Polynomiality for Bin Packing with a Constant Number of Item Types

MICHEL X. GOEMANS, Massachusetts Institute of Technology, USA
THOMAS ROTHVOSS, University of Washington, USA

We consider the bin packing problem with d different item sizes s_i and item multiplicities a_i, where all numbers are given in binary encoding. This problem formulation is also known as the *one-dimensional cutting stock problem*. In this work, we provide an algorithm that, for constant d, solves bin packing in polynomial time. This was an open problem for all $d \geq 3$. In fact, for constant d our algorithm solves the following problem in polynomial time: Given two d-dimensional polytopes P and Q, find the smallest number of integer points in P whose sum lies in Q. Our approach also applies to *high multiplicity* scheduling problems in which the number of copies of each job type is given in binary encoding and each type comes with certain parameters such as release dates, processing times, and deadlines. We show that a variety of high multiplicity scheduling problems can be solved in polynomial time if the number of job types is constant.

CCS Concepts: • **Theory of computation → Discrete optimization**; • **Mathematics of computing → Combinatorial optimization**;

Additional Key Words and Phrases: Bin packing, integer programming, high multiplicity scheduling, polynomial time algorithms

ACM Reference format:
Michel X. Goemans and Thomas Rothvoss. 2020. Polynomiality for Bin Packing with a Constant Number of Item Types. *J. ACM* 67, 6, Article 38 (October 2020), 21 pages.
https://doi.org/10.1145/3421750

1 INTRODUCTION

Let (s, a) be an instance for *bin packing* with *item sizes* $s_1, \ldots, s_d \in [0, 1]$ and a vector $a \in \mathbb{Z}_{\geq 0}^d$ of *item multiplicities*. In other words, our instance contains a_i many copies of an item of size s_i. In the following, we assume that s_i is given as a rational number and Δ is the largest number appearing in the denominator of s_i or the multiplicities a_i. Let $\mathcal{P} := \{x \in \mathbb{Z}_{\geq 0}^d \mid s^T x \leq 1\}$, see Figure 1. Now the goal is to select a minimum number of vectors from $\mathcal{P}$ that sum up to a, i.e.,

$$\min \left\{ \mathbf{1}^T \lambda \mid \sum_{x \in \mathcal{P}} \lambda_x \cdot x = a; \;\; \lambda \in \mathbb{Z}_{\geq 0}^{\mathcal{P}} \right\}, \tag{1}$$

Michel X. Goemans is supported by ONR grants N00014-11-1-0053, N00014-17-1-2177, and NSF contract 1115849. Thomas Rothvoss is supported by an Alfred P. Sloan Research Fellowship, a David and Lucile Packard Fellowship, and NSF CAREER Award 1651861. The research was done while the second author was postdoc at MIT.
Authors' addresses: M. X. Goemans, Department of Mathematics, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139-4307; email: goemans@math.mit.edu; T. Rothvoss, Department of Mathematics, University of Washington, Box 354350, Seattle, WA 98195-4350; email: rothvoss@uw.edu.

0004-5411/2020/10-ART38 $15.00
https://doi.org/10.1145/3421750

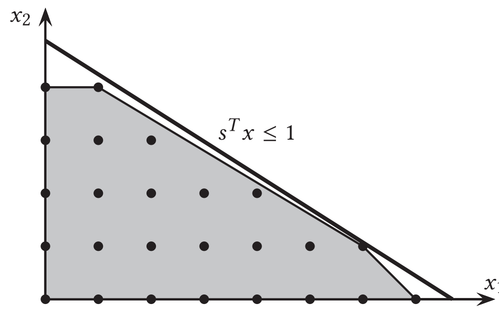


Fig. 1. Knapsack polytope for $s = (0.13, 0.205)$.

where λ_x is the *weight* that is given to $x \in \mathcal{P}$. This problem is also known as the *(one-dimensional) cutting stock problem* and its study goes back to the classical paper by Gilmore and Gomory [9]. Note that even for fixed dimension d, the problem is that both, the number of points $|\mathcal{P}|$ and the weights λ_x will be exponentially large. Let OPT and OPT_f be the optimum integral and fractional solution to (1). As bin packing for general d is strongly **NP**-hard [16], we are particularly interested in the complexity of bin packing if d is constant. For $d = 2$, it is true that $OPT = \lceil OPT_f \rceil$ and it suffices to compute and round an optimum fractional solution [23]. However, for $d \geq 3$, one might have $OPT > \lceil OPT_f \rceil$. Still, Reference [7] generalized the argument of Reference [23] to find a solution with at most $d - 2$ bins more than the optimum in polynomial time.

The best polynomial time algorithm previously known for constant $d \geq 3$ is an $OPT + 1$ approximation algorithm by Jansen and Solis-Oba [18] that runs in time $2^{2^{O(d)}} \cdot (\log \Delta)^{O(1)}$. Their algorithm uses a *mixed integer linear program* and is based on the following insights: (1) If all items are small, say, $s_i \leq \frac{1}{2d}$, then the integrality gap is at most one.[1] (2) If all items have constant size, then one can guess the points used in the optimum solution. It turns out that for arbitrary instances both approaches can be combined for an $OPT + 1$ algorithm. Also note that if the *mixed integer roundup conjecture* holds true, then there is indeed a significantly simpler algorithm achieving the same bound, again by Jansen and Solis-Oba [17]. However, to find an optimum solution, we cannot allow any error and a fundamentally different approach is needed.

Note that for general d, the recent algorithm of Hoberg and the second author provides solutions of cost at most $OPT + O(\log d)$ [12, 25], improving the classical Karmarkar-Karp algorithm with a guarantee of $OPT + O(\log^2 d)$ [20]. Both algorithms run in time polynomial in $\sum_{i=1}^d a_i$ and thus count in our setting as pseudopolynomial. In fact, those algorithms can still be cast as asymptotic FPTAS.

Bin packing and more generally the cutting stock problem belong to a family of problems that consist of selecting integer points in a polytope with multiplicities. In fact, several scheduling problems fall into this framework as well, where the polytope describes the set of jobs that are admissible on a machine under various constraints.

We give some notation needed throughout the article. For a set $X \subseteq \mathbb{R}^d$, we define the spanned *cone* as $\text{cone}(X) = \{\sum_{x \in X} \lambda_x x \mid \lambda_x \geq 0 \ \forall x \in X\}$ and the *integer cone* as $\text{int.cone}(X) := \{\sum_{x \in X} \lambda_x x \mid \lambda_x \in \mathbb{Z}_{\geq 0} \ \forall x \in X\}$. We will only consider rational polyhedra $P = \{x \in \mathbb{R}^d \mid Ax \leq b\}$ where both A and b are rational. We define $\text{enc}(P)$ as the number of bits that it takes to write down the rational numbers in A and in b. For definitness, one can use the definitions of Korte and Vygen [21], Chapter 4. However, we will only need that $\text{enc}(P)$ is polynomially related to $\max\{m, d, \log \Delta\}$ where

[1] Compute a basic solution λ to the LP and buy $\lfloor \lambda_x \rfloor$ times point x. Then assign the items in the remaining instance greedily.

m is the number of inequalities and Δ is the largest nominator or denominator appearing in any of the inequalities definining of P. For the remainder of this article we denote $\log(y) := \log_2(y)$. We point out that any rational inequalities can be multiplied by a least common multiple of the denominators to obtain an equivalent system $\tilde{A}x \leq \tilde{b}$ with integral coefficients whose encoding length is within a polynomial factor of the original one.

2 OUR CONTRIBUTIONS

In this article, we resolve the question of whether bin packing with a fixed number of item types can be solved in polynomial time.

Theorem 2.1. *For any bin packing instance (s, a) with $s \in [0,1]^d \cap \mathbb{Q}^d$ and $a \in \mathbb{Z}^d_{\geq 0}$, an optimum integral solution can be computed in time $(\log \Delta)^{2^{O(d)}}$ where $\Delta := \max\{\|a\|_\infty, \|\beta\|_\infty, 4\}$ where $s_i = \frac{\alpha_i}{\beta_i}$ with $\alpha_i \in \mathbb{Z}_{\geq 0}, \beta_i \in \mathbb{Z}_{\geq 1}$.*

This answers an open question posed by McCormick et al. [23] as well as by Eisenbrand and Shmonin [6]. In fact, the first article even conjectured this problem to be **NP**-hard for $d = 3$. Moreover, the polynomial solvability for general d was called a "*hard open problem*" by Filippi [8].

We derive Theorem 2.1 via the following general theorem for finding conic integer combinations in fixed dimension.

Theorem 2.2. *Given rational polyhedra $P, Q \subseteq \mathbb{R}^d$ where P is bounded, one can find a vector $y \in \text{int.cone}(P \cap \mathbb{Z}^d) \cap Q$ and a vector $\lambda \in \mathbb{Z}^{P\cap\mathbb{Z}^d}_{\geq 0}$ such that $y = \sum_{x \in P\cap\mathbb{Z}^d} \lambda_x x$ in time $enc(P)^{2^{O(d)}} \cdot enc(Q)^{O(1)}$ or decide that no such y exists. Moreover, the support of λ is always bounded by 2^{2d+1}.*

In fact, by choosing $P = \{\binom{x}{1} \in \mathbb{R}^{d+1}_{\geq 0} \mid s^T x \leq 1\}$ and $Q = \{a\} \times [0, b]$, we can decide in polynomial time, whether b bins suffice. Theorem 2.1 then follows using binary search. While for the proof strategy it will be convinient that P is bounded, it turns out that one can reduce the case where P is an unbounded rational polyhedron to Theorem 2.2. However, we postpone that reduction to Section 7.

Our main insight to prove Theorem 2.2 lies in the following structure theorem that says that, for fixed d, there is a pre-computable polynomial size set $X \subseteq P \cap \mathbb{Z}^d$ of special vectors that are *independent* of the target polytope Q with the property that, for any $y \in \text{int.cone}(P \cap \mathbb{Z}^d) \cap Q$, there is always a conic integer combination that has all but a constant amount of weight on a constant number of vectors in X.

Theorem 2.3 (Structure Theorem). *Let $P = \{x \in \mathbb{R}^d \mid Ax \leq b\}$ be a polytope with $A \in \mathbb{Z}^{m\times d}, b \in \mathbb{Z}^m$ and set $\Delta := \max\{\|A\|_\infty, \|b\|_\infty, 2\}$. Then there exists a set $X \subseteq P \cap \mathbb{Z}^d$ of size $|X| \leq N := m^d d^{O(d)} (\log \Delta)^d$ that can be computed in time $N^{O(1)}$ with the following property: For any vector $a \in \text{int.cone}(P \cap \mathbb{Z}^d)$ there exists an integral vector $\lambda \in \mathbb{Z}^{P\cap\mathbb{Z}^d}_{\geq 0}$ such that $\sum_{x\in P\cap\mathbb{Z}^d} \lambda_x x = a$ and*

(1) $\lambda_x \in \{0, 1\} \;\; \forall x \in (P \cap \mathbb{Z}^d) \setminus X$
(2) $|supp(\lambda) \cap X| \leq 2^{2d}$
(3) $|supp(\lambda) \setminus X| \leq 2^{2d}$.

With this structure theorem one can obtain Theorem 2.2 simply by computing X, guessing $\text{supp}(\lambda) \cap X$ and finding the corresponding values of λ and the vectors in $\text{supp}(\lambda) \setminus X$ with an integer program with a constant number of variables.

