Operations Research Letters 49 (2021) 908–913

Contents lists available at ScienceDirect

Operations Research Letters

www.elsevier.com/locate/orl

Parameterized complexity of configuration integer programs

Dušan Knop [a,*,1], Martin Koutecký [b,2], Asaf Levin [c,3], Matthias Mnich [d,4], Shmuel Onn [c,5]

[a] *Department of Theoretical Computer Science, Faculty of Information Technology, Czech Technical University in Prague, Prague, Czech Republic*
[b] *Charles University, Prague, Czech Republic*
[c] *Technion – Israel Institute of Technology, Haifa, Israel*
[d] *Hamburg University of Technology, Institute for Algorithms and Complexity, Hamburg, Germany*

ARTICLE INFO

Article history:
Received 6 August 2021
Received in revised form 31 October 2021
Accepted 2 November 2021
Available online 15 November 2021

Keywords:
Parameterized algorithms
Configuration IP
Surfing

ABSTRACT

Configuration integer programs (IP) have been key in the design of algorithms for NP-hard high-multiplicity problems. First, we develop fast exact (exponential-time) algorithms for Configuration IP and matching hardness results. Second, we showcase the implications of these results to bin-packing and facility-location-like problems.

1. Introduction

In 1961, Gilmore and Gomory [11] introduced the fundamental and widely influential notion of Configuration IP (ConfIP for short), and applied it to the Bin Packing problem. In Bin Packing, one is given a set $\mathcal{J}$ of n items with sizes $p_1, \ldots, p_n \in \mathbb{N}$, which need to be assigned (or *packed*) to a set of m identical bins of a common capacity $T \in \mathbb{N}$ such that the total size of items packed per bin is not larger than its capacity. This classical problem has long been investigated from the perspective of approximation algorithms [13,20], and in recent years also from the perspective of exact algorithms [3,12,16,18]. Gilmore and Gomory used the ConfIP to describe a Bin Packing solution by a list of tuples "(packing of one bin s, multiplicity μ of bins with packing s)". In this paper, we consider a more general form of Configuration IP. For motivation, observe that the natural input encoding for Bin Packing does not list the item sizes one by one; rather, the n items are classified into $d \ll n$ *item types* and the input specifies the size $p_j \in \mathbb{N}$ and the number n_j of items of type j. The so-called *high-multiplicity encoding* (a term first coined by Hochbaum and Shamir [14]) thus gives a size vector $\mathbf{p} = (p_1, \ldots, p_d)$ and a *multiplicity vector* $\mathbf{n} = (n_1, \ldots, n_d)$ with $\|\mathbf{n}\|_1 = n_1 + \cdots + n_d = n$. Observe now that, for any packing σ, each bin i defines a vector $\mathbf{x} = (x_1, \ldots, x_d)$, with x_j being the number of items of type j assigned to i by σ; such a vector $\mathbf{x}$ is called a *configuration*. This high-multiplicity bin packing problem (and many other optimization problems) can be rephrased in the following way, as was first observed by Gilmore and Gomory [11]. Let $\mathbb{N}$ be the set $\{0, 1, \ldots\}$. Let $\mathcal{C}_T = \left\{\mathbf{x} \in \mathbb{N}^d \mid \mathbf{px} \le T\right\}$ denote the set of configurations of size at most T. Then, to decide if the given instance admits a feasible solution amounts to deciding if $\mathbf{n}$ can be written as a sum of m configurations from $\mathcal{C}_T$, i.e., $\mathbf{n} = \sum_{i=1}^m \mathbf{x}^i$, with $\mathbf{x}^i \in \mathcal{C}_T$ for all $i = 1, \ldots, m$. This task leads to a relatively simple form of *configuration IP*, which has an integral variable $\lambda_{\mathbf{x}} \in \mathbb{N}$ for each configuration $\mathbf{x} \in \mathcal{C}_T$, and asks for a solution with $\sum_{\mathbf{x} \in \mathcal{C}_T} \lambda_{\mathbf{x}} = m$ and $\sum_{\mathbf{x} \in \mathcal{C}_T} \lambda_{\mathbf{x}} \cdot \mathbf{x} = \mathbf{n}$.

However, other optimization problems lead to more complex forms of Configuration IP. Maybe there are bins of different types, in which case τ denotes the number of *bin types* with μ^i being the number of bins of type i, where bins of a common type have common characteristics. Items may also have several other characteristics, and then an item type is the set of items with common characteristics. We may also incur a cost for each configuration. Thus, our Configuration IP is defined as follows.

* Corresponding author.
E-mail addresses: dusan.knop@fit.cvut.cz (D. Knop), koutecky@iuuk.mff.cuni.cz (M. Koutecký), levinas@ie.technion.ac.il (A. Levin), matthias.mnich@tuhh.de (M. Mnich), onn@ie.technion.ac.il (S. Onn).

1 Partially supported by the OP VVV MEYS funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics".
2 Partially supported by Charles University project UNCE/SCI/004, and by the project 19-27871X of GA ČR.
3 Partially supported by Israel Science Foundation grant 308/18.
4 Supported by DFG grant MN 59/4-1.
5 Partially supported by the Dresner chair and Israel Science Foundation grant 308/18.

https://doi.org/10.1016/j.orl.2021.11.005
0167-6377/

Configuration IP (ConfIP)
Input: Dimension $d \in \mathbb{N}$, finite sets $X^1, \ldots, X^\tau \subseteq \mathbb{Z}^d$, objective functions $f^1, \ldots, f^\tau : \mathbb{Z}^d \to \mathbb{Z}$, numbers $\mu^1, \ldots, \mu^\tau \in \mathbb{N}$, and a target vector $\mathbf{n} \in \mathbb{Z}^d$.
Find: $\min \big\{ \sum_{i=1}^{\tau} \sum_{\mathbf{x} \in X^i} f^i(\mathbf{x}) \cdot \lambda^i_{\mathbf{x}} \mid \sum_{i=1}^{\tau} \sum_{\mathbf{x} \in X^i} \mathbf{x} \cdot \lambda^i_{\mathbf{x}} = \mathbf{n},\ \sum_{\mathbf{x} \in X^i} \lambda^i_{\mathbf{x}} = \mu^i,\ \lambda^i_{\mathbf{x}} \in \mathbb{N}\ \forall \mathbf{x} \in X^i\ \forall i \in \{1, \ldots, \tau\} \big\}$.

We refer to van den Akker et al. [26] for practical investigations of ConfIP, and to Hochbaum and Shmoys [15], Alon et al. [1], Fernandez de la Vega and Lueker [9] and Karmarker and Karp [20] for studies of approximation algorithms based on ConfIP. Previous studies of the bin packing problem in the high-multiplicity settings were done by Jansen and Solis-Oba [17] and Jansen and Klein [16].

