



High-multiplicity N -fold IP via configuration LP

Dušan Knop¹ · Martin Koutecký² · Asaf Levin³ · Matthias Mnich⁴ · Shmuel Onn³

Received: 6 July 2021 / Accepted: 9 August 2022
© The Author(s) 2022

Abstract

N -fold integer programs (IPs) form an important class of block-structured IPs for which increasingly fast algorithms have recently been developed and successfully applied. We study *high-multiplicity* N -fold IPs, which encode IPs succinctly by presenting a description of each block *type* and a vector of block multiplicities. Our goal is to design algorithms which solve N -fold IPs in time polynomial in the size of the succinct encoding, which may be significantly smaller than the size of the explicit (non-succinct) instance. We present the first fixed-parameter algorithm for high-multiplicity N -fold IPs, which even works for convex objectives. Our key contribution is a novel proximity theorem which relates fractional and integer optima of the Configuration LP, a fundamental notion by Gilmore and Gomory [Oper. Res., 1961] which we generalize. Our algorithm for N -fold IP is faster than previous algorithms whenever the number of blocks is much larger than the number of block types, such as in N -fold IP models for various scheduling problems.

✉ Matthias Mnich
matthias.mnich@tuhh.de

Dušan Knop
dusan.knop@fit.cvut.cz

Martin Koutecký
koutecky@iuuk.mff.cuni.cz

Asaf Levin
levinas@ie.technion.ac.il

Shmuel Onn
onn@ie.technion.ac.il

- ¹ Department of Theoretical Computer Science, Faculty of Information Technology, Czech Technical University in Prague, Prague, Czech Republic
- ² Computer Science Institute, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic
- ³ Technion – Israel Institute of Technology, Haifa, Israel
- ⁴ Hamburg University of Technology, Institute for Algorithms and Complexity, Hamburg, Germany

Keywords Integer programming · Configuration IP · Fixed-parameter algorithms · Scheduling

Mathematics Subject Classification 90C10 · 90C27 · 49M27

1 Introduction

The fundamental INTEGER PROGRAMMING (IP) problem is to solve:

$$\min f(\mathbf{x}) : \mathbf{Ax} = \mathbf{b}, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \mathbf{x} \in \mathbb{Z}^n, \quad (\text{IP})$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $A \in \mathbb{Z}^{m \times n}$, $\mathbf{b} \in \mathbb{Z}^m$, and $\mathbf{l}, \mathbf{u} \in (\mathbb{Z} \cup \{\pm\infty\})^n$. Any IP instance with infinite bounds \mathbf{l}, \mathbf{u} can be reduced to an instance with finite bounds using standard techniques (solving the continuous relaxation and using proximity bounds to restrict the relevant region), so that from now on we will assume finite bounds $\mathbf{l}, \mathbf{u} \in \mathbb{Z}^n$. We denote $f_{\max} = \max_{\substack{\mathbf{x} \in \mathbb{Z}^n: \\ \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}}} |f(\mathbf{x})|$.

INTEGER PROGRAMMING is a fundamental problem with vast importance both in theory and practice. Because it is NP-hard already with a single row (by reduction from SUBSET SUM) or with A a 0/1-matrix (by reduction from VERTEX COVER), there is high interest in identifying tractable subclasses of IP. One such tractable subclass is N -fold IPs, whose constraint matrix A is defined as

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

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. We define $E := \begin{pmatrix} E_1^1 & E_1^2 & \cdots & E_1^N \\ E_2^1 & E_2^2 & \cdots & E_2^N \end{pmatrix}$, and call $E^{(N)}$ the N -fold product of E . 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. $\mathbf{x} = (\mathbf{x}^1, \dots, \mathbf{x}^N)$. 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 $i \in [N]$. Problem (IP) with $A = E^{(N)}$ is known as N -fold integer programming (N -fold IP).

Such block-structured matrices have been the subject of extensive research stretching back to the '70s [3–5, 15, 16, 28, 42, 44, 45], as this special structure allows applying methods like the Dantzig-Wolfe decomposition and others, leading to significant speed-ups in practice. On the theoretical side, the term “ N -fold IP” has been coined by De Loera et al. [9], and since then increasingly efficient algorithms have been developed and applied to various problems relating to N -fold IPs [2, 6, 25, 26, 29,

32]. This line of research culminated with an algorithm by Eisenbrand et al. [13] which solves N -fold IPs in time $(\|E\|_{\infty} r s)^{\mathcal{O}(r^2 s + r s^2)} \cdot N \log N \cdot \log \|\mathbf{u} - \mathbf{l}\|_{\infty} \cdot \log f_{\max}$ for all separable convex objectives f (i.e., when $f(\mathbf{x}) = \sum_{i=1}^n f_i(x_i)$ and each $f_i : \mathbb{R} \rightarrow \mathbb{R}$ is convex).

1.1 Our contribution

Previous algorithms for N -fold IP have focused on reducing the run-time dependency on N down to almost linear. Instead, our interest here is on N -fold IPs which model applications where many bricks are of the same *type*, that is, they share the same bounds, right-hand side, and objective function. For those applications, it is natural to encode an N -fold IP instance *succinctly* by describing each brick type by its constraint matrix, bounds, right-hand side, and objective function, and giving a vector of brick multiplicities. When the number of brick types τ is much smaller than the number N of bricks, e.g., if $N \approx 2^{\tau}$, this succinct instance is (much) smaller than the previously studied encoding of N -fold IP, and an algorithm running in time polynomial in the size of the succinct instance may be (much) faster than current algorithms. We call the N -fold IP where the instance is given succinctly the *huge N -fold IP* problem, and we present a fast algorithm for it:

Theorem 1 *Huge N -fold IP with any separable convex objective can be solved in time*

$$(\|E\|_{\infty} r s)^{\mathcal{O}(r^2 s + r s^2)} \text{poly}(\tau, t, \log \|\mathbf{l}, \mathbf{u}, \mathbf{b}, N, f_{\max}\|_{\infty}) .$$

A natural application of Theorem 1 are scheduling problems. In many scheduling problems, the number n of jobs that must be assigned to machines, as well as the number m of machines, are very large, whereas the number of types of jobs and the number of kinds of machines are relatively small. An instance of such a scheduling problem can thus be compactly encoded by simply stating, for each job type and machine kind, the number of jobs with that type and machines with that kind together with their characteristics (like processing time, weight, release time, due date, etc.), respectively. This key observation was made by several researchers [7, 37], until Hochbaum and Shamir [20] coined the term *high-multiplicity scheduling problem*. Clearly, many efficient algorithms for scheduling problems, where all jobs are assumed to be distinct, become exponential-time algorithms for the corresponding high-multiplicity problem.

Let us shortly demonstrate how Theorem 1 allows designing algorithms which are efficient for the succinct high-multiplicity encoding of the input. In modern computational clusters, it is common to have several kinds of machines differing by processing unit type (high single- or multi-core performance CPUs, GPUs), storage type (HDD, SSD, etc.), network connectivity, etc. However, the number of machine kinds τ is still much smaller (perhaps 10) than the number of machines, which may be in the order of tens of thousands or more. Many scheduling problems have N -fold IP models [31] where τ is the number of machine kinds and N is the number of machines. On these models, Theorem 1 would likely outperform the currently fastest N -fold IP algorithms. *Proof ideas.* To solve a high-multiplicity problem, one needs a succinct way to argue about solutions. In 1961, Gilmore and Gomory [17] introduced the fundamental and

widely influential notion of Configuration IP (ConfIP) which describes a solution (e.g., a schedule) by a list of pairs “(machine schedule s , multiplicity μ of machines with schedule s)”. The linear relaxation of ConfIP, called the Configuration LP (ConfLP), can often be solved efficiently, and is known to provide solutions of strikingly high quality in practice [41]; for example, the optimum of the ConfLP for BIN PACKING is conjectured to have value x such that an optimal integer packing uses $\leq \lceil x \rceil + 1$ bins [38]. However, surprisingly little is known *in general* about the structure of solutions of ConfIP and ConfLP, and how they relate to each other.

We define the Configuration IP and LP of an N -fold IP instance, and show how to solve the ConfLP quickly using the property that the ConfLP and ConfIP have polynomial encoding length even for huge N -fold IP. Our main technical contribution is a novel proximity theorem about N -fold IP, showing that a solution of its relaxation corresponding to the ConfLP optimum is very close to the integer optimum. Thus, the algorithm of Theorem 1 proceeds in three steps: **(1)** it solves the ConfLP, **(2)** it uses the proximity theorem to create a “residual” N' -fold instance with N' upperbounded by $(\|E\|_{\infty} r s)^{\mathcal{O}(rs)}$, and **(3)** it solves the residual instance by an existing N -fold IP algorithm.

