dependent on the number of inequalities.

Theorem 10. *Let $A \in \mathbb{Q}^{m\times n}$, $b \in \mathbb{Q}^m$, $c \in \mathbb{Q}^n$ and $u \in \mathbb{N}_+^n$. Then the integer linear program*

$$\max\left\{\langle c,x\rangle \mid Ax \le b,\ 0 \le x \le u,\ x \in \mathbb{Z}^n\right\}$$

can be solved in time $n^{O(m)} \cdot (2\log(1+\Delta))^{O(n+m)}$ where $\Delta := \max\{u_i \mid i = 1,\dots,n\}$.

Proof. Via binary search it suffices to solve the feasibility problem

$$\langle c,x\rangle \ge \gamma,\ Ax \le b,\ 0 \le x \le u,\ x \in \mathbb{Z}^n \tag{4}$$

in the same claimed running time. We apply the result of Frank and Tardos (Theorem 6) and replace c,γ,A,b by integer-valued objects of bounded $\|\cdot\|_\infty$-norm so that the feasible region of (4) remains the same. Hence we may indeed assume that $c \in \mathbb{Z}^n$, $\gamma \in \mathbb{Z}$, $A \in \mathbb{Z}^{m\times n}$ and $b \in \mathbb{Z}^m$ with $\|c\|_\infty, |\gamma|, \|A\|_\infty, \|b\|_\infty \le 2^{O(n^3)} \cdot \Delta^{O(n^2)}$. Any feasible solution x to (4) has a slack bounded by $\gamma - \langle c,x\rangle \le |\gamma| + \|c\|_\infty \cdot n \cdot \Delta \le N$ where we may choose $N := 2^{O(n^3)}\Delta^{O(n^2)}$. Similarly $b_i - A_i x \le N$ for all $i \in [n]$. We can then introduce slack variables $y \in \mathbb{Z}_{\ge 0}$ and $z \in \mathbb{Z}^m_{\ge 0}$ and consider the system

$$\begin{aligned} &\langle c,x\rangle + y = \gamma, && Ax + z = b,\\ &0 \le x \le u, && 0 \le y \le N,\ \ 0 \le z_j \le N\ \forall j \in [m],\\ &(x,y,z) \in \mathbb{Z}^{n+1+m} \end{aligned} \tag{5}$$

in equality form which is feasible if and only if (4) is feasible. Then Theorem 9 shows that such an integer linear program can be solved in time

$$2^{O(n+m)} \cdot \left(\prod_{i=1}^{n} \ln(1+u_i)\right) \cdot (\ln(1+N))^{m+1} \le n^{O(m)} \cdot (2\log(1+\Delta))^{O(n+m)}.$$

Subset Sum and Knapsack

The *subset-sum problem (with multiplicities)* is an integer program of the form (2) with one linear constraint. Polak and Rohwedder [29] have shown that subset-sum with multiplicities—that means $\sum_{i=1}^n x_i z_i = t, 0 \le x_i \le u_i\ \forall i \in [n], x \in \mathbb{Z}^n$—can be solved in time $O(n + z_{\max}^{5/3})$ times a polylogarithmic factor where $z_{\max} := \max_{i=1,\dots,n} z_i$. The algorithm of Frank and Tardos [15] (Theorem 6) finds an equivalent instance in which $z_{\max}$ is bounded by $2^{O(n^3)} u_{\max}^{O(n^2)}$. All-together, if each multiplicity is bounded by a polynomial $p(n)$, then the state-of-the-art for subset-sum with multiplicities is straightforward enumeration resulting in a running time $n^{O(n)}$ which is the current best running time for integer programming. We can significantly improve the running time in this regime. This is a direct consequence of Theorem 10.

Corollary 2. *The subset sum problem with multiplicities of the form $\sum_{i=1}^n x_i z_i = t, 0 \le x \le u, x \in \mathbb{Z}^n$ can be solved in time $2^{O(n)} \cdot (\log(1+\|u\|_\infty))^n$. In particular if each multiplicity is bounded by a polynomial $p(n)$, then it can be solved in time $(\log n)^{O(n)}$.*

Knapsack with multiplicities is the following integer programming problem

$$\max\left\{\langle c,x\rangle \mid x \in \mathbb{Z}^n_{\ge 0},\ \langle a,x\rangle \le \beta, 0 \le x \le u\right\}, \tag{6}$$

where $c, a, u \in \mathbb{Z}_{\geq 0}^n$ are integer vectors. Again, via the preprocessing algorithm of Frank and Tardos [15] (Theorem 6) one can assume that $\|c\|_\infty$ as well as $\|a\|_\infty$ are bounded by $2^{O(n^3)} u_{\max}^{O(n^2)}$. If each u_i is bounded by a polynomial in the dimension, then the state-of-the-art for this problem is again straightforward enumeration which leads to a running time of $n^{O(n)}$. Also in this regime, we can significantly improve the running time which is an immediate consequence of Theorem 10.

Corollary 3. *A knapsack problem* (6) *can be solved in time* $2^{O(n)} \cdot (\log(1 + \|u\|_\infty))^n$. *In particular if* $\|u\|_\infty$ *is bounded by a polynomial* $p(n)$ *in the dimension, it can be solved in time* $(\log n)^{O(n)}$.