Our contributions and paper outline. Our contribution is two-fold. First, in Section 3 we provide several fixed-parameter algorithms, and in Section 4 we prove hardness results for ConfIP, delineating the complexity landscape with regard to the most natural parameters showing that some trade-offs evident in our analysis are inevitable. Second, to showcase the usefulness and versatility of our approach, we apply in Section 5 our algorithms to high multiplicity problems in bin packing and surfing, a general model of facility location and multicommodity flows.

2. Preliminaries

For positive integers m, n we set $[m, n] = \{m, m+1, \ldots, n\}$ and $[n] = [1, n]$. We write vectors in boldface (e.g., $\mathbf{x}, \mathbf{y}$) and their entries in normal font (e.g., the i-th entry of $\mathbf{x}$ is x_i or $x(i)$). For $\alpha \in \mathbb{R}$, $\lfloor \alpha \rfloor$ is the floor of α, and $\lceil \alpha \rceil$ is the ceiling of α. For data object O, we denote by $\langle O \rangle$ its binary encoding length. The set $\mathbb{N}$ is the set of non-negative integers, that is, $\mathbb{N} = \{0, 1, 2, \ldots\}$.

The input of ConfIP can be given explicitly only in fairly limited scenarios. Thus, we assume that each (possibly very large) set X^i is defined succinctly. The following definition captures the case when each X^i is defined as a projection of integer points of a rational polytope.

Definition 1 *(P-representation).* For $i = 1, \ldots, \tau$, let $P^i \subseteq \mathbb{R}^{d+d^i}$ be a polytope and let $\pi^i((\mathbf{x}, \mathbf{x}')) = \mathbf{x} \in \mathbb{R}^d$ be a projection discarding the last d^i coordinates. We call the collection $P^1, \ldots, P^\tau$ a *P-representation* of $X^1, \ldots, X^\tau$ if $X^i = \pi^i(P^i) \cap \mathbb{Z}^d$, for each $i \in [\tau]$. Let each P^i by defined as $P^i = \{(\mathbf{x}, \mathbf{x}') \mid A^i(\mathbf{x}, \mathbf{x}') \le \mathbf{b}^i\}$ for some $A^i \in \mathbb{Z}^{m^i \times (d+d^i)}$ and $\mathbf{b}^i \in \mathbb{Z}^{m^i}$. The *parameters of a P-representation* are the following quantities: $M = \max_{i \in [\tau]} m^i$, $D = \max_{i \in [\tau]} d^i$, $\Delta = \max_{i \in [\tau]} \|A^i\|_\infty$, $L = \langle \Delta, \mathbf{b}^1, \ldots, \mathbf{b}^\tau \rangle$.

We consistently use superscripts to refer to objects and quantities related to the types (e.g., $X^i, d^i, f^i, \ldots$). To avoid confusion, we always use parentheses when intending to express exponentiation (e.g., $(d^i)^2$). With each X^i given implicitly, the objective functions f^i also must have implicit representations, or be given by oracles. We consider the following classes of objective functions:

- linear: given vectors $\mathbf{w}^1, \ldots, \mathbf{w}^\tau \in \mathbb{Z}^d$, let $f^i(\mathbf{x}) = \mathbf{w}^i \mathbf{x}$.
- convex: each $f^i(\mathbf{x})$ is a convex function.
- extension-separable convex: each
$$f^i(\mathbf{x}) = \min_{\mathbf{x}' : (\mathbf{x}, \mathbf{x}') \in P^i \cap \mathbb{Z}^{d+d'}} g^i(\mathbf{x}, \mathbf{x}')$$
for g^i a separable convex function. (In some of our applications the objective is only expressible as a separable convex function in terms of the auxiliary variables $\mathbf{x}'$.)
- concave: each $f^i(\mathbf{x})$ is a concave function.
- fixed-charge: each $f^i(\mathbf{x}) = c^i \in \mathbb{N}$ if $\mathbf{x} \neq \mathbf{0}$ and $f^i(\mathbf{x}) = 0$ otherwise; we call c^i a *penalty*.

For a ConfIP instance given in its P-representation, set
$$f_{\max} = \max_{i \in [\tau]} \max_{(\mathbf{x}, \mathbf{x}') \in \mathbb{Z}^{d+d^i} : A^i(\mathbf{x}, \mathbf{x}') \le \mathbf{b}^i} \left| f^i(\mathbf{x}) \right|.$$

ConfIP as N-fold IP. Below, we connect ConfIP with a special class of integer programs (IPs). The baseline IP is: $\min f(\mathbf{x}) : A\mathbf{x} = \mathbf{b},\ \mathbf{l} \le \mathbf{x} \le \mathbf{u},\ \mathbf{x} \in \mathbb{Z}^n$ where $f : \mathbb{R}^n \to \mathbb{R}$, $A \in \mathbb{Z}^{m \times n}$, $\mathbf{b} \in \mathbb{Z}^m$, and $\mathbf{l}, \mathbf{u} \in (\mathbb{Z} \cup \{\pm\infty\})^n$. We denote $f_{\max} = \max_{\mathbf{x} \in \mathbb{Z}^n : \mathbf{l} \le \mathbf{x} \le \mathbf{u}} |f(\mathbf{x})|$. A *generalized N-fold IP matrix* is defined as

$$E^{(N)} = \begin{pmatrix} E_1^1 & E_1^2 & \cdots & E_1^N \\ E_2^1 & 0 & \cdots & 0 \\ 0 & E_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & E_2^N \end{pmatrix}.$$

Here, $r, s, t, N \in \mathbb{N}$, $E^{(N)}$ is an $(r + Ns) \times Nt$-matrix, $E_1^i \in \mathbb{Z}^{r \times t}$ and $E_2^i \in \mathbb{Z}^{s \times t}$, $i \in [N]$, are integer matrices. Problem IP with $A = E^{(N)}$ is known as *generalized N-fold integer programming* (generalized N-fold IP) [7]. The structure of $E^{(N)}$ allows us to divide any Nt-dimensional object, such as the variables of $\mathbf{x}$, bounds $\mathbf{l}, \mathbf{u}$, or the objective f, into N *bricks* of size t, e.g. $\mathbf{x} = (\mathbf{x}^1, \ldots, \mathbf{x}^N)$. We use subscripts to index within a brick and superscripts to denote the index of the brick, i.e., x_j^i is the j^{th} variable of the i^{th} brick with $j \in [t]$ and $i \in [N]$. We call a brick *integral* if all of its coordinates are integral, and *fractional* otherwise.