1.2 Related work

Besides the references mentioned already, we point out that solving ConfLP is commonly used as subprocedure in approximation algorithms, e.g. [1, 14, 22, 27]. Jansen and Solis-Oba use a mixed ConfLP to give a parameterized $OPT + 1$ algorithm for bin packing [24]; Onn [36] gave a weaker form of Theorem 1 which only applies to the setting where $E_1^i = I$ and E_2^i is totally unimodular, for all i . Jansen et al. [25] extend the ConfIP to multiple “levels” of configurations. An extended version [31] of this paper shows how to model many scheduling problems as high multiplicity N -fold IPs, so that an application of Theorem 1 yields new parameterized algorithms for these problems. Knop and Koutecký [30] use our new proximity theorem to show efficient preprocessing algorithms (kernels) for scheduling problems.

There are currently several “fastest” algorithms for N -fold IP with standard (non-succinct) encoding. First, we have already mentioned the algorithm of Eisenbrand et al. [13]. Second, the algorithm of Jansen et al. [26] has a better parameter dependency of $(\|E\|_{\infty} r s)^{\mathcal{O}(r^2 s + s^2)}$ (as compared with $(\|E\|_{\infty} r s)^{\mathcal{O}(r^2 s + r s^2)}$ of the previous algorithm), but has a slightly worse dependence on N of $N \log^5 N$, and only works for linear objectives. Third, a recent algorithm of Cslovjecsek et al. [8] again only works for linear objectives and runs in time $(\|E\|_{\infty} s)^{\mathcal{O}(s^2)} \text{poly}(r) N \log^2(Nt) \log^2(\|\mathbf{l}, \mathbf{b}, f_{\max}\|_{\infty}) + (\|E\|_{\infty} r s)^{\mathcal{O}(r^2 s + s^2)} Nt$. While the authors claim that this constitutes the currently fastest algorithm, it seems that it is only potentially faster than prior work in a narrow parameter regime.

The third paper, by Cslovjecsek et al. [8], is the closest to ours in its approach: it solves a strong relaxation of N -fold IP which coincides with the ConfLP if each brick is of a distinct type, and which is generalized by the ConfLP (in our work) otherwise. The authors show that this relaxation can be solved in near-linear time, and then develop a proximity theorem similar to ours (but using different techniques)

and a dynamic program, which allows them to construct and solve a residual instance in linear time. An earlier version of our paper [31] stated a worse proximity bound than that of Cslovjecssek et al. [8], but our bound applies to separable convex objective whereas theirs [8] does not. Presently, we adapt one of their lemmas ([8, Lemma 3]) (Lemma 5) and a modeling idea (Sect. 3.4) to obtain the same proximity bound as they have [8], but which also works for separable convex objectives. It is likely that the complexity of our algorithm to solve the ConfLP could be improved along the lines of their work [8]. Despite these similarities, we highlight that only our algorithm solves the high-multiplicity version of N -fold IP.

2 Preliminaries

For positive integers m, n with $m \leq n$ we set $[m, n] = \{m, m + 1, \dots, n\}$ 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 , $\lceil \alpha \rceil$ is the ceiling of α , and we define $\{\alpha\} = \alpha - \lfloor \alpha \rfloor$, similarly for vectors where these operators are defined component-wise.

We call a brick of \mathbf{x} *integral* if all of its coordinates are integral, and *fractional* otherwise.

Huge N -fold IP. The *huge N -fold IP* problem is an extension of N -fold IP to the high-multiplicity scenario, where there are potentially *exponentially* many bricks. This requires a succinct representation of the input and output. The input to a huge N -fold IP problem with τ brick types 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 $\mathbf{l}^1, \dots, \mathbf{l}^\tau, \mathbf{u}^1, \dots, \mathbf{u}^\tau \in \mathbb{Z}^t, \mathbf{b}^0 \in \mathbb{Z}^r, \mathbf{b}^1, \dots, \mathbf{b}^\tau \in \mathbb{Z}^s$, functions $f^1, \dots, f^\tau: \mathbb{R}^t \rightarrow \mathbb{R}$ 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 $\mu^1, \dots, \mu^\tau \in \mathbb{N}$ 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 . The task is to solve (IP) with a matrix $E^{(N)}$ which has μ^i bricks of type i for each i . Onn [35] shows that for any solution, there exists a solution which is at least as good and has only few (at most $\tau \cdot 2^t$) distinct bricks. In Sect. 3 we show new bounds which do not depend exponentially on t .

2.1 Graver bases and the Steinitz lemma

Let \mathbf{x}, \mathbf{y} be n -dimensional vectors. We call \mathbf{x}, \mathbf{y} *sign-compatible* if they lie in the same orthant, that is, for each $i \in [n], x_i \cdot y_i \geq 0$. We call $\sum_i \mathbf{g}^i$ a *sign-compatible sum* if all \mathbf{g}^i are pair-wise sign-compatible. Moreover, we write $\mathbf{y} \sqsubseteq \mathbf{x}$ if \mathbf{x} and \mathbf{y} are sign-compatible and $|y_i| \leq |x_i|$ for each $i \in [n]$. Clearly, \sqsubseteq imposes a partial order, called “conformal order”, on n -dimensional vectors. For an integer matrix $A \in \mathbb{Z}^{m \times n}$, its *Graver basis* $\mathcal{G}(A)$ is the set of \sqsubseteq -minimal non-zero elements of the *lattice* of A , $\ker_{\mathbb{Z}}(A) = \{\mathbf{z} \in \mathbb{Z}^n \mid A\mathbf{z} = \mathbf{0}\}$. A *circuit* of A is an element $\mathbf{g} \in \ker_{\mathbb{Z}}(A)$ whose support $\text{supp}(\mathbf{g})$ (i.e., the set of its non-zero entries) is minimal under inclusion and whose entries are coprime. We denote the set of *circuits* of A by $\mathcal{C}(A)$. It is known

that $\mathcal{C}(A) \subseteq \mathcal{G}(A)$ [34, Definition 3.1 and remarks]. We make use of the following two propositions:

Proposition 1 (Positive Sum Property [34, Lemma 3.4]) *Let $A \in \mathbb{Z}^{m \times n}$ be an integer matrix. For any integer vector $\mathbf{x} \in \ker_{\mathbb{Z}}(A)$, there exists an $n' \leq 2n - 2$ and a decomposition $\mathbf{x} = \sum_{j=1}^{n'} \alpha_j \mathbf{g}_j$ with $\alpha_j \in \mathbb{N}$ for each $j \in [n']$, into a sum of $\mathbf{g}_j \in \mathcal{G}(A)$. For any fractional vector $\mathbf{x} \in \ker(A)$ (that is, $A\mathbf{x} = \mathbf{0}$), there exists a decomposition $\mathbf{x} = \sum_{j=1}^n \alpha_j \mathbf{g}_j$ into $\mathbf{g}_j \in \mathcal{C}(A)$, where $\alpha_j \geq 0$ for each $j \in [n]$.*

Proposition 2 (Separable convex superadditivity [10, Lemma 3.3.1]) *Let $f(\mathbf{x}) = \sum_{i=1}^n f_i(x_i)$ be separable convex, let $\mathbf{x} \in \mathbb{R}^n$, and let $\mathbf{g}_1, \dots, \mathbf{g}_k \in \mathbb{R}^n$ be vectors with the same sign-pattern from $\{\leq 0, \geq 0\}^n$, that is, belonging to the same orthant of \mathbb{R}^n . Then*

$$f\left(\mathbf{x} + \sum_{j=1}^k \alpha_j \mathbf{g}_j\right) - f(\mathbf{x}) \geq \sum_{j=1}^k \alpha_j (f(\mathbf{x} + \mathbf{g}_j) - f(\mathbf{x})) \quad (2)$$

for arbitrary integers $\alpha_1, \dots, \alpha_k \in \mathbb{N}$.

Our proximity theorem relies on the Steinitz Lemma, which has recently received renewed attention [11, 12, 23].