Bin packing can also be considered as a scheduling problem where the processing times correspond to the item sizes and the number of machines should be minimized, given a bound on the makespan. A variety of scheduling problems in the so-called high multiplicity setting can also be tackled using Theorem 2.2. Some of these scheduling applications are described in Section 6. For example we can solve in polynomial time the high multiplicity variant of minimizing the makespan for unrelated machines with machine-dependent release dates for a fixed number of job types and machine types. For an overview over the vast literature in high multiplicity scheduling we refer to the article of McCormick et al. [24] as well as the one by Hochbaum and Shamir [13].

3 PRELIMINARIES

In this section, we are going to review some known tools that we are going to use in our algorithm. The first one is Lenstra's well-known algorithm for integer programming, that runs in polynomial time as long as d is fixed.[2]

Theorem 3.1 (Lenstra [22], Kannan [19]). *Given $A \in \mathbb{Z}^{m\times d}$ and $b \in \mathbb{Z}^m$ with $\Delta := \max\{\|A\|_\infty, \|b\|_\infty, 2\}$. Then one can find an $x \in \mathbb{Z}^d$ with $Ax \leq b$ (or decide that none exists) in time $d^{O(d)} \cdot m^{O(1)} \cdot (\log \Delta)^{O(1)}$.*

For a polytope $P \subseteq \mathbb{R}^d$, the *integral hull* is the convex hull of the integral points, abbreviated with $P_I := \text{conv}(P \cap \mathbb{Z}^d)$ and the *extreme points* of P (also called *vertices*) are denoted by $\text{vert}(P)$. If we consider a low-dimensional polytope P, then P can indeed contain an exponential number of integral points—but only few of those can be extreme points of P_I.

Theorem 3.2 (Cook et al. [4, 11]). *Consider any polytope $P = \{x \in \mathbb{R}^d \mid Ax \leq b\}$ with $A \in \mathbb{Z}^{m\times d}$, $b \in \mathbb{Z}^m$ and $\Delta := \max\{\|A\|_\infty, \|b\|_\infty, 2\}$. Then $P_I = conv(P \cap \mathbb{Z}^d)$ has at most $m^d \cdot (O(\log \Delta))^d$ many extreme points. In fact a list of extreme points can be computed in time $d^{O(d)}(m \cdot \log(\Delta))^{O(d)}$.*

This bound is essentially tight. Bárány et al. [1] found a family of polytopes $P \subseteq \mathbb{R}^d$ (simplices, in fact) such that P_I has $\Omega(\varphi^{d-1})$ many extreme points where φ is the encoding length of P. A simple fact that we use frequently throughout the article is the following:

Lemma 3.3. *Let $P = \{x \in \mathbb{R}^d \mid Ax \leq b\}$ be a rational polyhedron where $A \in \mathbb{Z}^{m\times d}$ and $b \in \mathbb{Z}^m$ and set $\Delta := \max\{\|A\|_\infty, \|b\|_\infty\}$. Then any $x \in vert(P)$ has $\|x\|_\infty \leq M$ where $M := d! \cdot \Delta^d$. Moreover, if P is bounded then $P \subseteq [-M, M]^d$.*

Proof. By Cramer's rule, the coordinates of a vertex x of P are of the form $\det(Q)/\det(R)$ where Q and R are $d \times d$ matrices filled with entries from $\{-\Delta, \dots, +\Delta\}$. Then $\|x\|_\infty \leq |\det(Q)| \leq d!\Delta^d = M$ as one can see from Laplace formula. The "moreover" part follows from the observation that the $\|\cdot\|_\infty$-norm is maximized at a vertex if P is bounded. □

We will later refer to the coefficients λ_x as the *weight* given to x. For a vector $a \in \text{cone}(X)$, we know by *Carathéodory's Theorem* that there is always a corresponding vector $\lambda \geq \mathbf{0}$ with at most d non-zero entries and $a = \sum_{x\in X} \lambda_x x$. One may wonder how many points x are actually needed to generate some point in the integer cone. In fact, at least under the additional assumption that X is the set of integral points in a convex set, one can show that 2^d points suffice.[3] The arguments are crucial for our proofs, so we replicate the proof of Reference [6] to be self-contained.

Lemma 3.4 (Eisenbrand and Shmonin [6]). *For any polytope $P \subseteq \mathbb{R}^d$ and any integral vector $\lambda \in \mathbb{Z}_{\geq 0}^{P\cap\mathbb{Z}^d}$ there exists a $\mu \in \mathbb{Z}_{\geq 0}^{P\cap\mathbb{Z}^d}$ such that $|supp(\mu)| \leq 2^d$ and $\sum_x \mu_x x = \sum_x \lambda_x x$.*

Proof. For the sake of simplicity, we can replace the original P with $P := \text{conv}(x \mid \lambda_x > 0)$ without changing the claim. Let $f : \mathbb{R}^d \to \mathbb{R}$ be any strictly convex function, i.e., in particular we

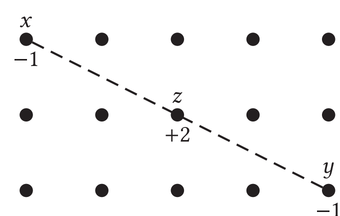
[2]Here, the original dependence of Reference [22] was $2^{O(d^3)}$, which was then improved by Kannan [19] to $d^{O(d)}$.
[3]For arbitrary $X \subseteq \mathbb{Z}^d$, one can show that a support of at most $O(d \log(dM))$ suffices, where M is the largest coefficient in a vector in X [6].

will use that

$$f\left(\frac{1}{2}x + \frac{1}{2}y\right) < \frac{1}{2}(f(x) + f(y)).$$

For example, $f(x) = \|x\|_2^2$ does the job. Let $(\mu_x)_{x \in P \cap \mathbb{Z}^d}$ be an integral vector with $\sum_{x \in P \cap \mathbb{Z}^d} \lambda_x x = \sum_{x \in P \cap \mathbb{Z}^d} \mu_x x$ that minimizes the potential function $\sum_{x \in P \cap \mathbb{Z}^d} \mu_x \cdot f(x)$ (note that there is at least one such solution, namely λ). In other words, we somewhat prefer points that are more in the "center" of the polytope. We claim that indeed $|\text{supp}(\mu)| \leq 2^d$.

For the sake of contradiction suppose that $|\text{supp}(\mu)| > 2^d$. Then there must be two points x, y with $\mu_x > 0$ and $\mu_y > 0$ that have the same *parity*, meaning that $x_i \equiv y_i \mod 2$ for all $i = 1, \ldots, d$. Then $z := \frac{1}{2}(x + y)$ is an integral vector and $z \in P$. Now we remove one unit of weight from both x and y and add 2 units to z.



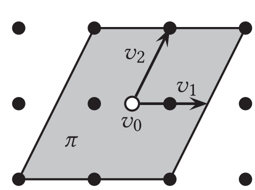
This gives us another feasible vector μ'. But the change in the potential function is $+2f(z) - f(x) - f(y) < 0$ by strict convexity of f, contradicting the minimality of μ. □

In fact, the bound is tight up to a constant factor. As it seems that this has not been observed in the literature before, we describe a construction in Section 8 where a support of size 2^{d-1} is actually needed.

A family of versatile and well-behaved polytopes is those of *parallelepipeds*. Recall that

$$\pi = \left\{ v_0 + \sum_{i=1}^{k} \mu_i v_i \mid -1 \leq \mu_i \leq 1 \ \forall i = 1, \ldots, k \right\}$$

is a *parallelepiped* with *center* $v_0 \in \mathbb{R}^d$ and *directions* $v_1, \ldots, v_k \in \mathbb{R}^d$. Usually one requires that the directions are linearly independent, that means $k \leq d$ and π is k-dimensional. We say that the parallelepiped is *integral* if all its 2^k many vertices are integral. Here is an example of an integral parallelepiped with $d = 2$ and $k = 2$:



Note that it is not necessary that all vectors $v_0, \ldots, v_k$ are integral.

4 PROOF OF THE STRUCTURE THEOREM

In this section, we are going to prove the structure theorem. The proof outline is as follows: We can show that the integral points in a polytope P can be covered with polynomially many integral parallelepipeds. The choice for X is then simply the set of vertices of those parallelepipeds. Now consider any vector a, which is a conic integer combination of points in P. Then by Lemma 3.4 we can assume that a is combined by using only a constant number of points in $P \cap \mathbb{Z}^d$. Consider

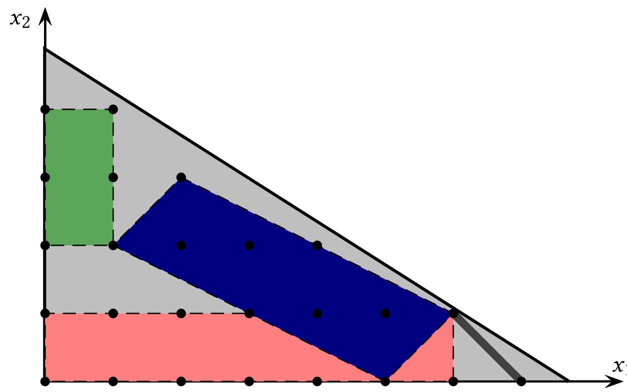


Fig. 2. Covering the integer points of a polytope with integral parallelepipeds.

such a point x^* and say it is used λ^* times. We will show that the weight λ^* can be almost entirely redistributed to the vertices of one of the parallelepipeds containing x^*.

Let us make these arguments more formal. We begin by showing that all the integer points in a polytope P can indeed be covered with polynomially many integral parallelepipeds as visualized in Figure 2. We will say that a polytope $S \subseteq \mathbb{R}^d$ is *symmetric around center* x_0 if $x_0 - x \in S \Leftrightarrow x_0 + x \in S$ for any $x \in \mathbb{R}^d$.

Lemma 4.1. *Let $P = \{x \in \mathbb{R}^d \mid Ax \leq b\}$ be a polytope where $A \in \mathbb{Z}^{m\times d}$ and $b \in \mathbb{Z}^m$ and set $\Delta := \max\{\|A\|_\infty, \|b\|_\infty, 2\}$. Then there exists a set Π of $|\Pi| \leq N := m^d d^{O(d)}(\log \Delta)^d$ many integral parallelepipeds such that*

$$P \cap \mathbb{Z}^d \subseteq \bigcup_{\pi\in\Pi} \pi \subseteq P.$$

Moreover, the set Π can be computed in time $N^{O(1)}$.

Proof. First, by Lemma 3.3 every point $x \in P$ has $\|x\|_\infty \leq d! \cdot \Delta^d$ and hence $|A_i x - b_i| \leq (d + 1)\Delta \cdot d! \cdot \Delta^d \leq (d + 1)! \cdot \Delta^{d+1}$. We want to partition the interval $[0, (d + 1)! \cdot \Delta^{d+1}]$ into smaller intervals $[\alpha_j, \alpha_{j+1}]$ such that for any integer values $p, q \in [\alpha_j, \alpha_{j+1}] \cap \mathbb{Z}$ one has $\frac{p}{q} \leq 1 + \frac{1}{d^2}$. For this we can choose $\alpha_j := (1 + \frac{1}{d^2})^{j-2}$ for $j = 1, \dots, K$ and $\alpha_0 := 0$. The number of intervals is $K \leq O(\log_{1+1/d^2}((d + 1)! \cdot \Delta^{d+1})) \leq O(d^3(\log \Delta + \log d))$.