The *huge N-fold IP* problem is the (high-multiplicity) extension of generalized N-fold IP where there are potentially *exponentially* many bricks. The input to a huge N-fold IP problem with τ *types of bricks* is defined by matrices $E_1^i \in \mathbb{Z}^{r \times t}$ and $E_2^i \in \mathbb{Z}^{s \times t}$, $i \in [\tau]$, vectors $\mathbf{l}^1, \ldots, \mathbf{l}^\tau$, $\mathbf{u}^1, \ldots, \mathbf{u}^\tau \in \mathbb{Z}^t$, $\mathbf{b}^0 \in \mathbb{Z}^r$, $\mathbf{b}^1, \ldots, \mathbf{b}^\tau \in \mathbb{Z}^s$, functions $f^1, \ldots, f^\tau : \mathbb{R}^t \to \mathbb{R}$ satisfying $\forall i \in [\tau], \forall \mathbf{x} \in \mathbb{Z}^t : f^i(\mathbf{x}) \in \mathbb{Z}$ and given by evaluation oracles, and integers $\mu^1, \ldots, \mu^\tau \in \mathbb{N}$ such that $\sum_{i=1}^{\tau} \mu^i = N$. We say that a brick is of type i if its lower and upper bounds are $\mathbf{l}^i$ and $\mathbf{u}^i$, its right hand side is $\mathbf{b}^i$, its objective is f^i, and the matrices appearing at the corresponding coordinates are E_1^i and E_2^i. We let E be the $2 \times \tau$ block matrix $E = \begin{pmatrix} E_1^1 & E_1^2 & \cdots & E_1^\tau \\ E_2^1 & 2_2^2 & \cdots & E_2^\tau \end{pmatrix}$. We refer to Onn [25] and Knop et al. [22,23] for studies of huge N-fold IP.

3. Algorithms for ConfIP

Our goal is to prove the following theorem.

Theorem 1. *Let $\mathcal{S}$ be a ConfIP instance given in its P-representation (if the objective f is convex or concave, we assume it is presented by an evaluation oracle), and let $N = \|\mu\|_1 = \sum_{i=1}^{\tau} \mu^i$ and $\hat{L} = L + \langle \mathbf{n}, f_{\max}, N \rangle$.*

1. *ConfIP with a linear, convex, or fixed-charge objective can be solved in time $(N(d+D))^{\mathcal{O}(N(d+D))} \hat{L}^{\mathcal{O}(1)}$, and is thus fixed-parameter tractable parameterized by N and $d + D$.*
2. *ConfIP with a concave objective can be solved in time $(MN(d+D) \cdot \log \Delta)^{\mathcal{O}(N(d+D))} \hat{L}^{\mathcal{O}(1)}$, and is thus fixed-parameter tractable parameterized by N, M, $d + D$, and with Δ given in unary.*
3. *ConfIP with a linear or an extension-separable convex objective can be solved in time $(Md\Delta)^{\mathcal{O}(M^2 d + d^2 M)} \hat{L}^{\mathcal{O}(1)}$, and is thus fixed-parameter tractable parameterized by M, d, and Δ.*

4 ConfIP *with a linear or fixed-charge objective can be solved in time* $(\tau dDM \log\Delta)^{\tau^{(d+D)^{\mathcal{O}(1)}}} \hat{L}^{\mathcal{O}(1)}$*, and is thus fixed-parameter tractable parameterized by* τ*, M, d, and D if* Δ *is given in unary.*

Part 3 and Part 4 thus mean that we can solve ConfIP either in doubly-exponential time parameterized by d, D, τ, m, and M and all numbers have to be given in unary, or single-exponential time parameterized by d, m, M, and the largest coefficient Δ (but for polynomial τ). In Part 2 and Part 4 we use the fact that $(\log\alpha)^{\beta} \le 2^{\beta^2/2} + \alpha^{o(1)}$ [5, Hint 3.18] to say that having $\log\Delta$ in the base amounts to fixed-parameter algorithms when Δ is given in unary. It is worth noting that the parameter-dependence for N-fold IP w.r.t. the "usual" parameters is $(\Delta Md)^{O(d^2M+M^2d)}$. As the general runtime depends on, e.g., ellipsoid method, we give a bound of $(L+N+D)^O(1)$ that suffices to compare with the above Theorem 1. The next lemma shows how to model ConfIP as huge N-fold IP.

Lemma 2. *Let a* ConfIP *instance* $\mathcal{S}$ *be given in its P-representation. Then in time* $(\tau+D+M+L+\langle\mu,\mathbf{n}\rangle)$*, one can construct a huge N-fold IP which models* $\mathcal{S}$ *and has parameters* $r=d$*,* $s=2M$*,* $t=d+D+M$*,* $\|E\|_\infty=\Delta$*,* $N=\|\mu\|_1$*, and with* f^i *being the objective for bricks of type i.*

Proof. Let $E_1=(I\ \mathbf{0})\in\mathbb{Z}^{d\times(d+D+M)}$ where I is the $(d\times d)$-identity matrix and $\mathbf{0}$ is a $(d\times(D+M))$-all-zero matrix, let $E_1^i=E_1$ for each $i\in[\tau]$, and let $\mathbf{b}^0=\mathbf{n}$. The last M coordinates of each brick will play the role of slack variables in order to model inequalities in the system $A^i\mathbf{x}\le\mathbf{b}^i$. For each $i\in[\tau]$, obtain E_2^i from A^i by adding $M-m^i$ zero rows and $D-d^i$ zero columns, and then appending from the right the $M\times M$ identity matrix, ensuring E_2^i has M rows and $d+D+M$ columns, and append $M-m^i$ zeroes to $\mathbf{b}^i$. Formally extend the objective function f^i to $d+D+M$ dimensions by making it ignore the last $M+D-d^i$ dimensions. For each $i\in[\tau]$, define $\mathbf{l}^i=\{-\infty\}^{d+d^i}\times\{0\}^{M+D-d^i}$ and $\mathbf{u}^i=\{+\infty\}^{d+d^i}\times\{0\}^{D-d^i}\times\{+\infty\}^M$. Let μ^i be the number of bricks of type i, for each $i\in[\tau]$.

It is easy to check that, for each brick $j\in[N]$ of type $i\in[\tau]$ of the resulting huge N-fold formulation, $\mathbf{x}^j$ restricted to the first $d+d^i$ coordinates can take on exactly the values of $P^i\cap\mathbb{Z}^{d+d^i}$. Moreover, the objective value of the brick is exactly the objective value of corresponding point of $P^i\cap\mathbb{Z}^{d+d^i}$ in the ConfIP problem. Finally, by the definition of E_1, the sum of the restrictions of all bricks to the first d coordinates is exactly $\mathbf{n}$. This shows that we have reduced the ConfIP instance to a huge N-fold IP instance in a way which allows us to recover the ConfIP optimum from the huge N-fold IP optimum. Moreover, the bounds are clearly as stated in the lemma. □

The first three parts of Theorem 1 are established next via application of previous results, as we describe next: Use Lemma 2 to obtain an N-fold IP instance. Part 1 for convex or linear functions follows by applying an algorithm by Dadush and Vempala [6] for solving convex IPs, which runs in time $p^{\mathcal{O}(p)}\hat{L}^{\mathcal{O}(1)}$, where $p=N(d+D)$ is the dimension as we can delete the slack variables and use the system of inequalities instead of the N-fold IP. For a fixed-charge objective, we guess, for each $i\in[\tau]$ where $\mathbf{0}\in X^i$, a number $\bar{\mu}^i\le\mu^i$ such that an optimal solution λ has $\lambda^i_{\mathbf{0}}=\mu^i-\bar{\mu}^i$. With this guess at hand, the objective is fully determined to be $\sum_{i=1}^{\tau}\bar{\mu}^i c^i$ and it remains to verify whether there exists a corresponding decomposition of $\mathbf{n}$ by solving ConfIP with the vector $\bar{\mu}$ instead of μ and without any objective. Finally, pick the best among all guesses whose corresponding ConfIP is feasible. There are at most $N^\tau\le N^N$ guesses. To prove Part 2, we use an algorithm by Cook et al. [4] to enumerate all vertices of the corresponding polyhedron in time $(\log\Delta\cdot MN(D+d))^{\mathcal{O}(N(D+d))}\hat{L}^{\mathcal{O}(1)}$. Since a minimum of a concave function is always attained at a vertex, it suffices to evaluate the objective on each vertex and return the best as the output. Part 3 is by applying the fixed-parameter algorithm for huge N-fold IP by Knop et al. [23].