Lemma 1 (Steinitz [40], Sevastjanov, Banaszczyk [39]) *Let $\|\cdot\|$ denote any norm, and let $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ be such that $\|\mathbf{x}_i\| \leq 1$ for $i \in [n]$ and $\sum_{i=1}^n \mathbf{x}_i = \mathbf{0}$. Then there exists a permutation $\pi \in S_n$ such that for all $k = 1, \dots, n$, the prefix sum satisfies $\left\|\sum_{i=1}^k \mathbf{x}_{\pi(i)}\right\| \leq d$.*

For an integer matrix A , we define $g_1(A) = \max_{\mathbf{g} \in \mathcal{G}(A)} \|\mathbf{g}\|_1$. When it could make a difference, we will state our bounds both in terms of $\|E\|_{\infty}$ (worst-case, when we have no other information) and in terms of $g_1(E_2) := \max_i g_1(E_2^i)$, e.g. in Lemma 10 and Theorem 2.

3 Proof of Theorem 1

We first give a relatively high-level description of the proof, before we present all its details.

3.1 Proof overview and ideas

3.1.1 Configuration LP and IP

Given an input to the huge N -fold IP, we first reformulate it as another IP, which we refer to as the Configuration IP. We then consider its fractional relaxation, the so-called Configuration LP. Our approach is to (efficiently) solve the Configuration LP, and bound the distance of its LP optimum to the integer optimum (of the Configuration

IP). We use this bound to reduce the input to the huge N -fold IP from a high-multiplicity input to an input of a standard N -fold IP which is small both in terms of the number of bricks and size of the bounding box. This small input we then solve using an existing N -fold IP algorithm. On this way, there are several non-trivial obstacles that we need to overcome.

We will refer to huge N -fold IP as HugeIP, its corresponding fractional relaxation as HugeCP (this is a convex program if the objective f is convex), the Configuration LP of the HugeIP as ConfLP, and to its integer version as ConfIP. We define a mapping φ from the solutions of ConfLP to the solutions of HugeCP which, for every variable y_c of the ConfLP introduces $\lfloor y_c \rfloor$ bricks with configuration \mathbf{c} , and then introduces $\sum_c \{y_c\}$ bricks with configuration $\frac{1}{\sum_c \{y_c\}} \sum_c \{y_c\} \cdot \mathbf{c}$ (i.e., an “average” configuration). We call a solution \mathbf{x}^* of HugeCP “conf-optimal” if it is the image $\varphi(\mathbf{y}^*)$ of some ConfLP optimum \mathbf{y}^* . One would hope that then the objective value of a conf-optimal solution \mathbf{x}^* in HugeCP and of \mathbf{y}^* in ConfLP were identical. While this is true for any linear objective f , it need not be true for a convex objective f . To overcome this impediment, we introduce an auxiliary objective \hat{f} which preserves the values of optima of ConfLP and conf-optimal solutions of HugeCP.

3.1.2 Proximity theorem

The bulk of our work is showing that for each conf-optimal solution \mathbf{x}^* of the HugeLP, there is an optimum \mathbf{z}^* of the HugeIP whose ℓ_1 -distance from \mathbf{x}^* is bounded by $P := (\|E\|_{\infty} r s)^{\mathcal{O}(rs)}$. We will show that we can obtain a ConfLP optimum \mathbf{y} with support of size at most $r + \tau$, and by the definition of φ (recall that $\mathbf{x}^* = \varphi(\mathbf{y})$), this means that \mathbf{x}^* has at most $r + \tau + 1$ distinct bricks (the $+1$ is due to φ creating an additional “average configuration” brick type). This, in turn, means that our bound on the ℓ_1 -distance between \mathbf{z}^* and \mathbf{x}^* says something about ConfLP and ConfIP: for any ConfLP optimum \mathbf{y} there is a ConfIP optimum \mathbf{y}^* in ℓ_1 -distance at most P where any configuration \mathbf{c} in the support of \mathbf{y}^* is at most P far from some configuration \mathbf{c}' in the support of \mathbf{y} . As far as we know, this is a unique result about the Configuration LP.

A way of bounding the distance between some types of optima in an integer program has been introduced by Hochbaum and Shanthikumar [21] and adapted to the setting of N -fold IP by Hemmecke et al. [19]. A somewhat different approach was later developed by Eisenbrand and Weismantel [11] in the setting of IPs with few rows, and was adapted to the setting of N -fold IPs soon after [12, 13]. The idea is as follows. Let \mathbf{x}^* be a HugeCP optimum, and \mathbf{z}^* be a HugeIP optimum, We call a non-zero integral vector $\mathbf{p} \sqsubseteq \mathbf{x}^* - \mathbf{z}^*$, i.e., which is sign-compatible (i.e., has the same sign-pattern) with $\mathbf{x}^* - \mathbf{z}^*$ and which is smaller in absolute value than $\mathbf{x}^* - \mathbf{z}^*$ in each coordinate, a cycle of $\mathbf{x}^* - \mathbf{z}^*$. If \mathbf{z}^* minimizes $\|\mathbf{x}^* - \mathbf{z}^*\|_1$, it can be shown that no cycle of $\mathbf{x}^* - \mathbf{z}^*$ exists. Moreover, if a cycle exists, then a cycle of ℓ_1 -norm at most B exists, which implies $\|\mathbf{x}^* - \mathbf{z}^*\|_1 \leq B$.

Notice that the previous argument assumes \mathbf{x}^* to be a HugeCP optimum: this cannot be replaced with a conf-optimal solution for the following reason. The existence of a cycle \mathbf{p} leads to a contradiction because either $\mathbf{z}^* + \mathbf{p}$ is also a HugeIP optimum (but closer to \mathbf{x}^*) or $\mathbf{x}^* - \mathbf{p}$ is also a HugeCP optimum (but closer to \mathbf{z}^*). But if \mathbf{x}^* is a conf-

optimal solution, we have no guarantee that $\mathbf{x}^* - \mathbf{p}$ is again a configurable solution, and the argument breaks down. This means that we need to restrict our attention to cycles with the property that if \mathbf{x}^* is a configurable solution, then $\mathbf{x}^* - \mathbf{p}$ is also configurable.

We call such a \mathbf{p} a *configurable cycle*. The next task is an analogy of the argument above: if \mathbf{x}^* is conf-optimal and \mathbf{z}^* is a HugeIP optimum, then the existence of a configurable cycle \mathbf{p} of $\mathbf{x}^* - \mathbf{z}^*$ leads to a contradiction. For that, we need the separability and convexity of the objective f and a careful use of the configurability of \mathbf{p} . With this argument at hand, we have reduced our task to bounding the norm of any configurable cycle (Lemma 7).

However, the main existing tool for showing proximity is by ruling out cycles. To overcome this, we develop new tools to deal with configurable cycles.

3.1.3 The algorithm

It remains to use our proximity bound P . As already hinted at, if two solutions differ in ℓ_1 -norm by at most P , then they may differ in at most P bricks. This means that we may fix all but P bricks for each configuration appearing in the ConfLP optimum. Since the size of the support of the ConfLP optimum is small ($r + \tau$), the total number of bricks to be determined is also small, and can be done using a standard N -fold IP algorithm in the required time complexity (Proof of Theorem 1)

To recap, the algorithm works in the following steps.

1. We solve the ConfLP and obtain its optimum \mathbf{y} by solving its Dual LP using a separation oracle. The separation oracle is implemented using a fixed-parameter algorithm for IP with small coefficients.
2. We use the ConfLP optimum \mathbf{y} to fix the solution on all but $(r + \tau)P$ bricks.
3. The remaining instance can be encoded as an N -fold IP with at most $(r + \tau)P$ bricks and solved using an existing algorithm.

Let us now go back to a detailed proof of Theorem 1.