Our next step is to partition P into *cells* such that points in the same cell have roughly the same slacks for all the constraints. For each sequence $j_1, \dots, j_m \in \{0, \dots, K - 1\}$ we define a cell $C = C(j_1, \dots, j_m)$ as

$$\left\{x \in \mathbb{R}^d \mid \alpha_{j_i} \leq b_i - A_i x \leq \alpha_{j_i+1}\ \forall i \in [m]\right\},$$

see Figure 3. In other words, we partition the polytope P using at most $M := m \cdot K$ many hyperplanes. By *Buck's Formula* [2], the number of k-dimensional cells in a d-dimensional arrangement of M hyperplanes is upper bounded by $f_k^{(d)}(M) := \sum_{i=d-k}^{d} \binom{M}{i}\binom{i}{d-k} \leq d \cdot (2M)^d$. Then the total number of cells of any dimension is generously bounded by $(d + 1) \cdot d \cdot (2M)^d \leq m^d d^{O(d)}(\log \Delta)^d$.

Before we continue, we want to comment on the geometry of the cells. Fix one of those cells $C = \left\{x \in \mathbb{R}^d \mid \alpha_{j_i} \leq b_i - A_i x \leq \alpha_{j_i+1}\ \forall i \in [m]\right\} \subseteq P$. Then for any coordinate i with $0 \leq \alpha_{j_i} < d^2$, we have $|\alpha_{j_i+1} - \alpha_{j_i}| < 1$. That means that the integer hull $\mathrm{conv}(C \cap \mathbb{Z}^d)$ is *flat* in direction A_i unless the cell has a slack of at least d^2 in that direction. □

Now we proceed to prove that there are only $d^{O(d)}$ integral parallelepipeds necessary to cover the integer points of this cell. We assume that $C \cap \mathbb{Z}^d \neq \emptyset$, otherwise there is nothing to do. Next,

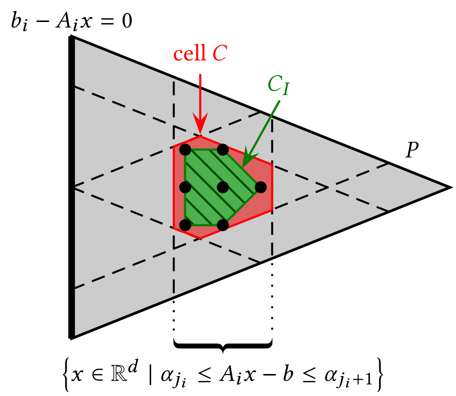


Fig. 3. Visualization of the slicing of P into cells.

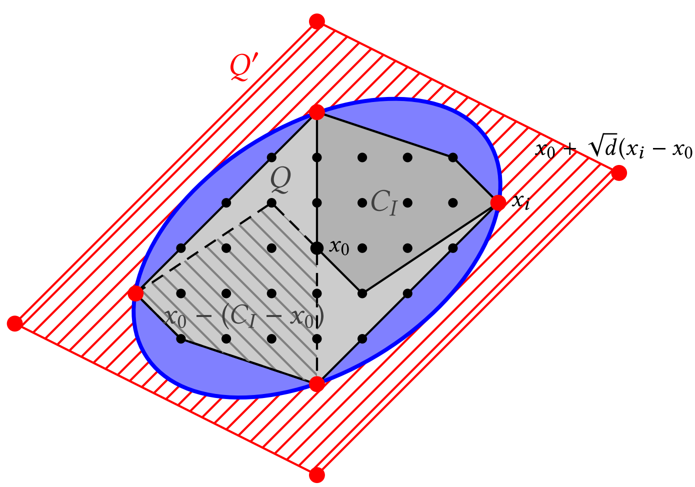


Fig. 4. Visualization of covering the integer points in a cell C_I: Start by obtaining the symmetric closure Q. Then compute the contact points of a minimum volume ellipsoid containing Q. Scale those points with $\sqrt{d}$ to obtain a polytope Q' with only $O(d^2)$ vertices containing C_I. Then extend a triangulation of Q' to $d^{O(d)}$ many parallelepipeds.

fix any integral point $x_0 \in C \cap \mathbb{Z}^d$ and define a slightly larger polytope $Q := \text{conv}(x_0 \pm (x_0 - x) \mid x \in C_I)$, see Figure 4. By construction, Q is a symmetric polytope with center x_0. Moreover all its vertices are integral, because vertices of C_I and x_0 are integral. The reason why we consider a symmetric polytope is the following classical theorem, which is paraphrased from John:

THEOREM 4.2 (JOHN [15]). *For any polytope $\tilde{P} \subseteq \mathbb{R}^d$ that is symmetric around center x_0, there are $k \leq \frac{1}{2}d(d+3)$ many extreme points $x_1, \ldots, x_k \in vert(\tilde{P})$ such that $\tilde{P} \subseteq conv(x_0 \pm \sqrt{d} \cdot (x_0 - x_j) \mid j \in [k])$.*

The original statement says that there is in fact an origin-centered ellipsoid E with $x_0 + \frac{1}{\sqrt{d}}E \subseteq \tilde{P} \subseteq x_0 + E$. But additionally John's Theorem provides a set of *contact points* in $\partial P \cap \partial E$ whose convex hull already contains the scaled ellipsoid $x_0 + \frac{1}{\sqrt{d}}E$. Moreover, the number of necessary contact points is at most $\frac{d}{2}(d+3)$, implying the above statement.

So we apply Theorem 4.2 to Q with center x_0 and obtain a list of points $x_1, \dots, x_k \in \mathrm{vert}(C_I)$ with $k \leq \frac{1}{2}d(d+3)$ such that

$$C_I \subseteq Q \subseteq \mathrm{conv}(x_0 \pm \lceil\sqrt{d}\rceil \cdot (x_0 - x_j) \mid j \in [k]) =: Q'.$$

Now it is not difficult to cover C_I with parallelepipeds of the form

$$\pi(J) := \Big\{x_0 + \sum_{j \in J} \mu_j (x_j - x_0) \mid |\mu_j| \leq \lceil\sqrt{d}\rceil \ \ \forall j \in J\Big\},$$

with $J \subseteq [k]$ and $\{x_j - x_0 \mid j \in J\}$ linearly independent. To see this take any point $x \in Q'$. By Carathéodory's Theorem, x lies already in the convex hull of x_0 plus at most d affinely independent vertices of Q', thus there is a subset of indices $J \subseteq [k]$ of size $|J| \leq d$ and signs $\varepsilon_j \in \{\pm 1\}$ with $x \in \mathrm{conv}(\{x_0\} \cup \{x_0 + \varepsilon_j \lceil\sqrt{d}\rceil \cdot (x_j - x_0) \mid j \in J\})$. Then clearly $x \in \pi(J)$.

Finally, it remains to show that all parallelepipeds $\pi(J)$ are still in P. Let $x = x_0 + \sum_{j \in J} \mu_j (x_j - x_0)$ with $|\mu_j| \leq \lceil\sqrt{d}\rceil$. We need to verify that x does not violate a constraint $i \in [m]$. First, consider the case that $j_i > 0$, that means C does not lie in the very first slice of constraint i. In this case, we have

$$b_i - A_i x \geq \underbrace{b_i - A_i x_0}_{\geq \alpha_{j_i}} - \sum_{j \in J} \underbrace{|\mu_j|}_{\leq \lceil\sqrt{d}\rceil} \cdot \underbrace{|A_i x_j - A_i x_0|}_{\leq \alpha_{j_i+1} - \alpha_{j_i} \leq \frac{\alpha_{j_i}}{d^2}} \geq 0,$$

where we crucially use that $|\alpha_{j_i+1} - \alpha_{j_i}| = (1 + \frac{1}{d^2})\alpha_{j_i} - \alpha_{j_i} = \frac{\alpha_{j_i}}{d^2}$. If indeed $j_i = 0$, then $|\alpha_{j_i+1} - \alpha_{j_i}| = (1 + \frac{1}{d^2})^{-1} < 1$ and for integer points x_j and x_0 one must have $|A_i x_j - A_i x_0| = 0$. Finally, observe that the number of subsets J of size at most d is $(\frac{1}{2}d(d+3))^d = d^{O(d)}$, which then gives the desired bound.

Now let us argue how to make this constructive in time $N^{O(1)}$. For each cell C, we list the vertices of the integer hull C_I in time $d^{O(d)} m^{O(d)} (\log \Delta)^{O(d)}$ by Theorem 3.2. Computing the minimum volume ellipsoid containing all those vertices is indeed a semidefinite program that can be solved in time polynomial in the encoding length of the vertices of C_I. We refer to Chapter 8 of Boyd and Vandenberghe [3] for details. The contact points can be inferred from the dual solution of this SDP and the associated parallelepipeds can be easily computed.

Note that one could have used the following simpler arguments to obtain a weaker bound that still leads to a polynomial time algorithm for bin packing if d is constant: First, each cell is defined by selecting m values $\alpha_{j_i} \in \{0, \dots, K-1\}$, hence the total number of cells is trivially upper bounded by K^m. Then every cell C_I has polynomially many vertices, hence it can be partitioned into polynomially many simplices. Then each simplex can be extended to a parallelepiped, whose union again covers C_I.

As a side remark, the partitioning with shifted hyperplanes was used before, e.g., in Reference [4] to bound the number of extreme points of $\mathrm{conv}(P \cap \mathbb{Z}^d)$. The next lemma says why parallelepipeds are so useful. Namely, the weight of any point in it can be almost completely redistributed to its vertices.

Lemma 4.3. *Given an integral parallelepiped π with vertices $X := \mathrm{vert}(\pi)$. Then for any $x^* \in \pi \cap \mathbb{Z}^d$ and $\lambda^* \in \mathbb{Z}_{\geq 0}$ there is an integral vector $\mu \in \mathbb{Z}_{\geq 0}^{\pi \cap \mathbb{Z}^d}$ such that*

(1) $\lambda^* x^* = \sum_{x \in \pi \cap \mathbb{Z}^d} \mu_x x$
(2) $|\mathrm{supp}(\mu) \setminus X| \leq 2^d$
(3) $\mu_x \in \{0, 1\} \ \forall x \notin X$.

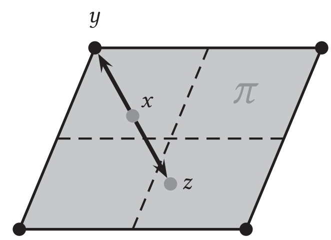


Fig. 5. Weight of y is redistributed to vertex in parallelepiped.

Proof. Let $\pi = \{v_0 + \sum_{i=1}^{k} \alpha_i v_i \mid |\alpha_i| \le 1\ \forall i = 1, \dots, k\}$ where v_0 is the (not necessarily integral) center of π. Consider a vector μ that satisfies (1) and minimizes the potential function $\sum_{x \notin X} \mu_x$ (i.e., the weight that lies on non-vertices of π). We claim that μ also satisfies (2) and (3).

First, consider the case that there is some point $x \in \pi \cap \mathbb{Z}^d$ that is not a vertex and has $\mu_x \ge 2$. We write $x = v_0 + \sum_{i=1}^{k} \alpha_i v_i$ with $|\alpha_i| \le 1$. Let[4] $y := v_0 + \sum_{i=1}^{k} \text{sign}(\alpha_i) \cdot v_i \in \pi \cap \mathbb{Z}^d$ be the vertex of π that we obtain by rounding α_i to ± 1, see Figure 5. The mirrored point $z = x + (x - y) = v_0 + \sum_{i=1}^{k} (2\alpha_i - \text{sign}(\alpha_i)) \cdot v_i$ lies in π as well and is also integral. Here we use the fact that $-1 \le (2\alpha_i - \text{sign}(\alpha_i)) \le 1$. As $x = \frac{1}{2}(y + z)$, we can reduce the weight on x by 2 and add 1 to μ_y and μ_z. We obtain again a vector that satisfies (1), but the weight $\sum_{x \notin X} \mu_x$ has decreased.