Thus, it remains to prove Part 4 of Theorem 1 that is our next goal. Our proof builds on a Structure Theorem of Goemans and Rothvoß and the idea of the proof of their main theorem [12, Theorem 2.2]. The Structure Theorem applies to the single-type setting and implies that for any solution λ corresponding to a decomposition of $\mathbf{n}$, there exists a solution $\hat{\lambda}$ whose support mostly lies within a precomputable and not-too-large set Y of "important" configurations. We first extend the Structure Theorem into the multitype setting, and then use it as follows. For each type i, we compute the set of "important" configurations Y^i, and then guess from it a small subset of configurations which will appear in the solution. Using this, we construct an ILP in small dimension, solve it, and derive from it an optimal solution λ. We take special care to enforce the multiplicity constraint (i.e., $\|\lambda^i\|_1=\mu^i$, for each $i\in[\tau]$) and argue how to encode a linear and a fixed-charge objective. Let us begin with the Structure Theorem of Goemans and Rothvoß [12]:

Proposition 3 *(Structure Theorem [12]). Let* $P=\{\mathbf{x}\mid A\mathbf{x}\le\mathbf{b}\}\subseteq\mathbb{R}^d$ *be a polytope with* $A\in\mathbb{Z}^{m\times d}$ *and* $\mathbf{b}\in\mathbb{Z}^m$ *such that all coefficients are bounded by* Δ *in absolute value. Then there exists a set* $Y\subseteq P\cap\mathbb{Z}^d$ *of size* $|Y|\le S:=m^d d^{\mathcal{O}(d)}(\log\Delta)^d$ *that can be computed in time* $S^{\mathcal{O}(1)}$ *with the following property. For every vector* $\mathbf{n}=\sum_{\mathbf{x}\in P\cap\mathbb{Z}^d}\lambda_{\mathbf{x}}\mathbf{x}$ *with* $\lambda\in\mathbb{N}^{P\cap\mathbb{Z}^d}$*, there exists a vector* $\hat{\lambda}\in\mathbb{N}^{P\cap\mathbb{Z}^d}$ *such that* $\mathbf{n}=\sum_{\mathbf{x}\in P\cap\mathbb{Z}^d}\hat{\lambda}_{\mathbf{x}}\mathbf{x}$*, satisfying the following three conditions* $\hat{\lambda}_{\mathbf{x}}\in\{0,1\}\,\forall\mathbf{x}\notin Y$*,* $|supp(\hat{\lambda})\cap Y|\le 2^{2d}$*,* $|supp(\hat{\lambda})\setminus Y|\le 2^{2d}$*.*

Next, we extend Proposition 3 to the multitype setting.

Lemma 4. *Let* $P^1,\dots,P^\tau$ *be a P-representation of* $X^1,\dots,X^\tau$*. Then, for each* $i\in[\tau]$*, there exists a set* $Y^i\subseteq P^i\cap\mathbb{Z}^{d+d^i}$ *of size* $|Y^i|\le S^i:=(m^i)^{(d+d^i)}(d+d^i)^{\mathcal{O}(d+d^i)}(\log\Delta)^{d+d^i}$ *that can be computed in time* $(S^i)^{\mathcal{O}(1)}$ *with the following property. For every vector*

$$\mathbf{n}=\sum_{i=1}^{\tau}\sum_{(\mathbf{x},\mathbf{x}')\in P^i\cap\mathbb{Z}^{d+d^i}}\lambda^i_{(\mathbf{x},\mathbf{x}')}\mathbf{x}$$

with non-negative integral λ*, there exists a non-negative integral vector* $\hat{\lambda}$ *such that* $\mathbf{n}=\sum_{i=1}^{\tau}\sum_{(\mathbf{x},\mathbf{x}')\in P^i\cap\mathbb{Z}^{d+d^i}}\hat{\lambda}^i_{(\mathbf{x},\mathbf{x}')}\mathbf{x}$*, and, for each* $i\in[\tau]$*,*

a) $\hat{\lambda}^i_{(\mathbf{x},\mathbf{x}')}\in\{0,1\}\,\forall(\mathbf{x},\mathbf{x}')\notin Y^i$*,* *b)* $|supp(\hat{\lambda}^i)\cap Y^i|\le 2^{2(d+d^i)}$*,*

c) $|supp(\hat{\lambda}^i)\setminus Y^i|\le 2^{2(d+d^i)}$*,* *d)* $\|\hat{\lambda}^i\|_1=\|\lambda^i\|_1$*.*

Proof. First, extend each P^i by a coordinate which is always 1, i.e., replace P^i by $\{(1,\mathbf{x},\mathbf{x}')\mid(\mathbf{x},\mathbf{x}')\in P^i\}$. This only increases the dimension by 1 and requires an additional equality constraint. Then, apply Proposition 3 to each P^i individually. Let $\mu^i=\|\lambda^i\|_1$ and $N=\|\lambda\|_1$. Observe that if $(N,\mathbf{n})=\sum_{i=1}^{\tau}\sum_{(1,\mathbf{x},\mathbf{x}')\in P^i\cap\mathbb{Z}^{d+d^i}}\lambda^i_{(1,\mathbf{x},\mathbf{x}')}(1,\mathbf{x})$, then there is a decomposition of $(N,\mathbf{n})$ into τ summands $(\mu^i,\mathbf{n}^i)=\sum_{(1,\mathbf{x},\mathbf{x}')\in P^i\cap\mathbb{Z}^{d+d^i}}\lambda^i_{(1,\mathbf{x},\mathbf{x}')}(1,\mathbf{x})$ to which Proposition 3 applies directly, and we obtain all points except for point d). To argue this last point, note that both decompositions $\lambda^i,\hat{\lambda}^i$ decompose $\mathbf{n}^i$ into μ^i points of $P^i\cap\mathbb{Z}^{d+d^i}$ and the claim holds. Thus, for each $i\in[\tau]$, Y^i is obtained by applying Proposition 3 to the extended polytope P^i and then projecting out the first coordinate of each element of the computed set. □