References

1. Miklós Ajtai, R.K., Sivakumar, D.: A sieve algorithm for the shortest lattice vector problem. In: Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, pp. 601–610 (2001)
2. Alon, N., Vũ, V.H.: Anti-hadamard matrices, coin weighing, threshold gates, and indecomposable hypergraphs. J. Comb. Theory Ser. A **79**(1), 133–160 (1997)
3. Artstein-Avidan, S., Giannopoulos, A., Milman, V.D.: Asymptotic geometric analysis. Part I, Volume 202 of Mathematical Surveys and Monographs. American Mathematical Society, Providence, RI (2015)
4. Barman, S.: Approximating nash equilibria and dense bipartite subgraphs via an approximate version of Caratheodory's theorem. In: Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing, pp. 361–369 (2015)
5. Bellman, R.: Dynamic programming. Science **153**(3731), 34–37 (1966)
6. Bertsimas, D., Vempala, S.: Solving convex programs by random walks. J. ACM (JACM) **51**(4), 540–556 (2004)
7. Blömer, J., Naewe, S.: Sampling methods for shortest vectors, closest vectors and successive minima. Theor. Comput. Sci. **410**(18), 1648–1665 (2009)
8. Bringmann, K.: A near-linear pseudopolynomial time algorithm for subset sum. In: Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1073–1084. SIAM (2017)
9. Cook, W., Hartmann, M., Kannan, R., McDiarmid, C.: On integer points in polyhedra. Combinatorica **12**(1), 27–37 (1992)
10. Dadush, D.: Integer Programming, Lattice Algorithms, and Deterministic Volume Estimation. Georgia Institute of Technology (2012)
11. Dadush, D.: A randomized sieving algorithm for approximate integer programming. Algorithmica **70**(2), 208–244 (2014)
12. Dyer, M., Frieze, A., Kannan, R.: A random polynomial-time algorithm for approximating the volume of convex bodies. J. ACM **38**(1), 1–17 (1991)
13. Eisenbrand, F., Hähnle, N., Niemeier, M.: Covering cubes and the closest vector problem. In: Proceedings of the Twenty-Seventh Annual Symposium on Computational Geometry, pp. 417–423 (2011)
14. Eisenbrand, F., Weismantel, R.: Proximity results and faster algorithms for integer programming using the Steinitz lemma. ACM Trans. Algorithms (TALG) **16**(1), 1–14 (2019)
15. Frank, A., Tardos, É.: An application of simultaneous diophantine approximation in combinatorial optimization. Combinatorica **7**(1), 49–65 (1987)
16. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization. Algorithms and Combinatorics, vol. 2. Springer, Heidelberg (1988). https://doi.org/10.1007/978-3-642-78240-4

17. Jansen, K., Rohwedder, L.: On integer programming and convolution. In: 10th Innovations in Theoretical Computer Science Conference (ITCS 2019). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2018)
18. Jiang, H.: Minimizing convex functions with integral minimizers. In: SODA, pp. 976–985. SIAM (2021)
19. Kannan, R.: Minkowski's convex body theorem and integer programming. Math. Oper. Res. **12**(3), 415–440 (1987)
20. Knop, D., Pilipczuk, M., Wrochna, M.: Tight complexity lower bounds for integer linear programming with few constraints. ACM Trans. Comput. Theory (TOCT) **12**(3), 1–19 (2020)
21. Lagarias, J.C., Odlyzko, A.M.: Solving low-density subset sum problems. J. ACM (JACM) **32**(1):229–246 (1985)
22. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische annalen **261**(ARTICLE), 515–534 (1982)
23. Lenstra Jr., H.W.: Integer programming with a fixed number of variables. Math. Oper. Res. **8**(4), 538–548 (1983)
24. Micciancio, D., Goldwasser, S.: Complexity of lattice problems - a cryptograhic perspective. The Kluwer International Series in Engineering and Computer Science, vol. 671. Springer, New York (2002). https://doi.org/10.1007/978-1-4615-0897-7
25. Micciancio, D., Voulgaris, P.: A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. In: Proceedings of the Forty-Second ACM Symposium on Theory of Computing, pp. 351–358 (2010)
26. Mirrokni, V., Leme, R.P., Vladu, A., Wong, S.C.: Tight bounds for approximate Carathéodory and beyond. In: International Conference on Machine Learning, pp. 2440–2448. PMLR (2017)
27. Nemhauser, G.L., Wolsey, L.A.: Integer programming. In: Nemhauser, G.L., et al. (eds.) Optimization. Handbooks in Operations Research and Management Science, chapter VI, vol. 1, pp. 447–527. Elsevier (1989)
28. Novikoff, A.B.: On convergence proofs for perceptrons. Technical report, Office of Naval Research, Washington, D.C. (1963)
29. Polak, A., Rohwedder, L., Węgrzycki, K.: Knapsack and subset sum with small items. In: 48th International Colloquium on Automata, Languages, and Programming (ICALP 2021), number CONF, pp. 106–1. Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2021)
30. Schrijver, A.: Polyhedral combinatorics. In: Graham, R., Grötschel, M., Lovász, L. (eds.) Handbook of Combinatorics, chapter 30, vol. 2, pp. 1649–1704. Elsevier (1995)
31. Schrijver, A.: Theory of Linear and Integer Programming. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley, Hoboken (1999)