So it remains to see what happens when all vectors in $(\pi \cap \mathbb{Z}^d) \setminus X$ carry weight at most 1. Well, if these are at most 2^d, then we are done. Otherwise, we can reiterate the arguments from Lemma 3.4. There will be 2 points of the same parity, which can be joined to create a new point carrying weight at least 2 and part of this weight can be redistributed to a vertex. This shows the claim. ◻

Now we simply combine Lemmas 3.4, 4.1, and 4.3.

Proof of Structure Theorem 2.3. We choose X as the at most $N = m^d d^{O(d)} (\log \Delta)^d$ many vertices of parallelepipeds Π that are constructed in Lemma 4.1 in running time $N^{O(1)}$ (there is an extra 2^d factor, that accounts for the maximum number of vertices per parallelepiped; this is absorbed by the O-notation). Now consider any vector $a \in \text{int.cone}(P \cap \mathbb{Z}^d)$. By Lemma 3.4 there is a vector $\mu \in \mathbb{Z}_{\ge 0}^{P \cap \mathbb{Z}^d}$ with $|\text{supp}(\mu)| \le 2^d$ and $a = \sum_x \mu_x \cdot x$. For every x with $\lambda_x > 0$ we consider a parallelepiped $\pi \in \Pi$ with $x \in \pi \cap \mathbb{Z}^d$. Then we use Lemma 4.3 to redistribute the weight from x to the vertices of π. For each parallelepiped, there are at most 2^d non-vertices with a weight of 1. In the case in which a vector is used by several parallelepipeds, we can further redistribute its weight to the vertices of one of the involved parallelepipeds. This process terminates as the total weight on X keeps increasing. We denote the new solution by λ. As we are using at most 2^d parallelepipeds, we have $|\text{supp}(\lambda) \cap X| \le 2^d \cdot 2^d$ and $|\text{supp}(\lambda) \setminus X| \le 2^d \cdot 2^d$. ◻

5 PROOF OF THE MAIN THEOREM

Now that we have the Structure Theorem, the claim of Theorem 2.2 is easy to show.

Proof of Main Theorem 2.2. As both polyhedra P and Q are assumed to be rational, we can write them as $P = \{x \in \mathbb{R}^d \mid Ax \le b\}$ and $Q = \{x \in \mathbb{R}^d \mid \tilde{A}x \le \tilde{b}\}$ with $A, b, \tilde{A}, \tilde{b}$ integral while the encoding length increases by at most a polynomial factor. We abbreviate $\Delta := \max\{\|A\|_\infty, \|b\|_\infty, 2\}$ and $\tilde{\Delta} := \max\{\|\tilde{A}\|_\infty, \|\tilde{b}\|_\infty, 2\}$.

[4] Recall that $\text{sign}(\alpha) = \begin{cases} 1 & \alpha \ge 0 \\ -1 & \alpha < 0 \end{cases}$

We compute the set X of size at most $N := m^d d^{O(d)}(\log\Delta)^d$ from Theorem 2.3 for the polytope P in time $N^{O(1)}$. Now let $y^* \in \text{int.cone}(P \cap \mathbb{Z}^d) \cap Q$ be an unkown target vector. Then we know by Theorem 2.3 that there is a vector $\lambda^* \in \mathbb{Z}_{\geq 0}^{P\cap\mathbb{Z}^d}$ such that $\sum_{x\in P\cap\mathbb{Z}^d} \lambda_x^* x = y^*$, $|\text{supp}(\lambda^*) \cap X| \leq 2^d$, $|\text{supp}(\lambda^*)\backslash X| \leq 2^d$ and $\lambda_x^* \in \{0,1\}$ for $x \in (P \cap \mathbb{Z}^d)\backslash X$.

After enumerating all subsets of $X' \subseteq X$ of cardinality $|X'| \leq 2^{2d}$ and running the subsequent test for each of them, we may assume to know the set $X' = X \cap \text{supp}(\lambda^*)$. Note that there are at most $N^{2^{2d}}$ sets to test.[5] At the expense of another factor $2^{2d}+1$ we guess the number $k = \sum_{x\notin X'} \lambda_x^* \in \{0,\dots,2^{2d}\}$ of extra points. Now we can set up an integer program with few variables. We use variables λ_x for $x \in X'$ to determine the correct multiplicities of the points in X. Moreover, we have variables $x_1,\dots,x_k \in \mathbb{Z}_{\geq 0}^d$ to determine which extra points to take with unit weight. Additionally we use a variable $y \in \mathbb{Z}^d$ to denote the target vector in polyhedron Q. The ILP is then of the form

$$\begin{array}{rcll} Ax_i & \leq & b & \forall i = 1,\dots,k \\ \sum_{x\in X'} \lambda_x x + \sum_{i=1}^k x_i & = & y & \\ \tilde{A}y & \leq & \tilde{b} & \\ \lambda_x & \in & \mathbb{Z}_{\geq 0} & \forall x \in X' \\ x_i, y & \in & \mathbb{Z}^d & \forall i = 1,\dots,k, \end{array}$$

and given that we made the correct guesses, this system has a solution. The number of variables is $|X'| + (k+1)d \leq 2^{O(d)}$ and the number of constraints is $km + d + \tilde{m} + |X'|d = 2^{O(d)}m + \tilde{m}$ as well. Note that the largest coefficient in the ILP is at most $\Delta' := \max\{d!\cdot\Delta^d, \tilde{\Delta}\}$ as $\|x\|_\infty \leq d!\cdot\Delta^d$ for $x \in X'$ (see Lemma 3.3), as well as $\max\{\|A\|_\infty, \|b\|_\infty\} \leq \Delta$ and $\max\{\|\tilde{A}\|_\infty, \|\tilde{b}\|_\infty\} \leq \tilde{\Delta}$. Hence the ILP can be solved in time $(2^{O(d)})^{2^{O(d)}} \cdot (2^{O(d)}m + \tilde{m})^{O(1)} \cdot (\log\Delta')^{O(1)} \leq 2^{2^{O(d)}} \cdot \text{enc}(P)^{O(1)} \cdot \text{enc}(Q)^{O(1)}$ via Theorem 3.1.

We note that the largest factor in upper bounding the total running time is indeed the term $N^{2^{2d}} \leq \text{enc}(P)^{2^{O(d)}}$ required to guess the correct choice of X'. Multiplying the different terms results in a total running time of the form $\text{enc}(P)^{2^{O(d)}} \cdot \text{enc}(Q)^{O(1)}$. □

Note that the structure theorem uses that integer combination are taken w.r.t. a set $X = P \cap \mathbb{Z}^d$ that is closed under taking convex combinations. However, without any assumption on the structure of X, the test $\text{int.cone}(X) \cap Q \neq \emptyset$ is **NP**-hard even for $d = 1$ and Q being a single point. To see this, recall that given positive integers $a_1,\dots,a_n$ with parameters $k \in \mathbb{N}$ and $S \geq 2$, it is **NP**-hard to decide whether exactly k of the numbers can be added to give exactly S [10]. Then for $S' := 2(a_1 + \cdots + a_n)$, this decision problem is equivalent to $\text{int.cone}(\{S'+a_1,\dots,S'+a_n\}) \cap \{kS'+S\} \neq \emptyset$.

6 HIGH MULTIPLICITY SCHEDULING

In this section, we want to demonstrate the power and versatility of our method by describing, how most scheduling problems with a constant number of job types and a constant number of machine types can be solved in polynomial time. For didactic purposes, we begin with a simple application and then discuss how the approach can be generalized to capture many more scheduling problems.

[5] Actually, we know that X' consists of the vertices of at most 2^d parallelepipeds, thus it suffices to incorporate a factor of N^{2^d}, but the improvement would be absorbed by the O-notation later, anyway.

6.1 Cutting Stock

In the *cutting stock* problem, we are given *item sizes* $s_1, \ldots, s_d \in \mathbb{N}$ and multiplicity $a_j \in \mathbb{N}$ for each item $j \in [d]$. Additionally, we have a list of m *bin types*, where bin type $i \in [m]$ has *capacity* $w_i \in \mathbb{N}$ and *cost* $c_i \in \mathbb{N}$. The task is to assign all the items to bins so that the items assigned to each bin do not exceed its capacity. The objective function is to minimize the cost where we pay an amount of c_i for each bin type i that is being used. The study of this problem goes back at least to the 1960's to the classical paper of Gilmore and Gomory [9]. In particular bin packing is equivalent to the case where $m = 1$.

We will now explain how this problem can be solved using the test from Theorem 2.2. First, using binary search we can reduce the optimization variant of cutting stock to the decision variant where for a given parameter $T \in \mathbb{Z}_{\geq 0}$ we have to decide whether there is a solution of objective function at most T. The intuitive way to model the problem is by defining m *knapsack polytopes* that include a coordinate for the objective function as well as a *target polytope* by setting

$$P_i := \left\{ \begin{pmatrix} x \\ c_i \end{pmatrix} \in \mathbb{R}^{d+1} \mid \sum_{j=1}^{d} x_j s_j \leq w_i \text{ and } x_j \geq 0 \ \forall j \in [d] \right\} \quad \forall i \in [m] \qquad \text{and}$$

$$Q := \left\{ \begin{pmatrix} a \\ t \end{pmatrix} \in \mathbb{R}^{d+1} \mid 0 \leq t \leq T \right\}.$$

Then the cutting stock problem indeed has a solution with objective function value at most T, if and only if

$$\text{int.cone}\Big((P_1 \cup \ldots \cup P_m) \cap \mathbb{Z}^{d+1} \Big) \cap Q \neq \emptyset.$$

The issue with this approach is that in general the union of polytopes is non-convex and Theorem 2.2 only applies to convex polytopes. But as we have integer variables at our disposal, this issue can be fixed using the *Big-M method*.

Theorem 6.1. *The cutting stock problem with d different item types and m different bin types can be solved in time $(\log \Delta)^{2^{O(d+m)}}$ where $\Delta := \max\{\|c\|_\infty, \|w\|_\infty, \|s\|_\infty, 4\}$.*

Proof. We set $M := \max\{w_i, c_i \mid i \in [m]\}$ and define

$$\tilde{P} := \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \mathbb{R}^{d+1+m}_{\geq 0} \mid \begin{array}{rcl} \sum_{j=1}^{d} s_j x_j & \leq & w_i + (1 - z_i)M \ \forall i \in [m] \\ y & \leq & c_i + (1 - z_i)M \ \forall i \in [m] \\ y & \geq & c_i - (1 - z_i)M \ \forall i \in [m] \\ \sum_{i \in [m]} z_i & = & 1 \end{array} \right\} \quad \text{and} \quad \tilde{Q} := \left\{ \begin{pmatrix} a \\ t \\ z \end{pmatrix} \mid 0 \leq t \leq T \right\}.$$