We are now ready to finish the proof of Part 4 of the theorem. First, compute the sets $Y^1, \ldots, Y^\tau$ from Lemma 4. Our goal now is to set up an ILP in small dimension, whose solution corresponds to an optimal solution λ with the properties of Lemma 4. Fix such an optimal λ. For each $i \in [\tau]$, guess a subset $Z^i \subseteq Y^i$ which satisfies $|Z^i| \le 2^{2(d+d^i)}$ and $|\mathrm{supp}(\lambda^i) \setminus Z^i| \le 2^{2(d+d^i)}$. Also guess the number $\bar{\mu}^i = |\mathrm{supp}(\lambda^i \setminus Z^i)|$. There are $\prod_{i=1}^{\tau} (S^i)^{\mathcal{O}(2^{2(d+D)})}$ choices. For each guess, we apply the following procedure, and then pick the best obtained value across all guesses and transform it into a solution of ConfIP. Introduce a variable $\lambda^i_{(\mathbf{z},\mathbf{z}')}$ for each $i \in [\tau]$ and each $(\mathbf{z},\mathbf{z}') \in Z^i$. Additionally, for each $i \in [\tau]$, introduce $\bar{\mu}^i$ vectors of variables $(\mathbf{x},\mathbf{x}')^i_j$. Then, consider the following constraints:

$$A^i(\mathbf{x},\mathbf{x}')^i_j \le \mathbf{b}^i \qquad \forall i \in [\tau],\ j \in [\bar{\mu}^i] \tag{1}$$

$$\sum_{i=1}^{\tau} \left[\sum_{(\mathbf{z},\mathbf{z}') \in Z^i} \lambda^i_{(\mathbf{z},\mathbf{z}')} \mathbf{z} + \sum_{j=1}^{\bar{\mu}^i} \mathbf{x}^i_j \right] = \mathbf{n} \tag{2}$$

$$\sum_{(\mathbf{z},\mathbf{z}') \in Z^i} \lambda^i_{(\mathbf{z},\mathbf{z}')} = \mu^i - \bar{\mu}^i \qquad \forall i \in [\tau] \text{ with } \mathbf{0} \notin X^i \tag{3}$$

$$\sum_{(\mathbf{z},\mathbf{z}') \in Z^i} \lambda^i_{(\mathbf{z},\mathbf{z}')} \le \mu^i - \bar{\mu}^i \qquad \forall i \in [\tau] \text{ with } \mathbf{0} \in X^i \tag{4}$$

$$\lambda^i_{(\mathbf{z},\mathbf{z}')} \in \mathbb{N} \qquad \forall i \in [\tau],\ \forall (\mathbf{z},\mathbf{z}') \in Z^i \tag{5}$$

$$(\mathbf{x},\mathbf{x}')^i_j \in \mathbb{Z}^{d+d^i} \qquad \forall i \in [\tau],\ \forall j \in [\bar{\mu}^i], \tag{6}$$

and, depending on the objective of the ConfIP instance, solve with one of the objectives

$$\mathrm{linear}(\lambda,\mathbf{x},\mathbf{x}') = \sum_{i=1}^{\tau} \left[\sum_{(\mathbf{z},\mathbf{z}') \in Z^i} \lambda^i_{(\mathbf{z},\mathbf{z}')} (\mathbf{w}^i \mathbf{z}) + \sum_{j=1}^{\bar{\mu}^i} \mathbf{w}^i \mathbf{x}^i_j \right], \text{ or,}$$

$$\text{fixed-charge}(\lambda,\mathbf{x},\mathbf{x}') = \sum_{i=1}^{\tau} \left[\sum_{(\mathbf{z},\mathbf{z}') \in Z^i} \lambda^i_{(\mathbf{z},\mathbf{z}')} c^i + \bar{\mu}^i c^i \right].$$

Constraints (1) and (6) ensure that the variable vectors $(\mathbf{x},\mathbf{x}')^i_j$ assume values from $P^i \cap \mathbb{Z}^{d+d^i}$, for each $i \in [\tau]$, enforcing the meaning that these variables represent the part of solution $\hat{\lambda}$ whose support does not lie in Y^i. Constraint (2) ensures that the solution indeed corresponds to a decomposition of $\mathbf{n}$ into points from $\bigcup (P^i \cap \mathbb{Z}^{d+d^i})$. The constraints (3)–(4) ensure that the number of non-zero configurations of type i is at most μ^i.

Let $(\lambda,\mathbf{x},\mathbf{x}')$ be an optimum of the ILP above, computed using an algorithm for ILP in small dimension (cf. [10,19]). We construct a solution λ^* of ConfIP as follows. For each $i \in [\tau]$ and each $\mathbf{z} \in \mathbb{Z}^d$ such that $\mathbf{z} \in \pi^i(P^i)$, let $\lambda^*(i,\mathbf{z}) = \left(\sum_{\mathbf{z}' \in \mathbb{Z}^{d^i} : (\mathbf{z},\mathbf{z}') \in P^i} \lambda(i,\mathbf{z},\mathbf{z}') + \sum_{j \in [\bar{\mu}^i], (\mathbf{x},\mathbf{x}')^i_j = (\mathbf{z},\mathbf{z}')} 1 \right)$, and let $\lambda^*(i,\mathbf{0}) = \mu^i - \sum_{\mathbf{z} \in (\pi^i(P^i) \setminus \mathbf{0}) \cap \mathbb{Z}^d} \lambda^*(i,\mathbf{z})$. We argue that λ^* is an optimal solution.

First, consider a linear objective. Observe that any solution of ConfIP induces a decomposition $\mathbf{n} = \sum_{i=1}^{\tau} \mathbf{n}^i$, and that this decomposition fully determines the objective function, which becomes $\sum_{i=1}^{\tau} \mathbf{w}^i \mathbf{n}^i$. Furthermore, Part d) of Lemma 4 guarantees that we can (almost) restrict our attention to the special sets Y^i without ruling out any decomposition of $\mathbf{n}$ into $\mathbf{n}^i$. Second, considering a fixed-charge objective, observe that the constraint (4) and our separate handling of $\lambda^*(i,\mathbf{0})$ encodes this objective appropriately.