3.2 Configurations of huge N -fold IP

Fix a huge N -fold IP instance with τ types. Recall that μ^i denotes the number of bricks of type i , and $\boldsymbol{\mu} = (\mu^1, \dots, \mu^\tau)$. We define for each $i \in [\tau]$ the set of configurations of type i as

$$C^i = \left\{ \mathbf{c} \in \mathbb{Z}^t \mid E_2^i \mathbf{c} = \mathbf{b}^i, \mathbf{l}^i \leq \mathbf{c} \leq \mathbf{u}^i \right\} .$$

Here we are interested in four instances of convex programming (CP) and convex integer programming (IP) related to huge N -fold IP. First, we have the *Huge IP*

$$\min f(\mathbf{x}) : E^{(N)} \mathbf{x} = \mathbf{b}, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \mathbf{x} \in \mathbb{Z}^{Nt}, \quad (\text{HugeIP})$$

and the *Huge CP*, which is a relaxation of (HugeIP),

$$\min \hat{f}(\mathbf{x}) : E^{(N)} \mathbf{x} = \mathbf{b}, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \mathbf{x} \in \mathbb{R}^{Nt} . \quad (\text{HugeCP})$$

We shall define the objective function \hat{f} later, for now it suffices to say that for all integral feasible $\mathbf{x} \in \mathbb{Z}^{Nt}$ we have $f(\mathbf{x}) = \hat{f}(\mathbf{x})$ so that indeed the optimum of (HugeCP) lower bounds the optimum of (HugeIP) and that \hat{f} is convex. Then, there is the *Configuration LP* of (HugeIP), that is, the following linear program:

$$\min \mathbf{v}\mathbf{y} = \min \sum_{i=1}^{\tau} \sum_{\mathbf{c} \in \mathcal{C}^i} f^i(\mathbf{c}) \cdot y(i, \mathbf{c}) \tag{3}$$

$$\begin{aligned} \sum_{i=1}^{\tau} E_1^i \sum_{\mathbf{c} \in \mathcal{C}^i} \mathbf{c}y(i, \mathbf{c}) &= \mathbf{b}^0, \\ \sum_{\mathbf{c} \in \mathcal{C}^i} y(i, \mathbf{c}) &= \mu^i \quad \forall i \in [\tau], \\ \mathbf{y} &\geq \mathbf{0}. \end{aligned} \tag{4}$$

Letting B be its constraint matrix and $\mathbf{d} = \begin{pmatrix} \mathbf{b}^0 \\ \boldsymbol{\mu} \end{pmatrix}$ be the right hand side, we can shorten (3)–(4) as

$$\min \mathbf{v}\mathbf{y} : B\mathbf{y} = \mathbf{d}, \mathbf{y} \geq \mathbf{0}. \tag{ConfLP}$$

Finally, by observing that $B\mathbf{y} = \mathbf{d}$ implies $y(i, \mathbf{c}) \leq \|\boldsymbol{\mu}\|_{\infty}$ for all $i \in [\tau], \mathbf{c} \in \mathcal{C}^i$, defining $C = \sum_{i \in [\tau]} |\mathcal{C}^i|$, leads to the *Configuration ILP*,

$$\min \mathbf{v}\mathbf{y} : B\mathbf{y} = \mathbf{d}, \mathbf{0} \leq \mathbf{y} \leq (\|\boldsymbol{\mu}\|_{\infty}, \dots, \|\boldsymbol{\mu}\|_{\infty})^T, \mathbf{y} \in \mathbb{N}^C. \tag{ConfILP}$$

A solution \mathbf{x} of (HugeCP) is *configurable* if, for every $i \in [\tau]$, each brick \mathbf{x}^j of type i is a convex combination of \mathcal{C}^i , i.e., $\mathbf{x}^j \in \text{conv}(\mathcal{C}^i)$. We shall define a mapping from solutions of (ConfLP) to configurable solutions of (HugeCP) as follows. For every solution \mathbf{y} of (ConfLP) we define a solution $\mathbf{x} = \varphi(\mathbf{y})$ of (HugeCP) to have $\lfloor y(i, \mathbf{c}) \rfloor$ bricks of type i with configuration \mathbf{c} and, for each $i \in [\tau]$, let $f^i = \sum_{\mathbf{c} \in \mathcal{C}^i} \{y(i, \mathbf{c})\}$ and let \mathbf{x} have f^i bricks with value $\hat{\mathbf{c}}_i = \frac{1}{f^i} \sum_{\mathbf{c} \in \mathcal{C}^i} \{y(i, \mathbf{c})\}\mathbf{c}$. (Because $\sum_{\mathbf{c} \in \mathcal{C}^i} y(i, \mathbf{c}) = \mu^i$ and $\sum_{\mathbf{c} \in \mathcal{C}^i} \lfloor y(i, \mathbf{c}) \rfloor$ is clearly integral, $f^i = \mu^i - \sum_{\mathbf{c} \in \mathcal{C}^i} \{y(i, \mathbf{c})\}$ is also integral.) Note that $\varphi(\mathbf{y})$ has at most as many fractional bricks as \mathbf{y} has fractional entries since each $f^i < 1$ and the number of non-zero f^i is at most the number of fractional entries of \mathbf{y} . Call a solution \mathbf{x} of (HugeCP) *conf-optimal* if there is an optimal solution \mathbf{y} of (ConfLP) such that $\mathbf{x} = \varphi(\mathbf{y})$.

We are going to introduce an auxiliary objective function \hat{f} , but we first want to discuss our motivation in doing so. The reader might already see that for any integer solution $\mathbf{y} \in \mathbb{Z}^C$ of (ConfILP), $\mathbf{v}\mathbf{y} = f(\varphi(\mathbf{y}))$ holds, as we shall prove in Lemma 4. Our natural hope would be that for a fractional optimum \mathbf{y}^* of (ConfLP) we would have $\mathbf{v}\mathbf{y}^* = f(\varphi(\mathbf{y}^*))$. However, by convexity of f and the construction of $\hat{\mathbf{c}}_i$ it only follows that $\mathbf{v}\mathbf{y}^* \geq f(\varphi(\mathbf{y}^*))$. Even worse, there may be two conf-optimal solutions \mathbf{x} and \mathbf{x}' with $f(\mathbf{x}) < f(\mathbf{x}')$. To overcome this, we define an auxiliary objective function \hat{f} with the property that for any conf-optimal solution \mathbf{x}^* of (HugeCP) and any optimal solution \mathbf{y}^* of (ConfLP), $\mathbf{v}\mathbf{y}^* = \hat{f}(\mathbf{x}^*)$.

Fix a brick \mathbf{x}^j of type i . We say that a multiset $\Gamma^j \subseteq (\mathcal{C}^i \times \mathbb{R}_{\geq 0})$ is a *decomposition* of \mathbf{x}^j and write $\mathbf{x}^j = \sum \Gamma^j$ if $\mathbf{x}^j = \sum_{(\mathbf{c}, \lambda_{\mathbf{c}}) \in \Gamma^j} \lambda_{\mathbf{c}} \mathbf{c}$ and $\sum_{(\mathbf{c}, \lambda_{\mathbf{c}}) \in \Gamma^j} \lambda_{\mathbf{c}} = 1$. We define the objective $\hat{f}(\mathbf{x})$ for all configurable solutions as $\hat{f}(\mathbf{x}) = \sum_{j=1}^N \hat{f}^i(\mathbf{x}^j)$, where

$$\hat{f}^i(\mathbf{x}^j) = \min_{\Gamma^j: \sum \Gamma^j = \mathbf{x}^j} \sum_{(\mathbf{c}, \lambda_{\mathbf{c}}) \in \Gamma^j} \lambda_{\mathbf{c}} \cdot f^i(\mathbf{c}) . \tag{5}$$

In a sense, $\hat{f}(\mathbf{x})$ is the value of the minimum (w.r.t. f) interpretation of \mathbf{x} as a convex combination of feasible integer solutions. Correspondingly, we call a decomposition Γ^j of \mathbf{x}^j *\hat{f} -optimal* if it is a minimizer of (5). Formally, we let $\hat{f}^i(\mathbf{x}^j) = f^i(\mathbf{x}^j)$ for a non-configurable \mathbf{x}^j in order to make the definition of (HugeCP) valid; however, we are never interested in the value of \hat{f} for non-configurable bricks in the following.

Lemma 2 *Let \mathbf{x} be a configurable solution of (HugeCP), and \mathbf{x}^j be a brick of type i . Then $f^i(\mathbf{x}^j) \leq \hat{f}^i(\mathbf{x}^j)$. If \mathbf{x}^j is integral, then $f^i(\mathbf{x}^j) = \hat{f}^i(\mathbf{x}^j)$.*

Proof By convexity of f^i we have

$$f^i(\mathbf{x}^j) = f^i \left(\sum_{(\mathbf{c}, \lambda_{\mathbf{c}}) \in \Gamma^j} \lambda_{\mathbf{c}} \mathbf{c} \right) \leq \sum_{(\mathbf{c}, \lambda_{\mathbf{c}}) \in \Gamma^j} \lambda_{\mathbf{c}} f^i(\mathbf{c}),$$

for any decomposition Γ^j of \mathbf{x}^j . If \mathbf{x}^j is integral, then $\Gamma^j = \{(\mathbf{x}^j, 1)\}$ is its optimal decomposition (not necessarily unique¹), concluding the proof. \square

Moreover, for each \mathbf{x}^j there is an \hat{f} -optimal decomposition Γ^j with $|\Gamma^j| \leq t + 1$ since \hat{f} -optimal decompositions correspond to optima of a linear program with $t + 1$ equality constraints, namely

$$\min \sum_{\mathbf{c} \in \mathcal{C}^i} \lambda_{\mathbf{c}} f^i(\mathbf{c}) \quad \text{s.t.} \quad \sum_{\mathbf{c} \in \mathcal{C}^i} \lambda_{\mathbf{c}} \mathbf{c} = \mathbf{x}^j, \quad \|\boldsymbol{\lambda}\|_1 = 1, \quad \boldsymbol{\lambda} \geq \mathbf{0} . \tag{6}$$

Let us describe the relationship of the objective values of the various formulations.