Here, z_i is the decision variable telling whether the vector (x, y, z) originates from bin type i. First, we claim that

$$\tilde{P} \cap \mathbb{Z}^{d+1+m} = \bigcup_{i=1}^{m} \left\{ \begin{pmatrix} x \\ c_i \\ e_i \end{pmatrix} \mid x \in \mathbb{Z}^d_{\geq 0} \text{ and } \sum_{j=1}^{d} s_j x_j \leq w_i \right\}, \tag{2}$$

where e_i is the ith unit vector in $\mathbb{R}^m$. So, consider a vector $(x, y, z) \in \tilde{P} \cap \mathbb{Z}^{d+1+m}$. Then by integrality and the constraint $\sum_{i \in [m]} z_i = 1$, there is a fixed index i with $z = e_i$. From the other constraints in $\tilde{P}$ we can infer that $y = c_i$ and $\sum_{j=1}^{d} s_j x_j \leq w_i + (1 - z_i)M = w_i$. In reverse, we can see that any vector (x, c_i, e_i) that is contained in the right-hand side of Equation (2) satisfies the constraints of $\tilde{P}$. In particular for $i' \neq i$ one has $\sum_{j=1}^{d} s_j x_j \leq w_i \leq w_{i'} + (1 - z_{i'})M$ as well as $c_i \leq c_{i'} + (1 - z_{i'})M$. Hence the test

$$\text{int.cone}\Big(\tilde{P} \cap \mathbb{Z}^{d+1+m} \Big) \cap \tilde{Q} \neq \emptyset$$

correctly decides whether the cutting stock instance has a solution of value at most T. It remains to analyze the running time. The maximum absolute value in the constraint matrices describing $\tilde{P}$ and $\tilde{Q}$ is upper bounded by $\max\{\Delta, 2M, T\} \leq d\Delta^2$, where we use $M \leq \Delta$ and[6] $T \leq \|a\|_1 \cdot \|c\|_\infty \leq d\Delta^2$. Moreover the number of variables is $d+1+m$ and the number of constraints is $O(d+m)$. Then the encoding length of both polytopes are bounded by $\text{enc}(\tilde{P}), \text{enc}(\tilde{Q}) \leq (m+d)^{O(1)} \cdot O(\log(d\Delta))$ and the claim follows from Theorem 2.2. □

From a more abstract point of view, we have solved the cutting stock problem by reducing it to a test of the form

$$\text{int.cone}\Big(X_1 \cup \ldots \cup X_m\Big) \cap Q \neq \emptyset,$$

with $X_i = P_i \cap \mathbb{Z}^d$, where $P_1, \ldots, P_m$ are polytopes and hence convex sets. While this approach was sufficient for cutting stock, it is more restricted than necessary. In fact, we can also handle sets of the form

$$X_i = \Big\{x \in \mathbb{Z}^d \mid \exists y \in \mathbb{Z}^{d_i} : A^i x + B^i y \leq b^i\Big\}.$$

Such sets are called *integer projections* and are extremely powerful. One should note that integer projections are in general not closed under taking convex combinations: It is possible that $x, y \in X_i$ with $\frac{1}{2}x + \frac{1}{2}y$ integral while $(\frac{1}{2}x + \frac{1}{2}y) \notin X_i$. In fact, one can even show that *every* finite set is an integer projection. Whether an approach using integer projections is efficient will be determined largely by the number d_i of extra variables as we will see.

6.2 Integer Conic Combinations of Unions of Integer Projections

Motivated by further applications to high multiplicity scheduling, we now give a generalization of the test from Theorem 2.2 to unions of integer projections.

THEOREM 6.2. *Given rational polytopes $P_1, \ldots, P_m \subseteq \mathbb{R}^{d+d_i}$ and a rational polyhedron $Q \subseteq \mathbb{R}^d$, define $X_i := \{x \in \mathbb{Z}^d \mid \exists y \in \mathbb{Z}^{d_i} : (x, y) \in P_i\}$ and $d_{aux} := \max\{d_i : i \in [m]\}$. Then there is an algorithm that decides correctly whether*

$$\textit{int.cone}(X_1 \cup \ldots \cup X_m) \cap Q \neq \emptyset$$

in time $(\sum_{i=1}^m \textit{enc}(P_i))^{2^{O(d+d_{aux}+m)}} \cdot \textit{enc}(Q)^{O(1)}$. In the affirmative case the algorithm provides a vector $\lambda = (\lambda_{i,x})_{i \in [m], x \in X_i}$ with $\lambda_{i,x} \in \mathbb{Z}_{\geq 0}$ and $(\sum_{i=1}^m \sum_{x \in X_i} \lambda_{i,x} \cdot x) \in Q$. Moreover, the size of the support of λ is bounded by $2^{2(d+d_{aux}+m)+1}$.

PROOF. Let us write $P_i = \{(x, y) \in \mathbb{R}^{d+d_i} \mid A^i x + B^i y \leq b^i\}$ and $Q = \{x \in \mathbb{R}^d \mid A'x \leq b'\}$, again assuming after multiplying with the least common denominator that A^i, B^i, b^i have only integer coefficients for all $i \in [m]$. Without loss of generality we may assume[7] that $d_i = d_{\text{aux}}$ for all $i \in [m]$. We abbreviate $\Delta := \max\{\|A^i\|_\infty, \|B^i\|_\infty, \|b^i\|_\infty : i \in [m]\}$ as the largest coefficient in the representation of any of the polytopes P_i. From Lemma 3.3, we know that each point $(x, y) \in P_i \cap \mathbb{Z}^{d+d_{\text{aux}}}$ satisfies $\|(x, y)\|_\infty \leq (d + d_{\text{aux}})! \cdot \Delta^{d+d_{\text{aux}}}$. Again, we use the Big-$M$ method to convert the union of polytopes into a single polytope. The choice of $M := (d + d_{\text{aux}} + 1) \cdot (d + d_{\text{aux}})! \cdot \Delta^{d+d_{\text{aux}}}$ will be large enough. We set

$$\tilde{P} := \left\{ \Big(x, y, z\Big) \in \mathbb{R}^{d+d_{\text{aux}}+m} \mid A^i x + B^i y \leq b^i + (1 - z_i) \cdot M \ \forall i \in [m]; \ \sum_{i=1}^m z_i = 1; z_i \geq 0 \ \ \forall i \in [m] \right\}$$

[6]We can limit T by the following argument: the cutting stock instance can only be feasible if $\|s\|_\infty \leq \|w\|_\infty$. Then assigning each item to a single bin of maximum capacity results in a solution of cost at most $\|a\|_1 \cdot \|c\|_\infty$.

[7]More formally, this means we can replace each polytope $P_i = \{(x, y) \in \mathbb{R}^{d+d_i} \mid A^i x + B^i y \leq b^i\}$ by a polytope $P_i = \{(x, y) \in \mathbb{R}^{d+d_{\text{aux}}} \mid A^i x + (B^i, \mathbf{0}) y \leq b^i; y_j = 0 \ \forall d_i < j \leq d_{\text{aux}}\}$ without changing feasibility of the problem.

and extend the target polyhedron to $\tilde{Q} := Q \times \mathbb{R}^{d_{\text{aux}}} \times \mathbb{R}^m = \{(x, y, z) \mid Ax \leq b\}$. We can prove that $\tilde{P}$ indeed encodes the union of P_i's:

Claim. *If e_i denotes the ith unit vector in $\mathbb{R}^m$, then one has*

$$\tilde{P} \cap \mathbb{Z}^{d+d_{\text{aux}}+m} = \bigcup_{i=1}^{m} \left\{ \left(x, y, e_i\right) \mid (x, y) \in P_i \cap \mathbb{Z}^{d+d_{\text{aux}}} \right\}. \tag{3}$$

Proof of claim. Let $(x, y, z) \in \tilde{P} \cap \mathbb{Z}^{d+d_{\text{aux}}+m}$. Then by integrality and the constraint $\sum_{i=1}^m z_i = 1$ we know that $z = e_i$ for some index $i \in [m]$. Then one has $A^i x + B^i y \leq b^i + \mathbf{0} \cdot M$ and hence $(x, y) \in P_i \cap \mathbb{Z}^{d+d_{\text{aux}}}$. For the reverse direction, let (x, y, e_i) be contained in the right-hand side of (3). Clearly the constraint $A^i x + B^i y \leq b^i + (1 - z_i)M$ is satisfied. Now consider a constraint for $i' \neq i$. Then one has $A^{i'} x + B^{i'} y \leq d\|A^{i'}\|_\infty \|x\|_\infty + d_{\text{aux}} \|B^{i'}\|_\infty \|y\|_\infty \leq b^{i'} + (1 - z_{i'})M$ using the bound $\|(x, y)\|_\infty \leq (d + d_{\text{aux}})! \cdot \Delta^{d+d_{\text{aux}}}$ and making use of the choice of M. Hence $(x, y, e_i) \in \tilde{P}$ and the claim follows. □

The proven claim implies that the condition $\text{int.cone}(X_1 \cup \ldots \cup X_m) \cap Q \neq \emptyset$ holds if and only if

$$\text{int.cone}(\tilde{P} \cap \mathbb{Z}^{d+d_{\text{aux}}+m}) \cap \tilde{Q} \neq \emptyset. \tag{4}$$

Hence, we can apply Theorem 2.2 to decide Equation (4). In the affirmative case, Theorem 2.2 will return an integer conic combination $\tilde{\lambda}$ satisfying $\sum_{(x,y,z) \in \tilde{P} \cap \mathbb{Z}^{\tilde{d}}} \tilde{\lambda}_{(x,y,z)} \cdot (x, y, z) \in \tilde{Q}$ where the support size is bounded by $2^{2\tilde{d}+1}$ with $\tilde{d} := d + d_{\text{aux}} + m$. Then for $i \in [m]$ and $x \in X_i$, we set $\lambda_{i,x} := \sum_y \tilde{\lambda}_{(x,y,e_i)}$ and return the vector $\lambda = (\lambda_{i,x})_{i \in [m], x \in X_i}$ as the desired integer conic combination.

It remains to estimate the running time that is required by the algorithm behind Theorem 2.2. From the construction of $\tilde{P} \subseteq \mathbb{R}^{\tilde{d}}$ we see that $\text{enc}(\tilde{P}) \leq \sum_{i=1}^m O(\text{enc}(P^i)) + O(m \log(M))$. Finally, $\log(M) \leq (d + d_{\text{aux}})^{O(1)} \log(\Delta)$ is bounded by a polynomial in $\sum_{i=1}^m \text{enc}(P_i)$. The Theorem then follows. □

6.3 A General High Multiplicity Scheduling Framework

Next, we want to formulate a very general scheduling problem that captures most scheduling problems. Then, we will argue that this general problem can be solved in polynomial time if the underlying parameters are constant.

For our scheduling framework we assume to have *job types* $j \in [d]$ with $a_j \in \mathbb{Z}_{\geq 0}$ many copies of type j. Moreover, we have *machine types* $i \in [m]$ with a maximum number $n_i \in \mathbb{Z}_{\geq 0}$ of machines that are available. There are two types of cost: We incur a cost of f_i for each machine of type i that is used, and we pay c_{ij} for each copy of job j that is assigned to a machine of type i.

So far, we have not specified more parameters of the jobs, such as release times, deadlines, and running times, and we also have not specified whether we allow preemption or not. For our general framework, the only requirement that we make is that the configurations of jobs that can be scheduled on a copy of machine i can be described as an integer projection

$$X_i = \left\{ x \in \mathbb{Z}^d_{\geq 0} : \exists y \in \mathbb{Z}^{d_i} : (x, y) \in K_i \right\},$$

where K_i is a polytope. In other words, if $x \in X_i$, then it has to be possible to schedule x_j jobs of type j on a machine of type i. The performance of our method depends on the number of constraints in K_i and most crucially on the number of extra variables d_i that are used.

In particular, this captures the classical settings of preemptive and non-preemptive scheduling with release times, running times and deadlines. Formally, the *general scheduling problem* is to assign the jobs to machines to minimize the cost, which can be written as an integer linear program of the form

$$\min \sum_{i\in[m]} \sum_{x\in X_i} \left(f_i + \left(\sum_{j\in[d]} c_{ij} x_j \right) \right) \lambda_{i,x}, \tag{5}$$

$$\sum_{i\in[m]} \sum_{x\in X_i} \lambda_{i,x} x = a, \tag{6}$$

$$\sum_{x\in X_i} \lambda_{i,x} \leq n_i \quad \forall i \in [m], \tag{7}$$

$$\lambda_{i,x} \in \mathbb{Z}_{\geq 0} \quad \forall i \in [m]\ \forall x \in X_i. \tag{8}$$

The variable $\lambda_{i,x}$ denotes how many machines of type i should be packed with configuration x. The objective function pays a "fixed cost" of f_i for each copy of machine i plus a "variable cost" of c_{ij} for each single job of type j that is assigned to a machine i. The constraint (6) forces that every job copy is assigned and Equation (7) guarantees that we do not use more than n_i copies of machine i.