Regarding runtime, the number of times we solve the ILP constructed above is equal to the number of guesses of the sets Z^i and the numbers $\bar{\mu}^i$, which is bounded by

$$\begin{aligned}
&\prod_{i=1}^{\tau} (S^i)^{\mathcal{O}(2^{2(d+D)})} \\
&\le \left((M)^{(d+d^{\max})} (d+D)^{\mathcal{O}(d+D)} (\log \Delta)^{(d+D)} \right)^{\tau 2^{\mathcal{O}(d+D)}} \\
&\le \left((M+d+D+\log \Delta)^{d+d_{\max}} \right)^{\tau 2^{\mathcal{O}(d+D)}} \\
&\le (M+d+D+\log \Delta)^{\tau^{(d+D)^{\mathcal{O}(1)}}} .
\end{aligned}$$

The ILP we have constructed has dimension at most $p = \sum_{i=1}^{\tau} (|Z^i| + \bar{\mu}^i(d+d^i)) \le \tau 2^{2(d+D)} + 2^{2(d+D)}(d+D) \le (\tau+d+D)2^{2(d+D)}$, and can be solved in time $p^{\mathcal{O}(p)} \hat{L}^{\mathcal{O}(1)}$ by Kannan's algorithm [19] for solving ILPs (recall that $\hat{L} = \langle N, \mathbf{n}, \mathbf{b}^1, \ldots, \mathbf{b}^\tau, \Delta, f_{\max} \rangle$). Hence, the total runtime is bounded by $(dDM \log \Delta)^{\tau^{(d+D)^{\mathcal{O}(1)}}} \hat{L}^{\mathcal{O}(1)}$. This concludes the proof of the last part of the theorem; thus, we have established Theorem 1. □

Remark. Goemans and Rothvoß prove a similar statement [12, Corollary 5.1] to Part 4 of Theorem 1, where the input $\mathbf{n}$ and the coefficients $\mathbf{w}$ have to be given in unary if one desires an FPT algorithm, whereas in our case they can be given in binary. The difference is that they invoke the Structure Theorem on a polytope P which is a disjunctive formulation of the union of polyhedra $P^1 \cup \cdots \cup P^\tau$. This disjunctive construction, however, introduces a large coefficient, increasing Δ. Similarly, a linear objective could be handled in their setting by introducing an extra variable x_{d+1} and setting $x_{d+1} = \mathbf{w}\mathbf{x}$, but this constraint would again increase Δ. We circumvent both of these limitations by using the Structure Theorem directly.

4. Hardness of ConfIP

Proposition 5. *Solving* ConfIP

1. *is* W[1]*-hard parameterized by d only, even if Δ is given in unary and no objective function;*
2. *with a fixed-charge objective is* NP*-hard even with $d = 1$ and $\Delta = 1$ (but with large penalties);*
3. *with a separable concave quadratic objective is **a)*** NP*-hard even with $d = 2$ and $\Delta = 1$, and **b)*** W[1]*-hard parameterized by d even when the largest coefficient of the objective is given in unary and $\Delta = 1$.*

Proof. Part 1. Consider the Unary Bin Packing problem, which takes as input n items of sizes $o_1, \ldots, o_n \in \mathbb{N}$ with $\max_{i \in [n]} o_i = O \le \mathrm{poly}(n)$, a capacity $B \in \mathbb{N}$, and an integer $k \in \mathbb{N}$, and we ask whether the items can be packed into k bins of capacity B. Jansen et al. [18] have shown that Unary Bin Packing is W[1]-hard parameterized by k, even for *tight* instances where $\sum_{i=1}^{n} o_i = kB$.

We shall construct a ConfIP instance with n types. We let P^i, for each $i \in [n]$, be defined by the system of inequalities $\sum_{j=1}^{k} x_j = o_i, \quad \sum_{j=1}^{k} y_j = 1, \quad x_j \le O y_j, \quad \forall j \in [k], \quad \mathbf{x} \ge \mathbf{0}$. We let $d = k$, $d^i = k$, and $\mu^i = 1$, for each $i \in [n]$, and define the $\mathbf{y}$ variables to be the one which are discarded by the projection π^i. Finally, we let $\mathbf{n}$ be a k-dimensional vector of all B. It is easy to see that $\pi^i(P^i) \cap \mathbb{Z}^k = \{(o_i, 0, \ldots, 0), (0, o^i, 0, \ldots, 0), \ldots, (0, \ldots, 0, o_i)\}$ and each element encodes the bin into which item i is assigned. Thus, the ConfIP instance is feasible if and only if there exists an assignment of items to bins such that the sum of item sizes of each bin is B.

Part 3b). We will continue working with the ConfIP instance constructed above. Recall that minimizing a concave function is equivalent to maximizing a convex function, which is the perspective we shall take here. Our goal now is to model the constraint

$x_j \leq O y_j$ which involves the big coefficient O by the objective. Let each P^i now be only defined by the constraints $\sum_{j=1}^{k} x_j = o_i$ and $\mathbf{x} \geq \mathbf{0}$, so $d = k$ and $d^i = 0$ for each $i \in [n]$. Let $f^i(\mathbf{x}) = \sum_{j=1}^{k} f_j^i(x_j)$ with $f_j^i(x_j) = (x_j - \frac{o_i}{2})^2$. This means each f_j^i is maximized exactly at the endpoints of the feasible interval $[0, o_i]$. Thus, a solution with value $\sum_{i=1}^{n} (\frac{o_i}{2})^2$ must have each $x_j \in \{0, o_i\}$ and thus corresponds to a packing; it is easy to check that no better value is attainable concluding the claim.

Part 2. In the Partition problem we are again given n numbers $o_1, \ldots, o_n$, but now their size can be large. The task is to decide if there is a subset $I \subseteq [n]$ of indices such that $\sum_{i \in I} o_i = \sum_{i \notin I} o_i$. We again construct a ConfIP instance with n types. This time, each P^i is simply a segment defined by $0 \leq x \leq o_i$, and let $\mu^i = 1$ for each $i \in [n]$. Let $a = \frac{1}{2} \sum_{i=1}^{n} o_i$. Then, for each $i \in [n]$, let $f^i(x) = o_i$ if $x \neq 0$. We claim that there is a solution of value a if and only if the Partition instance was a "yes"-instance.

In one direction, let I be a solution of the Partition instance. Then setting $\lambda(i, o_i) = 1$ if $i \in I$ and $\lambda(i, 0) = 1$ otherwise clearly defines a decomposition of a into elements o_i and has objective value a. In the other direction, assume for contradiction that there was a solution of value a but the instance was a "no" instance. Let I be the set of indices of the types $i \in [n]$ for which $\lambda(i, 0) = 0$. By our definition of the f^i's it must hold that $\sum_{i \in I} o_i = a$, which means that I certifies the instance was a "yes" instance, a contradiction.