Lemma 3 *For any feasible solution $\tilde{\mathbf{y}}$ of (ConfLP),*

$$\mathbf{v}\tilde{\mathbf{y}} \geq \hat{f}(\varphi(\tilde{\mathbf{y}})) . \tag{7}$$

Proof Let $\tilde{\mathbf{x}} = \varphi(\tilde{\mathbf{y}})$. We can decompose $\hat{f}(\varphi(\tilde{\mathbf{y}})) = U_1 + U_2$, where U_1 is the cost of integer bricks of $\varphi(\tilde{\mathbf{y}})$ and U_2 is the cost of its fractional bricks. It is easy to see that $U_1 = \mathbf{v}\lfloor \tilde{\mathbf{y}} \rfloor$ by the equality of f^i and \hat{f}^i , for all $i \in [\tau]$, over integer vectors. We shall further decompose the value U_2 into costs of fractional bricks of each type. For each $i \in [\tau]$, the cost of each fractional brick of type i is at most $\frac{1}{\beta^i} \sum_{\mathbf{c} \in \mathcal{C}^i} \{\tilde{y}(i, \mathbf{c})\} f^i(\mathbf{c})$ because

¹ e.g., potentially $\mathbf{x}^j = \frac{1}{2}((\mathbf{x}^j + \mathbf{c}) + (\mathbf{x}^j - \mathbf{c}))$ for some \mathbf{c} , and Γ^j is optimal for any linear objective.

the decomposition $\left\{ \left(\mathbf{c}, \frac{1}{f^i} \{\tilde{y}^i\} \right) \mid \mathbf{c} \in \mathcal{C}^i \right\}$ of $\hat{\mathbf{c}}_i$ (recall that $\hat{\mathbf{c}}_i = \frac{1}{f^i} \sum_{\mathbf{c} \in \mathcal{C}^i} \{\tilde{y}^i(i, \mathbf{c})\} \mathbf{c}$) is merely a feasible (not necessarily optimal) solution of (6) Summing this estimate up over all f^i fractional bricks of type i gives $f^i \cdot \frac{1}{f^i} \sum_{\mathbf{c} \in \mathcal{C}^i} \{\tilde{y}^i(i, \mathbf{c})\} f^i(\mathbf{c}) = \mathbf{v}^i \{\tilde{y}^i\}$, concluding the proof. \square

Lemma 4 *Let $\hat{\mathbf{y}}$ be an optimum of (ConfILP), \mathbf{z}^* be an optimum of (HugeIP), \mathbf{y}^* be an optimum of (ConfLP), $\tilde{\mathbf{x}} = \varphi(\mathbf{y}^*)$, and \mathbf{x}^* be a configurable optimum of (HugeCP). Then*

$$\hat{f}(\mathbf{z}^*) = f(\mathbf{z}^*) = f(\varphi(\hat{\mathbf{y}})) = \mathbf{v}\hat{\mathbf{y}} \geq \mathbf{v}\mathbf{y}^* = \hat{f}(\tilde{\mathbf{x}}) = \hat{f}(\mathbf{x}^*) .$$

Proof We have $\hat{f}(\mathbf{z}^*) = f(\mathbf{z}^*)$ by equality of \hat{f} and f on integer solutions (Lemma 2), and $f(\mathbf{z}^*) = f(\varphi(\hat{\mathbf{y}})) = \mathbf{v}\hat{\mathbf{y}}$ by the definition of φ and the fact that $\hat{\mathbf{y}}$ is an integer optimum. Clearly, $\mathbf{v}\hat{\mathbf{y}} \geq \mathbf{v}\mathbf{y}^*$, because (ConfLP) is a relaxation of (ConfILP) and thus the former lower bounds the latter.

Let us construct a mapping ϕ for any configurable solution \mathbf{x} of (HugeCP). Start with $\phi(\mathbf{x}) = \mathbf{y} = \mathbf{0}$. For each brick \mathbf{x}^j of type i let Γ^j be a \hat{f} -optimal decomposition of \mathbf{x}^j and update $y_c^i := y_c^i + \lambda_c$ for each $(\mathbf{c}, \lambda_c) \in \Gamma^j$. Now it is easy to see that

$$\mathbf{v}\phi(\mathbf{x}^*) = \hat{f}(\mathbf{x}^*) . \tag{8}$$

Our goal is to argue that $\mathbf{v}\mathbf{y}^* = \hat{f}(\tilde{\mathbf{x}}) = \hat{f}(\mathbf{x}^*)$. We have $\hat{f}(\tilde{\mathbf{x}}) = \hat{f}(\varphi(\mathbf{y}^*)) \leq \mathbf{v}\mathbf{y}^*$ by (7), but by optimality of \mathbf{y}^* and (8) it must be that $\mathbf{v}\phi(\tilde{\mathbf{x}}) = \hat{f}(\tilde{\mathbf{x}}) \geq \mathbf{v}\mathbf{y}^*$ and hence $\mathbf{v}\mathbf{y}^* = \hat{f}(\tilde{\mathbf{x}})$. Similarly,

$$\hat{f}(\mathbf{x}^*) = \mathbf{v}\phi(\mathbf{x}^*) \geq \mathbf{v}\mathbf{y}^* \geq \hat{f}(\varphi(\mathbf{y}^*))$$

with the “=” by (8), the first “ \geq ” by optimality of \mathbf{y}^* , and the second “ \geq ” by (7). However, since $\hat{f}(\varphi(\mathbf{y}^*)) \geq \hat{f}(\mathbf{x}^*)$ by optimality of \mathbf{x}^* , all inequalities are in fact equalities and thus $\mathbf{v}\mathbf{y}^* = \hat{f}(\mathbf{x}^*)$. \square

Remark 1 We only need the properties of \hat{f} that we have proved so far. To gain a little bit more intuition, consider the dual of the LP (6). Notice that the set of right hand sides \mathbf{x}^j whose optimum is attained by a particular set of configurations $\text{supp}(\lambda)$ is a polyhedron. Call such a set a cell. This means that \hat{f} is a convex function which is linear in each cell. Another observation is that \hat{f} is non-separable.

We do not have a more intuitive explanation of \hat{f} . It would be tempting to think that \hat{f} is the piece-wise linear approximation of f in which, for every $i \in [Nt]$, we replace each segment of f_i between two adjacent integers $k, k + 1$ by the affine function going through the points $(k, f_i(k))$ and $(k + 1, f_i(k + 1))$. However, this turns out to be incorrect: for example, say that $f_1(x_1) = |x_1 - 1|$ (thus $f_1(0) = f_1(2) = 1$ and $f_1(1) = 0$) and that we set $x_1 = 2x_2$ for a new integer variable x_2 . This constraint ensures that x_1 only takes on even values. Thus, x_1 never attains the value 1 and $\hat{f}_1(1) \geq 1$ even though the piece-wise linear approximation of f_1 has value 0 at 1.

Bounding the number of fractional coordinates.