Theorem 6.3. *Consider a general scheduling problem $\mathcal{I}$ with d many job types, m many machine types represented by $X_i = \{x \in \mathbb{Z}^d_{\geq 0} \mid \exists y \in \mathbb{Z}^{d_i} : (x, y) \in K_i\}$ with rational polytopes K_i. The general scheduling problem can be solved in time $enc(\mathcal{I})^{2^{O(m+d+d_{aux})}}$ where $d_{aux} := \max_{i\in[m]} d_i$ is the maximum number of auxiliary variables and $enc(\mathcal{I})$ is the total encoding length of the vectors c, f and n plus the sum of the encoding lengths of the polytopes K_i for $i \in [m]$.*

Proof. We may assume that all the cost parameters c_{ij} and f_i are integers. It suffices to find a solution of total cost T or determine that there is none. The claim then follows by performing a binary search on T. Let us write $K_i = \{(x, y) \mid A^i x + B^i y \leq b^i\}$. Now, define

$$P_i := \left\{ \left((x, c_i^T x, f_i, e_i), y \right) \in \mathbb{R}^d \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}^m \times \mathbb{R}^{d_i} \mid (x, y) \in K_i \right\}$$

and

$$\tilde{X}_i := \left\{ \left(x, c_i^T x, f_i, e_i \right) \in \mathbb{Z}^{d+2+m} \mid \exists y \in \mathbb{Z}^{d_i} : \left((x, c_i^T x, f_i, e_i), y \right) \in P_i \right\},$$

where e_i is the ith unit vector in $\mathbb{R}^m$. Note that any vector in $\tilde{X}_i$ represents a packing of a machine of type i where we have extra coordinates for variable cost and an extra coordinate for the fixed cost f_i as well as extra coordinates for the vector e_i. Then the integer linear program from (5)–(8) has a solution of value at most T, if and only if

$$\text{int.cone}\left(\tilde{X}_1 \cup \ldots \cup \tilde{X}_m \right) \cap \tilde{Q} \neq \emptyset \quad \text{where} \quad \tilde{Q} = \left\{ \left(a, C, F, \mu \right) \mid \begin{array}{c} C + F \leq T \\ \mu \leq n \end{array} \right\}. \tag{9}$$

Then we apply Theorem 6.2 to decide the condition in Equation (9). For the running time note that the integer projections $\tilde{X}_i$ live in $\mathbb{R}^{d+2+m}$ and require $d_i \leq d_{\text{aux}}$ many extra integer variables. □

6.4 Preemptive Scheduling

In the next two subsections, we want to discuss how the polytopes K_i should be chosen to represent preemptive and non-preemptive schedules. We begin with *preemptive* scheduling (without migration). Note that once the assignment to machines is done, the Earliest-Deadline First policy (EDF) gives an optimum preemptive schedule [5]. This allows to check in polynomial time whether a given set of jobs is schedulable one a single machine, even if the number of job types is not constant.

LEMMA 6.4. *Given jobs $j \in [d]$, each with running time $p_j \in \mathbb{Z}_{\geq 0}$, release time $r_j \in \mathbb{Z}_{\geq 0}$ and deadline $d_j \in \mathbb{Z}_{\geq 0}$. Then the set of multi-sets of such jobs that are EDF-schedulable on a single machine can be described as $X = \{x \in \mathbb{Z}^d_{\geq 0} : x \in K\}$ where K is a polytope with $enc(K) \leq O(d^3 \cdot \log \Delta)$ and $\Delta := \max\{p_j, r_j, d_j, 2 : j \in [d]\}$.*

PROOF. Consider a single machine of type i and a vector $x \in \mathbb{Z}^d_{\geq 0}$ of jobs and we wonder how to determine whether the jobs in x can be scheduled on a single machine, i.e., how to test if the EDF schedule of a set of jobs containing x_j copies of job j will meet all the deadlines. If we consider a time interval $[t_1, t_2]$, then it is clear that the total running time of all jobs that have both, release time and deadline in $[t_1, t_2]$ cannot be larger than the length $t_2 - t_1$; otherwise, the schedule must be infeasible. In fact, for the EDF-scheduling policy, this is also a sufficient condition.[8] Moreover, it is clear that one does not need to consider *all* time intervals, but just those whose end points lie in the set $T := \{r_j, d_j \mid j \in [d]\}$ of critical points. Thus we can define K as the set of vectors $x \in \mathbb{R}^d_{\geq 0}$ such that

$$\sum_{\substack{j \in [d]: \\ r_j,\, d_j \in [t_1,\, t_2]}} x_j p_j \leq t_2 - t_1 \quad \forall t_1, t_2 \in T : t_1 \leq t_2. \tag{10}$$

Observe that a job vector $x \in \mathbb{Z}^d_{\geq 0}$ can be scheduled if and only if $x \in K$. The bound on $\text{enc}(K)$ follows from the fact that K has d variables, $O(d^2)$ constraints and all coefficients are bounded by Δ. □

6.5 Non-preemptive Scheduling

Next, we consider scheduling without preemption. In contrast to the preemptive case, even in the single machine case, finding a feasible non-preemptive schedule is **NP**-hard [10] in general. However, we will see that for fixed number d of job types, the set of feasible schedules allows a compact description:

LEMMA 6.5. *Given jobs $j \in [d]$, each with running time $p_j \in \mathbb{Z}_{\geq 0}$, release time $r_j \in \mathbb{Z}_{\geq 0}$ and deadline $d_j \in \mathbb{Z}_{\geq 0}$. Then the set of multi-sets of such jobs that are schedulable non-preemptively on a single machine can be described as $X = \{x \in \mathbb{Z}^d_{\geq 0} : \exists y \in \mathbb{Z}^{O(d^2)} : (x, y) \in K\}$ with $enc(K) \leq O(d^4 \log \Delta)$ where $\Delta := \max\{p_j, r_j, d_j, 2 : j \in [d]\}$.*

PROOF. While we index the jobs of the instance as $j \in \{1, \ldots, d\}$, it will be notationally convinient to set $r_0 := 0, p_0 := 1, d_0 := \infty$, which allows us to use job 0 to represent any idle time of the machine.

Let $T := \{r_j, d_j \mid j \in [d]\} = \{t_1, \ldots, t_{2d}\}$ be the $2d$ critical points, sorted so that $t_1 \leq \cdots \leq t_{2d}$. The crucial observation is that in a feasible schedule, we can arbitrarily permute jobs that have both start and end time in an interval $[t_k, t_{k+1}]$. Let us imagine that the schedule is *cyclic* in the sense that the schedule processes first some (possibly 0) copies of job type 0, then some (again, possibly 0) jobs of type 1, and so on until type d; then the scheduler starts again with jobs of type 0. The interval from a job 0 interval to the beginning of the next job 0 interval is called a *cycle*. Note that the number of copies of job j that are scheduled in a cycle can very well be 0, so indeed such a cyclic schedule trivially exists. Moreover, we want to restrict that a job of type j is only allowed to be scheduled in a cycle if the *complete* cycle is contained in $[r_j, d_j]$. But again this restriction is achievable as we could split cycles if needed.

[8]This can be easily derived from Hall's condition for the existence of perfect matchings in bipartite graphs and the optimality of EDF.

We claim that there exists a schedule with at most $4d$ many cycles. Following our earlier observation it is clear that whenever 2 cycles are completely contained in some interval $[t_k, t_{k+1}]$ of consecutive points, then we could also join them. Hence we have at most $2d$ cycles contained in intervals of the form $[t_k, t_{k+1}]$ plus at most $2d$ cycles that include a critical point.

We introduce an auxiliary variable y_{jk}, which tells us how many copies of job j are processed in the kth cycle. Additionally we have a binary variable z_{jk} telling us whether jobs of type j can be processed in the kth cycle. Moreover, the kth cycle runs in $[\tau_{k-1}, \tau_k]$ (with $\tau_0 := 0$). Then the polytope K whose integral points correspond to feasible schedules can be defined as

$$\begin{array}{rcll} x_j & = & \sum_{k=1}^{4d} y_{jk} & \forall j \in \{1,\dots,d\} \\ \tau_k & = & \sum_{\ell \leq k} \sum_{j=0}^{d} p_j y_{j\ell} & \forall k \in [4d] \\ y_{jk} & \leq & \Delta \cdot z_{jk} & \forall j \in [d] \;\; \forall k \in [4d] \\ \tau_{k-1} & \geq & r_j - \Delta(1 - z_{jk}) & \forall j \in [d] \;\; \forall k \in [4d] \\ \tau_k & \leq & d_j + \Delta(1 - z_{jk}) & \forall j \in [d] \;\; \forall k \in [4d] \\ y_{jk}, \tau_k & \geq & 0 & \forall j \in \{0,\dots,d\} \; \forall k \in [4d] \\ z_{jk} & \in & [0,1] & \forall j \in [d] \; \forall k \in [4d]. \end{array} \tag{11}$$

For example, if $y_{jk} > 0$, then this forces that $z_{jk} = 1$ and hence $r_j \leq \tau_{k-1} \leq \tau_k \leq d_j$. A vector $x \in \mathbb{Z}^d_{\geq 0}$ can be non-preemptively scheduled if and only if there are integral τ, y, z such that $(x, \tau, y, z) \in K$. Since K has $O(d^2)$ variables and $O(d^2)$ constraints, we obtain the bound $\mathrm{enc}(K) \leq O(d^4 \log \Delta)$. $\square$

One might be tempted to wonder whether the number of variables could be reduced at the expense of more constraints, which might still improve the running time. But for non-preemptive scheduling we run into the problem that the set of vectors x that can be scheduled on a single machine is not closed under taking convex combinations.[9] In fact, some additional variables are necessary to write those vectors as integer projection of a convex set.

6.6 A Scheduling Application

As said earlier, many scheduling problems, that involve a constant number of job types and machine types can be handled by our framework. For the sake of demonstration, let us describe one natural setting:

Corollary 6.6. *Given job types $j \in [d]$ and machine types $i \in [m]$ where job j has machine-dependent release time $r_{ij} \in \mathbb{Z}_{\geq 0}$, deadline $d_{ij} \in \mathbb{Z}_{\geq 0}$ and running time $p_{ij} \in \mathbb{Z}_{\geq 0}$ on a machine of type $i \in [m]$. Moreover we have $a_j \in \mathbb{Z}_{\geq 0}$ copies of job type j and using a copy of a machine of type $i \in [m]$ incurs a cost of $c_i \in \mathbb{Z}_{\geq 0}$. Then one can find an optimum assignment of jobs to machines minimizing the total machine cost under preemptive [non-preemptive, resp.] scheduling in time $(\log \Delta)^{2^{O(d+m)}}$ $[(\log \Delta)^{2^{O(d^2+m)}}$, respectively], where $\Delta := \max\{\|a\|_\infty, \|r\|_\infty, \|d\|_\infty, \|p\|_\infty, 4\}$.*

Proof. For the preemptive case, choose K_i as in Lemma 6.4 with $d_{\mathrm{aux}} = 0$. In the preemptive case, we choose K_i as in Lemma 6.5 with $d_{\mathrm{aux}} = O(d^2)$ auxiliary variables. $\square$

It is not difficult to incorporate other objective functions. For example, if the goal is to *minimize the makespan*, then one can perform a binary search on the target makespan D. In each search step one sets the deadlines d_{ij} to the current value of D and then runs Theorem 6.3 to check feasibility. As another example, consider the case that the objective function is to *minimize the tardy jobs.*

[9] A simple example is the following: Consider a set of $d = 3$ job types with $\{(r_j, d_j, p_j) \mid j = 1, 2, 3\} = \{(0, 300, 150), (100, 102, 1), (200, 202, 1)\}$. The vectors $x' = (2, 0, 0)$ and $x'' = (0, 2, 2)$ can both be scheduled in a non-preemptive way. But the convex combination $\frac{1}{2}(x' + x'') = (1, 1, 1)$ cannot be scheduled.