Part 3a). We continue with the Partition problem. This time we let P^i be defined by $x_1 + x_2 = o_i$ and $x_1, x_2 \geq 0$, and let $\mu^i = 1$ for each i. Again, let $a = \frac{1}{2} \sum_{i=1}^{n} o_i$, and let $\mathbf{n} = (a, a)$. Our goal now is to use a separable convex quadratic maximization objective to enforce that either $x_1 = o_i$ or $x_2 = o_i$. We let $f_1^i(x_1) = (x_1 - \frac{o_i}{2})^2$ and similarly $f_1^i(x_2) = (x_2 - \frac{o_i}{2})^2$. It is easy to check that the only way to get $2(\frac{o_i}{2})^2$ is if either x_1 or x_2 is o_i. Thus, it is enough to check whether a solution exists with value $2\sum_{i=1}^{n} (\frac{o_i}{2})^2$. By the arguments above, this is the case if and only if the Partition instances was a "yes" instance. □

Recall Lemma 2, which states that ConfIP reduces to huge N-fold IP. So far we have used it to obtain positive results by encoding ConfIP as N-fold IP and then applying various fixed-parameter algorithms to the obtained IPs. Now it will be useful to obtain hardness of N-fold IP from Proposition 5. Specifically, applying Lemma 2 to the hardness instances of the previous proposition gives the existence of hard N-fold instances with parameters $r = M$, $s = d$, $t = d + D + M$, $\|E\|_\infty = \Delta$, and $N = \|\mu\|_1\|_1$, which implies:

Corollary 6. *Solving N-fold IPs is* W[1]*-hard parameterized by r, s, and t when $\|E\|_\infty$ is unary. Solving N-fold IPs is* NP*-hard with a fixed-charge objective even with $r = s = t = 1$ and $\Delta = 1$ (but with large penalties). Solving N-fold IPs with a separable concave quadratic objective is **a)*** NP*-hard even with $r = t = 2$ and $s = \|E\|_\infty = 1$ (but with large coefficients in the objective), and **b)*** W[1]*-hard parameterized by r and t, even when $s = \|E\|_\infty = 1$ and the largest coefficient of the objective is given in unary.*

5. Applications: packing problems and surfing

In this section we use ConfIP to model several applications; cf. Knop et al. [24] for applications in scheduling. For each application, we provide a ConfIP instance whose parameters we link with the parameters of the problem instance it encodes, and then use Theorem 1 to obtain fixed-parameter algorithms.

Packing problems. Here, we show that several variants of Bin Packing in the high-multiplicity setting can be modeled as ConfIP. In the Multiple Knapsack problem we are given positive integers B, d, d', τ, vectors $\mathbf{b}^i \in \mathbb{N}^{d'}$ of knapsack capacities and μ^i of knapsack multiplicities for each knapsack type $i \in [\tau]$, and vectors $\mathbf{s}_j$ of item sizes for $j \in [d]$. The task is to partition the $n = \sum_{j=1}^{d} n_j$ items into $B = \sum_{i=1}^{\tau} \mu^i$ bins such that sum of items (which are vectors) packed into a bin does not exceed its capacity (in any dimension). Bin Packing is the case when the dimension of an item d' is equal to 1, the number of bin types τ is 1, and we do binary search over the number n of bins necessary to pack all the items in order to find the smallest such value. Another variant of Bin Packing which is generalized by Multiple Knapsack is Bin Packing with Cardinality Constraints, where items are one-dimensional but each bin additionally has a limit on the *number* of items it can pack [21]. This is modeled as Multiple Knapsack by representing each item as a 2-dimensional vector, with the first dimension being the item's size, and the second dimension being 1, and with the second coordinate of the knapsack capacity tuple representing the limit on the number of items it can pack, hence $d = 2$, $\tau = 1$, and we find the smallest necessary number of bins using binary search. Cutting stock is the related problem where $d' = 1$, τ is the number of roll lengths, $\mu^i = n$ and with the i-th objective function being a fixed-charge objective incurring a cost $c^i \in \mathbb{N}$ for each bin of size i which is used. In the Bin Packing with General Cost Structures (GCBP) [2,8] problem we are given n items with integer sizes $s_1, \ldots, s_n$ and a monotonically non-decreasing concave function $f : [n] \to \mathbb{R}_{\geq 0}$ with $f(0) = 0$. The cost of a bin containing p items is $f(p)$. The task is to find a packing of all items into (at most) B bins each of which contains items of total size at most the common integer capacity such that the total cost is minimized. By the above discussion, it suffices to obtain fixed-parameter algorithms for the Multiple Knapsack problem and for GCBP, and thus obtain fixed-parameter algorithms to the other problems mentioned above.

Theorem 7. *Let $\sigma = \max_{i \in [d]}(\|\mathbf{s}_i\|_\infty)$.* Multiple Knapsack *is fixed-parameter tractable parameterized by $d + d' + B$, by $d + d' + \sigma$, and by $d + d' + \tau$ if σ is given in binary.*

Proof. The following constraints define the polytope P^i of possible configurations of items in a knapsacks of type $i \in [\tau]$: $\sum_{j=1}^{d} s_{j,\delta} x_j \leq b_\delta^i \quad \forall \delta \in [d'], \quad x_j \geq 0 \quad \forall j \in [d]$. There is no objective since we only have to decide whether a packing into B knapsacks of a given type exists. In summary, let $\mathcal{I}$ be an instance of Multiple Knapsack, and let $\sigma = \max_{i \in [d]}(\|\mathbf{s}_i\|_\infty)$, then the ConfIP model we have outlined has an empty objective function and the following parameters $\mathcal{S}(\Delta) = \sigma$, $\mathcal{S}(M) = d'$, $\mathcal{S}(d) = d$, $\mathcal{S}(d^i) = 0$, $\mathcal{S}(N) = B$, and $\mathcal{S}(\tau) = \tau$. Applying Parts 1, 3 and 4 of Theorem 1 proves the theorem. □

Theorem 8. *GCBP is fixed-parameter tractable when parameterized by d and all s_i are given in unary.*

Proof. Here, there is only one type of bin, hence $\tau = 1$. The polytope P^1 describing the set of configurations of items of a bin is given simply by the knapsack constraint $\sum_{j=1}^{d} s_j x_j \leq b \quad x_j \geq 0 \quad \forall j \in [d]$. The objective function is $f^1(\mathbf{x}) = f(\sum_{j=1}^{d} x_j)$ that is concave. Thus, let $\mathcal{I}$ be an instance of GCBP. There is a ConfIP model of $\mathcal{I}$ with a concave objective and parameters $\mathcal{S}(\Delta) = \sigma$, $\mathcal{S}(M) = 1$, $\mathcal{S}(d) = d$, $\mathcal{S}(d^i) = 0$, $\mathcal{S}(N) = B$, and $\mathcal{S}(\tau) = 1$. Part 2 of Theorem 1 yields the theorem. □

Surfing. In the Surfing problem we have d' *commodities* and d'' *servers* ("service providers"), with each server $j \in [d'']$ declaring a *supply vector* $\mathbf{s}_j \in \mathbb{N}^{d'}$ indicating how much of each commodity it is capable of supplying. Moreover, N is a large number of *surfers* of τ types such that there are $\mu^i \in \mathbb{N}$ surfers of type i and $N = \mu^1 + \cdots + \mu^\tau$, and for each type $i \in [\tau]$ there is a *demand vector* $\delta^i \in \mathbb{N}^{d'}$ with respect to the commodities, a *capacity vector* $\gamma^i \in \mathbb{N}^{d''}$ with respect to the servers, and a cost vector $\mathbf{c}^i \in \mathbb{N}^{d' \cdot d''}$ with respect to commodity-server pairs. The task is to determine, for each surfer, how much commodity they should buy from each server so as to satisfy the surfer's demand for each commodity, stay within capacity bounds, and minimize total cost, while also staying within each server's supply. We denote $\delta = (\delta^1, \ldots, \delta^\tau)$, $\gamma = (\gamma^1, \ldots, \gamma^\tau)$, and $\mathbf{c} = (\mathbf{c}^1, \ldots, \mathbf{c}^\tau)$.