Lemma 5 (Adaptation of [8, Lemma 4.1]) *An optimal vertex solution \mathbf{y}^* of (ConfLP) has at most $2r$ fractional coordinates.*

Proof Notice that if a brick $(\mathbf{y}^*)^i$ is a vertex of the set $Q^i := \text{conv}\{\mathbf{y}^i \in \mathbb{R}^{C^i} \mid \mathbf{1}\mathbf{y}^i = \mu^i, \mathbf{y}^i \geq \mathbf{0}\}$, then it is integral. Thus, any brick of \mathbf{y}^* which is fractional cannot be a vertex of Q^i and hence there exists a direction $\mathbf{e}^i \in \text{Ker}_{\mathbb{Z}}(\mathbf{1})$ and a length $\lambda^i > 0$ such that $(\mathbf{y}^*)^i \pm \lambda^i \mathbf{e}^i \in Q^i$. For the sake of contradiction, assume there are $r + 1$ bricks of \mathbf{y}^* which contain a fractional coordinate and I is the index set of such bricks. Hence we have \mathbf{e}^i, λ^i as above for each $i \in I$. We abuse the notation and treat C^i as a matrix whose columns are the configurations. Consider the vectors $E_1^i C^i \lambda^i \mathbf{e}^i \in \mathbb{R}^r$: because there are $r + 1$ of them, they are linearly dependent, and, by rescaling, there must be coefficients $\tilde{\lambda}^i$ such that $|\tilde{\lambda}^i| \leq \lambda^i$ for each $i \in I$ and $\sum_{i \in I} E_1^i C^i \tilde{\lambda}^i \mathbf{e}^i = \mathbf{0}$. Define $\mathbf{e} \in \mathbb{R}^C$ (recall that C is the total number of configurations) such that its i -th brick is equal to $\tilde{\lambda}^i \mathbf{e}^i$ if $i \in I$, and is $\mathbf{0}$ otherwise. Then $\mathbf{y}^* \pm \mathbf{e}$ are both feasible solutions of (ConfLP), and thus \mathbf{y}^* is not a vertex solution—a contradiction.

So far, we have shown there are at most r fractional bricks of \mathbf{y}^* . Notice that all we needed for that was $r + 1$ linearly dependent vectors which can be added to some brick in both directions while preserving feasibility. Because $\mathbf{e}^i \in \text{Ker}_{\mathbb{Z}}(\mathbf{1})$ for each $i \in I$, we can decompose \mathbf{e}^i into elements of $\mathcal{G}(\mathbf{1})$, which are exactly vectors with one 1 and one -1 . Hence, to avoid the contradiction above, there can be at most r vectors \mathbf{e}^i , and, additionally, all of them must belong to $\mathcal{G}(\mathbf{1})$. Thus, the resulting vector \mathbf{e} has support of size at most $2r$, and \mathbf{y}^* has at most $2r$ fractional coordinates. \square

Finding a conf-optimal solution with small number of fractional bricks.

Our goal is to show that the proximity of any conf-optimal solution \mathbf{x}^* of (HugeCP) from an integer optimum \mathbf{z}^* of (HugeIP) depends on the number of fractional bricks. This number, by definition of φ , depends on the number of fractional coordinates of the corresponding solution \mathbf{y} of (ConfLP). The following lemma shows how to produce optima of (ConfLP) with small support. We emphasize that our proximity theorem does not require that the fractional solution be optimal but rather conf-optimal.

Lemma 6 *There is an algorithm that finds an optimal vertex solution \mathbf{y}^* of (ConfLP) with $|\text{supp}(\mathbf{y}^*)| \leq r + \tau$ and at most $2r$ fractional coordinates, and a conf-optimal solution $\mathbf{x}^* = \varphi(\mathbf{y}^*)$ of (HugeCP) with at most $2r$ fractional bricks, in time $g_1(E_2)^{\mathcal{O}(s)} \text{poly}(rt\tau \log \|f_{\max}, \mathbf{1}, \mathbf{u}, \mathbf{b}, \boldsymbol{\mu}, E\|_{\infty})$.*

Proof The proof has three parts. First, we describe how to find an optimal basic solution of the dual of (ConfLP). Next, we identify $r + \tau$ inequalities of this dual which fully determine the optimal dual LP solution. Finally, we show how to use this information to solve (ConfLP) itself.

Recall that τ is the number of brick types in the huge N -fold instance. Since (ConfLP) has exponentially many variables, we take the standard approach and solve the dual LP of (ConfLP) by the ellipsoid method and the equivalence of optimization

and separation. The Dual LP of (ConfLP) in variables $\alpha \in \mathbb{R}^r, \beta \in \mathbb{R}^\tau$ is:

$$\begin{aligned} \max \quad & \mathbf{b}^0 \alpha + \sum_{i=1}^{\tau} \mu^i \beta^i \\ \text{s.t.} \quad & (\alpha E_1^i) \mathbf{c} - f^i(\mathbf{c}) \leq -\beta^i \quad \forall i \in [\tau], \forall \mathbf{c} \in \mathcal{C}^i. \end{aligned} \tag{9}$$

To verify feasibility of (α, β) , we need, for each $i \in [\tau]$, to maximize the left-hand side of (9) over all $\mathbf{c} \in \mathcal{C}^i$ and check if it is at most $-\beta^i$. This corresponds to finding integer variables \mathbf{c} which for given (α, β) solve

$$\min (f^i(\mathbf{c}) - (\alpha E_1^i) \mathbf{c}) = - \max ((\alpha E_1^i) \mathbf{c} - f^i(\mathbf{c})) : E_2^i \mathbf{c} = \mathbf{b}^i, \mathbf{l}^i \leq \mathbf{c} \leq \mathbf{u}^i, \mathbf{c} \in \mathbb{Z}^l.$$

This program can be solved in time $T''' \leq g_1(E_2)^{O(s)} l^3 \cdot \text{poly}(\log \|\mathbf{b}^i, \mathbf{l}^i, \mathbf{u}^i, \|E\|_\infty\|_\infty)$ [33, Theorem 4].

Grötschel et al. [18, Theorem 6.4.9] show that an optimal solution of an LP (even one which is a vertex [18, Remark 6.5.2]) can be found in a number of calls to a separation oracle which is polynomial in the dimension and the encoding length of the inequalities returned by a separation oracle. Clearly the inequalities (9) have encoding length bounded by $\log \|f_{\max}, \mathbf{l}, \mathbf{u}, \mathbf{b}, \mu\|_\infty$ and thus $T = \text{poly}(r\tau \log \|f_{\max}, \mathbf{l}, \mathbf{u}, \mathbf{b}, \mu, E\|_\infty)$ calls to a separation oracle are sufficient to find an optimal vertex solution, which amounts to $T \cdot T'''$ arithmetic operations.

Next, we will identify $r + \tau$ inequalities determining the previously found optimal vertex solution of the dual of (ConfLP). Observe that the dimension of the dual LP is the number of rows of the primal LP, which is $r + \tau$. Since each point in $(r + \tau)$ -dimensional space is fully determined by $r + \tau$ linearly independent inequalities, there must exist a subset I of $r + \tau$ inequalities among the T inequalities considered by the ellipsoid method which fully determines the dual optimum. We can find them as follows.

We initialize I to be the empty set. Taking the T considered inequalities one by one, we process the inequality if it is satisfied as equality by the given optimal basic solution for the dual LP, and we discard other inequalities. If we process the current inequality and either some inequality of I or the present inequality is dominated² by an inequality that can be obtained as a non-negative linear combination of the others, discard it; otherwise, include it in I and continue. Testing whether an inequality $\mathbf{d}\mathbf{z} \leq e'$ is dominated by a non-negative combination of a system of inequalities $D\mathbf{z} \leq \mathbf{e}$ can be decided by solving

$$\min \alpha \mathbf{e} \quad \text{s.t.} \quad \alpha^\top D = \mathbf{d}, \alpha \geq \mathbf{0}, \tag{10}$$

and checking whether the optimal value is at most e' . If it is, then the solution α encodes a non-negative linear combination of the inequalities $D\mathbf{z} \leq \mathbf{e}$ which yields an inequality dominating $\mathbf{d}\mathbf{z} \leq e'$, and if it is not, then such a combination does

² An inequality $\mathbf{a}\mathbf{x} \leq b$ is dominated by $\mathbf{c}\mathbf{x} \leq d$ if for every \mathbf{x} such that $\mathbf{c}\mathbf{x} \leq d$ we also have $\mathbf{a}\mathbf{x} \leq b$.

not exists. Thus, when a new inequality is considered, we solve (10) for at most $r + \tau$ inequalities (the new one and all less than $r + \tau$ already selected ones), and there are at most T inequalities considered. The time needed to solve (10) is $\text{poly}(r + \tau, \log \|\mathbf{l}, \mathbf{u}, \mathbf{b}, f_{\max}, E\|_{\infty})$ because its dimension is at most $r + \tau$ and its encoding length is at most $\log \|\mathbf{l}, \mathbf{u}, \mathbf{b}, f_{\max}, E\|_{\infty}$. Altogether, we need time

$$T \cdot (r + \tau) \cdot \text{poly}(r + \tau, \log \|\mathbf{l}, \mathbf{u}, \mathbf{b}, f_{\max}, E\|_{\infty}) \leq \text{poly}(r\tau \log \|f_{\max}, \mathbf{l}, \mathbf{u}, \mathbf{b}, \mu, E\|_{\infty}) =: T'.$$