One can set the machine cost to 0 and add a dummy machine with infinite resources that charges a cost of 1 per scheduled job. Then the minimum cost solution will try do maximize the jobs that are assigned to the regular machines. This concludes the discussion on scheduling.

7 FINDING INTEGER CONIC COMBINATIONS FOR UNBOUNDED POLYHEDRA

So far we have only considered integer combinations $\text{int.cone}(P \cap \mathbb{Z}^d)$ where P was a *bounded* polyhedron. Now we will discuss how that boundedness assumption can be removed. We will see that P can be decomposed so that the test $\text{int.cone}(P \cap \mathbb{Z}^d) \cap Q \neq \emptyset$ is equivalent to a test where points come from the union of two bounded polytopes. Such a test can then be decided by our Theorem 6.2.

THEOREM 7.1. *Let $P, Q \subseteq \mathbb{R}^d$ be rational polyhedra. Then the condition $\text{int.cone}(P \cap \mathbb{Z}^d) \cap Q \neq \emptyset$ can be decided in time $enc(P)^{2^{O(d)}} \cdot enc(Q)^{O(1)}$. In the affirmative case, a vector $(\lambda_x)_{x \in P \cap \mathbb{Z}^d}$ of support $2^{O(d)}$ with $\sum_{x \in P \cap \mathbb{Z}^d} \lambda_x x \in Q$ can be computed in the same time.*

PROOF. We write $P = \{x \in \mathbb{R}^d \mid Ax \leq b\}$. Again we may also assume that A and b have only integral entries and set $\Delta := \max\{\|A\|_\infty, \|b\|_\infty, 2\}$. The *Minkowski-Weil Theorem* (see, e.g., Reference [27]) tells us that one can decompose $P = S + C$ where $S \subseteq \mathbb{R}^d$ is a polytope and $C = \{x \in \mathbb{R}^d \mid Ax \leq \mathbf{0}\}$ is the *characteristic cone* of P. As P is rational, a valid choice[10] is $S = P \cap [-M, M]^d$ where $M := d!\Delta^d$. Also the cone C is rational and hence $C = \text{cone}\{a_1, \ldots, a_N\}$ for some integer vectors $a_1, \ldots, a_N \in \mathbb{Z}^d$. Similarly to before, one can argue that coefficients of size $\|a_i\|_\infty \leq M$ suffice[11] for all $i \in [N]$. The next step is to restrict the cone to a bounded region by setting $R := C \cap [-dM, dM]^d$. In the following, we abbreviate $\text{int.cone}_+(X) := \{\sum_{x \in X} \lambda_x x \mid \lambda_x \in \mathbb{Z}_{\geq 0} \text{ and } \lambda \neq \mathbf{0}\}$ as the integer conic combinations of a set X where the combination with $\lambda = \mathbf{0}$ is excluded. Now we can analyze the decomposition:

CLAIM I. *One has $\text{int.cone}_+(P \cap \mathbb{Z}^d) = \text{int.cone}_+((S + R) \cap \mathbb{Z}^d) + \text{int.cone}(R \cap \mathbb{Z}^d)$.*

PROOF OF CLAIM I. For the direction "$\subseteq$," we will show that $P \cap \mathbb{Z}^d \subseteq \text{int.cone}_+((S + R) \cap \mathbb{Z}^d) + \text{int.cone}(R \cap \mathbb{Z}^d)$ holds. Consider a point $x \in P \cap \mathbb{Z}^d$ and write it as $x = s + c$ where $s \in S$ and $c \in C$ (here neither s nor c is guaranteed to be integral). Then by the definition of the cone C, we can write $c = \sum_{i=1}^N \mu_i a_i$ for some coefficients $\mu_i \in \mathbb{R}_{\geq 0}$ where by *Carathéodory's Theorem* (again see, e.g., Reference [27]) we may assume that $|\text{supp}(\mu)| \leq d$. We write

$$c = \sum_{i=1}^N \mu_i a_i = \underbrace{\sum_{i=1}^N (\mu_i - \lfloor \mu_i \rfloor) a_i}_{=:r} + \underbrace{\sum_{i=1}^N \lfloor \mu_i \rfloor a_i}_{=:c_{\text{int}}},$$

where $c_{\text{int}} \in \text{int.cone}\{a_1, \ldots, a_N\} \subseteq \text{int.cone}(R \cap \mathbb{Z}^d)$. Moreover, we know that $r \in R$, because $\sum_{i=1}^N (\mu_i - \lfloor \mu_i \rfloor) \leq d$ and $\|a_i\|_\infty \leq M$ for $i \in [N]$. Then $x = (s + r) + c_{\text{int}}$ where x and c_{int} are integer vectors and so $(s + r) \in (S + R) \cap \mathbb{Z}^d$.

---

[10]If P has vertices, then by Lemma 3.3, S contains all the vertices and the result is immediate. However, it is possible that P contains lines. Hence, we consider a maximal index set $I \subseteq [d]$ so that the restriction $P' = \{x \in P \mid x_i = 0 \ \forall i \in I\}$ still satisfies $P' + C = P$. Then P' does not contain a line while it intersects every minimal face of P. The minimal faces of P' are vertices and every $x \in \text{vert}(P')$ has $\|x\|_\infty \leq d!\Delta^d = M$. Then $\text{vert}(P') \subseteq S$, which concludes the argument.

[11]Consider the polytope $C_{\text{bounded}} := C \cap [-1, 1]^d$. We know that the vertices of C_{bounded} span the cone, i.e., $\text{cone}(\text{vert}(C_{\text{bounded}})) = C$. While the vertices of C_{bounded} are in general not integral, we can scale them to become integral with bounded coefficients. More precisely by Cramer's rule, every $x \in \text{vert}(C_{\text{bounded}})$ is of the form $x = (\frac{\det(R_1)}{\det(T)}, \ldots, \frac{\det(R_d)}{\det(T)})$ where $R_1, \ldots, R_d, T$ are $d \times d$ matrices filled with entries from $\{-\Delta, \ldots, \Delta\}$. Then, $\det(T) \cdot x \in \mathbb{Z}^d$ and moreover $\|\det(T) \cdot x\|_\infty = \max_{i \in [d]} |\det(R_i)| \leq d!\Delta^d = M$.

For the direction "⊇," take integer conic combinations $\sum_{x\in(S+R)\cap\mathbb{Z}^d}\lambda_x x$ with $\lambda_x\in\mathbb{Z}_{\geq 0}$ and $\lambda\neq\mathbf{0}$ and $\sum_{y\in R\cap\mathbb{Z}^d}\mu_y y$ with $\mu_y\in\mathbb{Z}^d_{\geq 0}$. Fix a vector $x^*\in(S+R)\cap\mathbb{Z}^d$ with $\lambda_{x^*}\geq 1$ (this is where we need that $\lambda\neq\mathbf{0}$!). Set $x^{**}:=x^*+\sum_{y\in R\cap\mathbb{Z}^d}\mu_y y$ and note that $x^{**}\in P\cap\mathbb{Z}^d$. Define a new integer conic combination $\tilde{\lambda}$ by

$$\tilde{\lambda}_{x^{**}}:=\lambda_{x^{**}}+1,\quad \tilde{\lambda}_{x^*}:=\lambda_{x^*}-1,\quad \tilde{\lambda}_x:=\lambda_x\ \forall x\in((S+R)\cap\mathbb{Z}^d)\setminus\{x^*,x^{**}\},\quad \tilde{\lambda}_x:=0 \text{ otherwise.}^{12}$$

In words, we have moved all the weight from points in $R\cap\mathbb{Z}^d$ to a single copy of a single point in $P\cap\mathbb{Z}^d$. Then $\sum_{x\in P\cap\mathbb{Z}^d}\tilde{\lambda}_x x=\sum_{x\in(S+R)\cap\mathbb{Z}^d}\lambda_x x+\sum_{y\in R\cap\mathbb{Z}^d}\mu_y y$, which shows Claim I. □

It appears that the overall Theorem almost follows by applying Theorem 6.2, if there was not the slight issue that Claim I contains the expression $\text{int.cone}_+(\ldots)$, which forbids the all-0 coefficients. However, this issue can be fixed by extending the dimension of the polytopes by 1. In particular we will use $P\times\{1\}=\{\binom{x}{1}\mid x\in P\}$, which is a $(d+1)$-dimensional polyhedron. Note that if $\mathbf{0}\in Q$, then trivially $\text{int.cone}(P\cap\mathbb{Z}^d)\cap Q\neq\emptyset$. Hence we may assume from now on that $\mathbf{0}\notin Q$. Our goal is to reformulate the target condition

$$\text{int.cone}(P\cap\mathbb{Z}^d)\cap Q\neq\emptyset. \tag{12}$$

First, note that as $\mathbf{0}\notin Q$, the coefficient vector $\lambda=\mathbf{0}$ is useless for Equation (12) and so Equation (12) is equivalent to

$$\text{int.cone}_+\Big((P\times\{1\})\cap\mathbb{Z}^{d+1}\Big)\cap\Big(Q\times[1,\infty)\Big)\neq\emptyset. \tag{13}$$

Then by Claim I, Equation (13) is equivalent to

$$\Big(\text{int.cone}_+\Big(((S+R)\times\{1\})\cap\mathbb{Z}^{d+1}\Big)+\text{int.cone}\Big((R\times\{0\})\cap\mathbb{Z}^{d+1}\Big)\Big)\cap\Big(Q\times[1,\infty)\Big)\neq\emptyset. \tag{14}$$

Then Equation (14) is equivalent to

$$\text{int.cone}\Big(\Big(((S+R)\times\{1\})\cap\mathbb{Z}^{d+1}\Big)\cup\Big((R\times\{0\})\cap\mathbb{Z}^{d+1}\Big)\Big)\cap\Big(Q\times[1,\infty)\Big)\neq\emptyset, \tag{15}$$

where the extra coordinate enforces that at least one point from $((S+R)\times\{1\})\cap\mathbb{Z}^{d+1}$ needs to have positive weight in any valid integer conic combination. Now we call Theorem 6.2 to decide condition (15) using that $(S+R)\times\{1\}$ and $R\times\{0\}$ are bounded. Note that in the affirmative case, an integer conic combination satisfying Equation (15) can be transformed into one using points of $P\cap\mathbb{Z}^d$ using the argument from Claim I.

For the running time, note that Theorem 6.2 is called for $(d+1)$-dimensional polytopes whose encoding length is bounded polynomially in $\text{enc}(P)$. □

8 THE EISENBRAND-SHMONIN THEOREM IS TIGHT

In this section, we want to describe an example that shows that the Eisenbrand-Shmonin result described in Lemma 3.4 is tight up to a factor of 2. To the best of our knowledge, this was not known before. Fix a dimension $d\geq 2$ and let $k:=2^{d-1}$. Let us define a set $X\subset\mathbb{Z}^d$ of k points as

$$\Big\{\Big(1+x_1,\ldots,1+x_{d-1},(4k)^{1+\sum_{i=1}^{d-1}2^{i-1}x_i}\Big)\mid x_i\in\{0,1\}\ \ \forall i\in[d-1]\Big\}.$$

For example, for $d=3$, we obtain

$$\begin{pmatrix}1\\1\\(4k)\end{pmatrix},\begin{pmatrix}2\\1\\(4k)^2\end{pmatrix},\begin{pmatrix}1\\2\\(4k)^3\end{pmatrix},\begin{pmatrix}2\\2\\(4k)^4\end{pmatrix}.$$

[12] In the fringe case where $x^{**}=x^*$, set $\tilde{\lambda}_x:=\lambda_x$ for all x.