Theorem 9. Surfing *can be solved in time* $(d^2)^{\mathcal{O}(d^2)} \langle N, \mathbf{n}, \delta, \gamma, \mathbf{c} \rangle^{\mathcal{O}(1)}$, *i.e., single-exponential in* $d = d' \cdot d''$ *and polynomial in the binary encoding of the rest of the input data.*

Proof. We model Surfing as ConfIP in the following way. We let $d = d' \cdot d''$ and we let $\mathbf{n} = (\mathbf{s}_1, \ldots, \mathbf{s}_{d''})$. Let x_{jk} be a variable describing how much commodity j a given surfer is buying from the server k. The polytope P^i describing assignments satisfying the surfer's demands and staying within bounds is given by the following constraints: $\sum_{k=1}^{d''} x_{jk} = \delta^i_j \quad \forall j \in [d'] \qquad \sum_{j=1}^{d'} x_{jk} \le \gamma^i_k \quad \forall k \in [d''] \qquad \mathbf{x} \ge \mathbf{0}$. The objective function of a surfer of type $i \in [\tau]$ is $f^i(\mathbf{x}) = \sum_{j=1}^{d'} \sum_{k=1}^{d''} c^i_{jk} x_{jk}$. It remains to deal with the fact that we do not have to use up all the available supply, so we introduce a "slack" surfer type $\tau + 1$ with only non-negativity constraints and no objective. Therefore, we conclude that there is a ConfIP model $\mathcal{S}$ for Surfing with parameters $\mathcal{S}(\Delta) = 1$, $\mathcal{S}(M) = d' + d''$, $\mathcal{S}(d) = d = d' \cdot d''$, $\mathcal{S}(d^i) = 0$, $\mathcal{S}(N) = N$, $\mathcal{S}(\tau) = \tau + 1$. Applying Part 3 of Theorem 1 then yields the theorem. □

References

[1] N. Alon, Y. Azar, G.J. Woeginger, T. Yadid, Approximation schemes for scheduling on parallel machines, J. Sched. 1 (1) (1998) 55–66.
[2] S. Anily, J. Bramel, D. Simchi-Levi, Worst-case analysis of heuristics for the bin packing problem with general cost structures, Oper. Res. 42 (2) (1994) 287–298.
[3] M. Bannach, S. Berndt, M. Maack, M. Mnich, A. Lassota, M. Rau, M. Skambath, Solving packing problems with few small items using rainbow matchings, in: Proc. MFCS 2020, in: Leibniz Int. Proc. Informatics, vol. 170, 2020, pp. 11:1–11:14.
[4] W. Cook, M. Hartmann, R. Kannan, C. McDiarmid, On integer points in polyhedra, Combinatorica 12 (1) (1992) 27–37.
[5] M. Cygan, F.V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, Parameterized Algorithms, Springer, Cham, 2015.
[6] D. Dadush, S.S. Vempala, Near-optimal deterministic algorithms for volume computation via M-ellipsoids, Proc. Natl. Acad. Sci. USA 110 (48) (2013) 19237–19245.
[7] F. Eisenbrand, C. Hunkenschröder, K. Klein, M. Koutecký, A. Levin, S. Onn, An algorithmic theory of integer programming, Technical report, 2019, http://arxiv.org/abs/1904.01361.
[8] L. Epstein, A. Levin, Bin packing with general cost structures, Math. Program. 132 (1–2, Ser. A) (2012) 355–391.
[9] W. Fernandez de la Vega, G.S. Lueker, Bin packing can be solved within $1 + \varepsilon$ in linear time, Combinatorica 1 (4) (1981) 349–355.
[10] A. Frank, É. Tardos, An application of simultaneous Diophantine approximation in combinatorial optimization, Combinatorica 7 (1) (1987) 49–65.
[11] P.C. Gilmore, R.E. Gomory, A linear programming approach to the cutting-stock problem, Oper. Res. 9 (1961) 849–859.
[12] M.X. Goemans, T. Rothvoß, Polynomiality for bin packing with a constant number of item types, in: Proc. SODA 2014, 2014, pp. 830–839.
[13] R. Hoberg, T. Rothvoss, A logarithmic additive integrality gap for bin packing, in: Proc. SODA 2017, 2017, pp. 2616–2625.
[14] D.S. Hochbaum, R. Shamir, Strongly polynomial algorithms for the high multiplicity scheduling problem, Oper. Res. 39 (4) (1991) 648–653.
[15] D.S. Hochbaum, D.B. Shmoys, Using dual approximation algorithms for scheduling problems: theoretical and practical results, J. Assoc. Comput. Mach. 34 (1) (1987) 144–162.
[16] K. Jansen, K. Klein, About the structure of the integer cone and its application to bin packing, Math. Oper. Res. 45 (4) (2020) 1498–1511.
[17] K. Jansen, R. Solis-Oba, A polynomial time $OPT + 1$ algorithm for the cutting stock problem with a constant number of object lengths, Math. Oper. Res. 36 (4) (2011) 743–753.
[18] K. Jansen, S. Kratsch, D. Marx, I. Schlotter, Bin packing with fixed number of bins revisited, J. Comput. Syst. Sci. 79 (1) (2013) 39–49.
[19] R. Kannan, Minkowski's convex body theorem and integer programming, Math. Oper. Res. 12 (3) (1987) 415–440.
[20] N. Karmarkar, R.M. Karp, An efficient approximation scheme for the one-dimensional bin-packing problem, in: Proc. FOCS 1982, 1982, pp. 312–320.
[21] H. Kellerer, U. Pferschy, Cardinality constrained bin-packing problems, Ann. Oper. Res. 92 (1999) 335–348.
[22] D. Knop, M. Koutecký, M. Mnich, Combinatorial n-fold integer programming and applications, Math. Prog. 184 (2020) 1–34.
[23] D. Knop, M. Koutecký, A. Levin, M. Mnich, S. Onn, High multiplicity n-fold IP via configuration LP, Technical report, 2021, https://research.koutecky.name/papers/huge-nfolds.pdf.
[24] D. Knop, M. Koutecký, A. Levin, M. Mnich, S. Onn, Parameterized algorithms for high-multiplicity scheduling, Technical report, 2021, https://research.koutecky.name/papers/hm-scheduling.pdf.
[25] S. Onn, Huge multiway table problems, Discrete Optim. 14 (2014) 72–77.
[26] J.M. van den Akker, J.A. Hoogeveen, S.L. van de Velde, Parallel machine scheduling by column generation, Oper. Res. 47 (6) (1999) 862–872.