Finally, let the *restricted (ConfLP)* be the (ConfLP) restricted to the variables corresponding to the inequalities in I . We claim that an optimal solution to the restricted (ConfLP) is also an optimal solution to (ConfLP). To see that, use LP duality: the optimal objective value of the dual LP restricted to inequalities in I is the same as one of the dual optima, and thus an optimal solution of the restricted (ConfLP) must be an optimal solution of (ConfLP). We solve the restricted (ConfLP) using any polynomial LP algorithm in time $T'' \leq \text{poly}((r + \tau), \log \|f_{\max}, \mathbf{l}, \mathbf{u}, \mu, \mathbf{b}^0, E\|_{\infty})$. The resulting total time complexity is thus $T \cdot T''' + T' + T''$ to construct the restricted (ConfLP) instance and time T'' to solve it, $T \cdot T''' + T' + T''$ total, which is upper bounded by $g_1(E_2)^{\mathcal{O}(s)} \text{poly}(r\tau \log \|f_{\max}, \mathbf{l}, \mathbf{u}, \mathbf{b}, \mu, E\|_{\infty})$, as claimed.

Let \mathbf{y}^* be an optimum of (ConfLP) we have thus obtained. Since $|I| \leq r + \tau$, the support of \mathbf{y}^* is of size at most $r + \tau$. By Lemma 5, \mathbf{y}^* has at most $2r$ fractional coordinates. Now setting $\mathbf{x}^* = \varphi(\mathbf{y}^*)$ is enough, since we have already argued (see definition of φ) that \mathbf{x}^* has at most as many fractional bricks as \mathbf{y}^* has fractional coordinates and \mathbf{x}^* can be computed from \mathbf{y}^* in $\mathcal{O}(r + \tau)$ time. \square

3.3 Proximity theorem

Let us give a plan for the next subsection. We wish to prove that for every conf-optimal solution \mathbf{x}^* of (HugeCP) there is an integer solution \mathbf{z}^* of (HugeIP) nearby. In the following, let \mathbf{x}^* be a conf-optimal solution of (HugeCP) and \mathbf{z}^* be an optimal solution of (HugeIP) minimizing $\|\mathbf{x}^* - \mathbf{z}^*\|_1$. A technique for proving proximity theorems which was introduced by Eisenbrand and Weismantel [11] works as follows. A vector $\mathbf{h} \in \mathbb{Z}^{Nt}$ is called a *cycle* of $\mathbf{x}^* - \mathbf{z}^*$ if $\mathbf{h} \neq \mathbf{0}$, $E^{(N)}\mathbf{h} = \mathbf{0}$, and $\mathbf{h} \sqsubseteq \mathbf{x}^* - \mathbf{z}^*$. It is not too difficult to see that if \mathbf{x}' is an optimal (not necessarily conf-optimal) solution of (HugeCP) with the objective f , then there cannot exist a cycle of $\mathbf{x}' - \mathbf{z}^*$ (cf. proof of Lemma 9). Based on a certain decomposition of $\mathbf{x}' - \mathbf{z}^*$ into integer and fractional smaller dimensional vectors and by an application of the Steinitz Lemma, the existence of a cycle is proven unless $\|\mathbf{x}' - \mathbf{z}^*\|_1$ is roughly bounded by the number of fractional bricks of \mathbf{x}' . However, we cannot apply this technique directly as an optimal solution \mathbf{x}' of (HugeCP) might have many fractional bricks. At the same time, an existence of a cycle \mathbf{h} of $\mathbf{x}^* - \mathbf{z}^*$ does not necessarily contradict that $\|\mathbf{x}^* - \mathbf{z}^*\|_1$ is minimal, because $\mathbf{x}^* + \mathbf{h}$ might not be a configurable solution, which is an essential part of the argument.

All of this leads us to introduce a stronger notion of a cycle. We say that $\mathbf{h} \in \mathbb{Z}^{Nt}$ is a *configurable cycle* of $\mathbf{x}^* - \mathbf{z}^*$ (with respect to \mathbf{x}^*) if (1) \mathbf{h} is a cycle of $\mathbf{x}^* - \mathbf{z}^*$, (2) for each brick $j \in [N]$ of type $i \in [\tau]$ there exists an \hat{f} -optimal decomposition Γ^j of

$(\mathbf{x}^*)^j$ such that we may write $\mathbf{h}^j = \sum_{(\mathbf{c}, \lambda_{\mathbf{c}}) \in \Gamma^j} \lambda_{\mathbf{c}} \mathbf{h}_{\mathbf{c}}$, and **(3)** for each $(\mathbf{c}, \lambda_{\mathbf{c}}) \in \Gamma^j$ we have $\mathbf{h}_{\mathbf{c}} \sqsubseteq \mathbf{c} - (\mathbf{z}^*)^j$ and $\mathbf{h}_{\mathbf{c}} \in \text{Ker}_{\mathbb{Z}}(E_2^i)$. Soon we will show that if $\|\mathbf{x}^* - \mathbf{z}^*\|_1$ is minimal, $\mathbf{x}^* - \mathbf{z}^*$ does not have a configurable cycle. The next task becomes to show how large must $\|\mathbf{x}^* - \mathbf{z}^*\|_1$ be in order for a configurable cycle to exist. Recall that the technique of Eisenbrand and Weismantel [11] can be used to rule out an existence of a (regular) cycle, not a configurable cycle. To overcome this, we “lift” both \mathbf{x}^* and \mathbf{z}^* to a higher-dimensional space and show that a cycle in this space corresponds to a configurable cycle in the original space. Only then are we ready to prove a proximity bound using the aforementioned technique.

Lemma 7 *If \mathbf{h} is a configurable cycle of $\mathbf{x}^* - \mathbf{z}^*$, then $\mathbf{x}^* - \mathbf{h}$ is configurable.*

Proof Fix $j \in [N]$. Let \mathbf{p} be the brick $(\mathbf{x}^* - \mathbf{h})^j$ and let $i \in [\tau]$ be its type. Now \mathbf{p} can be written as $\mathbf{p} = \sum_{(\mathbf{c}, \lambda_{\mathbf{c}}) \in \Gamma^j} \lambda_{\mathbf{c}} (\mathbf{c} - \mathbf{h}_{\mathbf{c}})$. Furthermore, we have $E_2^i(\mathbf{c} - \mathbf{h}_{\mathbf{c}}) = E_2^i \mathbf{c} = \mathbf{b}^j$, and, by $\mathbf{h} \sqsubseteq \mathbf{x}^* - \mathbf{z}^*$, we also have $\mathbf{l} \leq \mathbf{x}^* - \mathbf{h} \leq \mathbf{u}$. \square

We now need a technical lemma:

Lemma 8 *Let \mathbf{x}^* be a conf-optimal solution of (HugeCP), let \mathbf{z}^* be an optimum of (HugeIP), and let \mathbf{h}^* be a configurable cycle of $\mathbf{x}^* - \mathbf{z}^*$. Then*

$$\hat{f}(\mathbf{z}^* + \mathbf{h}^*) + \hat{f}(\mathbf{x}^* - \mathbf{h}^*) \leq \hat{f}(\mathbf{z}^*) + \hat{f}(\mathbf{x}^*) . \tag{11}$$