We sort $X = \{a_1, \dots, a_k\}$ according to their length, i.e., $\|a_i\|_\infty = (4k)^i$ and define $P := \mathrm{conv}(X)$.

Lemma 8.1. *The integer conic combination $y := a_1 + \cdots + a_k$ is unique, thus there are 2^{d-1} points necessary to obtain y as an integer conic combination of points in $P \cap \mathbb{Z}^d$.*

Proof. First, we observe that there is no other integer point in the convex hull P because of the first $d-1$ coordinates of the a_i's. In other words $P \cap \mathbb{Z}^d = X$. We want to argue that due to the enormous growth of the last coordinate, each a_i has to be used exactly once to obtain y. For this sake, consider any integer conic combination

$$y = \sum_{i=1}^{k} \lambda_i a_i.$$

Note that $y_j = 3 \cdot 2^{d-2}$ for $j \in \{1, \dots, d-1\}$, thus $0 \le \lambda_i \le 3 \cdot 2^{d-2} < 2k$ for $i = 1, \dots, k$. We want to argue that $\lambda_1 = \ldots = \lambda_k = 1$ is the only possibility. Suppose this is not the case and let i^* be the largest index with $\lambda_{i^*} \neq 1$. We will see that if $\lambda_{i^*} = 0$, then the combined vector is too short and if $\lambda_{i^*} \ge 2$, then it is too long. More formally, we inspect the difference

$$\Big\| \sum_{i=1}^{k} (\lambda_i - 1) a_i \Big\|_\infty \ \ge\ |\lambda_{i^*} - 1| \cdot \|a_{i^*}\|_\infty - \sum_{i=1}^{i^*-1} \underbrace{\lambda_i}_{\le 2k} \cdot \underbrace{\|a_i\|_\infty}_{=(4k)^i}, \tag{16}$$

$$\ge\ |\lambda_{i^*} - 1| \cdot (4k)^{i^*} - \sum_{i=1}^{i^*-1} 2k \cdot (4k)^i, \tag{17}$$

$$\ge\ (4k)^{i^*} \cdot \Big(|\lambda_{i^*} - 1| - \frac{1}{2} \underbrace{\Big(\sum_{j \ge 0} (4k)^{-j} \Big)}_{\le 3/2} \Big), \tag{18}$$

$$\ge\ \underbrace{\Big(|\lambda_{i^*} - 1| - \frac{3}{4} \Big)}_{\ge 1/4} \cdot (4k)^{i^*} > 0. \tag{19}$$

In Equation (16) we use the reverse triangle inequality and the fact that $|\lambda_i - 1| = 0$ for $i > i^*$. In Equation (17) we make use of $\lambda_i \le 2k$ and $\|a_i\|_\infty = (4k)^i$. To bound the sum in Equation (18) we use that $2k \ge 4$. Finally, in Equation (19) we reach the conclusion of $\Big\| (\sum_{i=1}^{k} \lambda_i a_i) - y \Big\|_\infty > 0$, which is a contradiction to the assumption that λ is a valid integer conic combination for y. □

Note that $\|a_k\|_\infty = (4k)^k = 2^{\Theta(d 2^d)}$, hence our construction uses numbers that are doubly-exponentional in d. An argument of Reference [6] based on the pigeonhole principle shows that for a set $X \subseteq \mathbb{Z}^d$, every point in int.cone(X) admits an integer conic combination whose support size is bounded by $O(d \cdot \log(dM))$, where $M := \max\{\|x\|_\infty \mid x \in X\}$. In other words, any set of integral vectors X where some vector int.cone(X) requires a support of size $\Omega(2^d)$ must have $M \ge 2^{\Omega(2^d)}$. Hence the doubly exponentially large numbers are indeed necessary.

Follow-up work. After publication of the conference version of this article, Jansen and Klein [14] proved a modification of the Structure Theorem 2.3 where the points X are simply the vertices of $\mathrm{conv}(P \cap \mathbb{Z}^d)$. Interestingly, in order for that Theorem to be true, the used number of points from $(P \cap \mathbb{Z}^d) \setminus X$ counted with multiplicity has to be increased from $2^{\Theta(d)}$ to $2^{2^{\Theta(d)}}$. Their main algorithmic result is the following:

Theorem 8.2 (Jansen-Klein [14]). *Given rational polytopes $P, Q \subseteq \mathbb{R}^d$, one can find a vector $y \in$ int.cone$(P \cap \mathbb{Z}^d) \cap Q$ and a vector $\lambda \in \mathbb{Z}_{\geq 0}^{P \cap \mathbb{Z}^d}$ in time $|vert(P_I)|^{2^{O(d)}} \cdot enc(P)^{O(1)} \cdot enc(Q)^{O(1)}$ such that $y = \sum_{x \in P \cap \mathbb{Z}^d} \lambda_x x$, or decide that no such y exists.*

Here vert(P_I) are the extreme points of the convex hull of integers points in P. Plugging in the worst case bounds of Cook et al. [4, 11] does not improve over our running time of $O(\log \Delta)^{2^{O(d)}}$ even for bin packing in d dimensions. However, one might imagine applications where the polytopes in question are guaranteed to have rather few vertices so that the approach of Reference [14] dominates.

Open problems. A natural open problem that arises from this work is whether the double exponential running time is necessary. We pose it as an open problem whether bin packing in d dimensions can be solved in time $f(d) \cdot O(\log(\Delta))^{O(1)}$, where $f(d)$ is an arbitrary function depending on the dimension. Phrased differently, we ask whether bin packing is *fixed-parameter trackable* with the dimension as parameter.

We point out that there are problems in high multiplicity scheduling that remain unsolved. For example, imagine that we are given d different rectangles with dimensions $w_i \times h_i$ and multiplicity n_i as well as a $W \times H$ sized bin. The question is whether all the rectangles can be packed into this bin (without rotating the rectangles). Schiermeyer [26] conjectures that this problem is solvable in polynomial time if d is fixed. Our framework does not seem to apply (at least not without structural insights into the optimum rectangle packing).

REFERENCES

[1] I. Bárány, R. Howe, and L. Lovász. 1992. On integer points in polyhedra: A lower bound. *Combinatorica* 12, 2 (1992), 135–142. DOI: https://doi.org/10.1007/BF01204716

[2] R. C. Buck. 1943. Partition of space. *Am. Math. Month.* 50, 9 (1943), 541–544. DOI: https://doi.org/10.1080/00029890.1943.11991447

[3] S. Boyd and L. Vandenberghe. 2014. *Convex Optimization.* Cambridge University Press. DOI: https://doi.org/10.1017/CBO9780511804441

[4] W. J. Cook, M. Hartmann, R. Kannan, and C. McDiarmid. 1992. On integer points in polyhedra. *Combinatorica* 12, 1 (1992), 27–37.

[5] M. L. Dertouzos. 1974. Control robotics: The procedural control of physical processes. In *Proceedings of the IFIP Congress.* 807–813.

[6] F. Eisenbrand and G. Shmonin. 2006. Carathéodory bounds for integer cones. *Oper. Res. Lett.* 34, 5 (2006), 564–568.

[7] C. Filippi and A. Agnetis. 2005. An asymptotically exact algorithm for the high-multiplicity bin packing problem. *Math. Program.* 104, 1 (2005), 21–37.

[8] C. Filippi. 2007. On the bin packing problem with a fixed number of object weights. *Eur. J. Operat. Res.* 181, 1 (2007), 117–126.

[9] P. C. Gilmore and R. E. Gomory. 1961. A linear programming approach to the cutting-stock problem. *Operat. Res.* 9, 6 (1961), 849–859.

[10] M. R. Garey and D. S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Company, New York, NY.

[11] M. Hartmann. 1988. Cutting Planes and the Complexity of the Integer Hull. Retrieved from http://hdl.handle.net/1813/8702.

[12] R. Hoberg and T. Rothvoss. 2017. A logarithmic additive integrality gap for bin packing. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'17).* 2616–2625. DOI: https://doi.org/10.1137/1.9781611974782.172

[13] D. Hochbaum and R. Shamir. 1991. Strongly polynomial algorithms for the high multiplicity scheduling problem. *Operat. Res.* 39, 4 (1991), 648–653. http://www.jstor.org/stable/171172

[14] K. Jansen and K. Klein. 2017. About the structure of the integer cone and its application to bin packing. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'17).* Society for Industrial and Applied Mathematics, Philadelphia, PA, 1571–1581.

[15] F. John. 1948. Extremum problems with inequalities as subsidiary conditions. In *Studies and Essays Presented to R. Courant on his 60th Birthday, January 8, 1948.* Interscience Publishers, Inc., New York, NY, 187–204.

[16] D. S. Johnson. 1992. The NP-completeness column: An ongoing guide: The tale of the second prover. *J. Algor.* 13, 3 (September 1992), 502–524.

[17] K. Jansen and R. Solis-Oba. 2011. A simple OPT+1 algorithm for cutting stock under the modified integer round-up property assumption. *Inf. Process. Lett.* 111, 10 (2011), 479–482. DOI: https://doi.org/10.1016/j.ipl.2011.02.009

[18] K. Jansen and R. Solis-Oba. 2010. An OPT + 1 algorithm for the cutting stock problem with constant number of object lengths. In *Proceedings of the 14th International Conference on Integer Programming and Combinatorial Optimization (IPCO'10)*, Friedrich Eisenbrand and F. Bruce Shepherd (Eds.). 438–449.

[19] R. Kannan. 1987. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.* 12, 3 (August 1987), 415–440. DOI: https://doi.org/10.1287/moor.12.3.415

[20] N. Karmarkar and R. M. Karp. 1982. An efficient approximation scheme for the one-dimensional bin-packing problem. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*. IEEE, New York, 312–320.

[21] B. Korte and J. Vygen. 2002. *Combinatorial Optimization—Theory and Algorithms* (2nd ed.). Springer-Verlag.

[22] H. W. Lenstra, Jr. 1983. Integer programming with a fixed number of variables. *Math. Oper. Res.* 8, 4 (1983), 538–548. DOI: https://doi.org/10.1287/moor.8.4.538

[23] S. T. McCormick, S. R. Smallwood, and F. C. R. Spieksma. 1997. Polynomial algorithms for multiprocessor scheduling with a small number of job lengths. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'97)*. 509–517.

[24] S. Thomas McCormick, Scott R. Smallwood, and Frits C. R. Spieksma. 2001. A polynomial algorithm for multiprocessor scheduling with two job lengths. *Math. Oper. Res.* 26, 1 (2001), 31–49. DOI: https://doi.org/10.1287/moor.26.1.31.10590

[25] T. Rothvoss. 2013. Approximating bin packing within O(log OPT * Log Log OPT) bins. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS'13)*. 20–29. DOI: https://doi.org/10.1109/FOCS.2013.11

[26] I. Schiermeyer. 1994. Reverse-fit: A 2-optimal algorithm for packing rectangles. In *Proceedings of the 2nd Annual European Symposium on Algorithms (ESA'94)*. Springer-Verlag, Berlin, 290–299. http://dl.acm.org/citation.cfm?id=647904.739467

[27] A. Schrijver. 1999. *Theory of Linear and Integer Programming*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley.

Received February 2018; revised April 2020; accepted August 2020