Proof We begin by a simple observation: let $g : \mathbb{R} \rightarrow \mathbb{R}$ be a convex function, $x \in \mathbb{R}$, $z \in \mathbb{Z}$, and $r \in \mathbb{Z}$ be such that $r \sqsubseteq x - z$ (that is, there is some ρ , $0 \leq \rho \leq 1$, such that $r = \rho \cdot (x - z)$). By convexity of g we have that

$$g(z + r) + g(x - r) \leq g(z) + g(x) . \tag{12}$$

Fix $j \in [N]$ and $\mathbf{z} = (\mathbf{z}^*)^j$, $\mathbf{x} = (\mathbf{x}^*)^j$, $\mathbf{h} = (\mathbf{h}^*)^j$, and let i be the type of brick j . Since \mathbf{h}^* is a configurable cycle there exists an \hat{f} -optimal decomposition Γ of \mathbf{x} such that, for each $(\mathbf{c}, \lambda_{\mathbf{c}}) \in \Gamma$, there exists a $\mathbf{h}_{\mathbf{c}} \sqsubseteq \mathbf{c} - \mathbf{z}$, $\mathbf{h}_{\mathbf{c}} \in \text{Ker}_{\mathbb{Z}}(E_2^i)$, and $\mathbf{h} = \sum_{(\mathbf{c}, \lambda_{\mathbf{c}}) \in \Gamma} \lambda_{\mathbf{c}} \mathbf{h}_{\mathbf{c}}$. Due to separability of f we may apply (12) independently to each coordinate, obtaining for each \mathbf{c}

$$f^i(\mathbf{z} + \mathbf{h}_{\mathbf{c}}) + f^i(\mathbf{c} - \mathbf{h}_{\mathbf{c}}) \leq f^i(\mathbf{z}) + f^i(\mathbf{c}) .$$

Since all arguments of f^i are integral, we immediately get

$$\hat{f}^i(\mathbf{z} + \mathbf{h}_{\mathbf{c}}) + \hat{f}^i(\mathbf{c} - \mathbf{h}_{\mathbf{c}}) \leq \hat{f}^i(\mathbf{z}) + \hat{f}^i(\mathbf{c}) .$$

Aggregating according to Γ , we get (recall that we have $\sum_{(\mathbf{c}, \lambda_{\mathbf{c}}) \in \Gamma} \lambda_{\mathbf{c}} = 1$)

$$\begin{aligned} \sum_{(\mathbf{c}, \lambda_{\mathbf{c}}) \in \Gamma} \lambda_{\mathbf{c}} \left(\hat{f}^i(\mathbf{z} + \mathbf{h}_{\mathbf{c}}) + \hat{f}^i(\mathbf{c} - \mathbf{h}_{\mathbf{c}}) \right) &\leq \sum_{(\mathbf{c}, \lambda_{\mathbf{c}}) \in \Gamma} \lambda_{\mathbf{c}} \left(\hat{f}^i(\mathbf{z}) + \hat{f}^i(\mathbf{c}) \right) \\ &= \hat{f}^i(\mathbf{z}) + \sum_{(\mathbf{c}, \lambda_{\mathbf{c}}) \in \Gamma} \lambda_{\mathbf{c}} \hat{f}^i(\mathbf{c}), \end{aligned}$$

where by \hat{f} -optimality of Γ the right-hand side is equal to $\hat{f}^i(\mathbf{z}) + \hat{f}^i(\mathbf{x})$. As for the left-hand side, observe that decompositions $\Gamma' = \{(\mathbf{z} + \mathbf{h}_c, \lambda_c) \mid (\mathbf{c}, \lambda_c) \in \Gamma\}$ and $\Gamma'' = \{(\mathbf{c} - \mathbf{h}_c, \lambda_c) \mid (\mathbf{c}, \lambda_c) \in \Gamma\}$ satisfy $\sum \Gamma' = \mathbf{z} + \mathbf{h}$ and $\sum \Gamma'' = \mathbf{x} - \mathbf{h}$ but are only feasible (not necessarily optimal) solutions of (6). Thus, we have

$$\hat{f}^i(\mathbf{z} + \mathbf{h}) + \hat{f}^i(\mathbf{x} - \mathbf{h}) \leq \sum_{(\mathbf{c}, \lambda_c) \in \Gamma} \lambda_c \left(\hat{f}^i(\mathbf{z} + \mathbf{h}_c) + \hat{f}^i(\mathbf{c} - \mathbf{h}_c) \right).$$

Combining over Γ then yields

$$\hat{f}^i(\mathbf{z} + \mathbf{h}) + \hat{f}^i(\mathbf{x} - \mathbf{h}) \leq \hat{f}^i(\mathbf{z}) + \hat{f}^i(\mathbf{x}),$$

and since we have proven this claim for every brick j , aggregation over bricks concludes the proof of the main claim (11). □

Let us show that if \mathbf{x}^* and \mathbf{z}^* are as stated, then there is no configurable cycle of $\mathbf{x}^* - \mathbf{z}^*$.

Lemma 9 *Let \mathbf{x}^* be a conf-optimal solution of (HugeCP) and let \mathbf{z}^* be an optimal solution of (HugeIP) such that $\|\mathbf{x}^* - \mathbf{z}^*\|_1$ is minimal. Then there is no configurable cycle of $\mathbf{x}^* - \mathbf{z}^*$.*

Proof For the sake of contradiction, suppose that there exists a configurable cycle \mathbf{h}^* of $\mathbf{x}^* - \mathbf{z}^*$. By Lemma 8, one of two cases must occur:

Case 1: $\hat{f}(\mathbf{z}^* + \mathbf{h}^*) \leq \hat{f}(\mathbf{z}^*)$. Then $\mathbf{z}^* + \mathbf{h}^*$ is an optimal integer solution (by $\mathbf{h} \sqsubseteq \mathbf{x}^* - \mathbf{z}^*$ we have $\mathbf{l} \leq \mathbf{z}^* + \mathbf{h} \leq \mathbf{u}$ and by $\mathbf{h}^* \in \ker_{\mathbb{Z}}(E^{(N)})$ we have $E^{(N)}(\mathbf{z}^* + \mathbf{h}) = \mathbf{b}$) which is closer to \mathbf{x}^* , a contradiction to minimality of $\|\mathbf{x}^* - \mathbf{z}^*\|_1$.

Case 2: $\hat{f}(\mathbf{x}^* - \mathbf{h}^*) < \hat{f}(\mathbf{x}^*)$. Since \mathbf{h}^* is a configurable cycle, Lemma 7 states that $\mathbf{x}^* - \mathbf{h}^*$ is configurable, so we have a contradiction with conf-optimality of \mathbf{x}^* . □

Overview of the remainder of the proof

In order to use existing proximity arguments to bound the norm of a cycle, our plan is to move into an extended (higher-dimensional) space which corresponds to decomposing each brick \mathbf{x}^i of \mathbf{x}^* into configurations as $\mathbf{x}^i = \sum_c \lambda_c \mathbf{c}$ – each summand becomes a new brick in the extended space.

We denote this new higher-dimensional representation of \mathbf{x}^* with respect to Γ as $\uparrow \mathbf{x}^*$ and call it the *rise* of \mathbf{x}^* , and define similarly the rise of \mathbf{z}^* (with respect to a given decomposition of each brick of \mathbf{x}^*). The situation gets very delicate at this point.

First, we require that each decomposition of a brick of \mathbf{x}^* is optimal with respect to the auxiliary objective \hat{f} so that we can use the argument about non-existence of a cycle. Second, because the proximity bound depends on the number of fractional bricks of $\uparrow \mathbf{x}^*$, we require that the decomposition of each brick is small, i.e., into only few elements. Third, we require that each coefficient λ_c is of the form $1/q_c$ for an integer q_c , because we need to ensure that, for a corresponding cycle brick \mathbf{h}_c , $\lambda_c^{-1} \mathbf{h}_c$

is an integer vector, so $\lambda_{\mathbf{c}}^{-1}$ has to be an integer. To ensure the second and third condition simultaneously, we first show that there is a decomposition of each brick of size at most $t + 1$ and with each coefficient bounded by P , and then show that each fraction p/q can be written as an *Egyptian fraction* $p/q = 1/a_1 + 1/a_2 + \dots + 1/a_c$ with $c \leq 2 \log_2 q$ (Lemmas 10–12). (Bounds on the length of Egyptian fractions have been studied in the past and our bound is not the best possible, but in order to use our proximity theorem, we need exact and not merely asymptotic bounds, so we prove this worse but exact bound of $2 \log_2 q$.) We call a decomposition of a brick satisfying all three criteria given above a *small scalable decomposition*.

Fix a small scalable decomposition for each brick of \mathbf{x}^* , and let $\uparrow \mathbf{x}^*$ be the rise of \mathbf{x}^* with respect to this decomposition. Since this decomposition is small, $\uparrow \mathbf{x}^*$ has at most $\text{poly}(\|E\|_\infty, r, s)$ fractional bricks. Moreover, the other properties above allow us to say the following: if \mathbf{r} is a cycle of $\uparrow \mathbf{x}^* - \uparrow \mathbf{z}^*$, then the compression of \mathbf{r} back to the original space is a configurable cycle of $\mathbf{x}^* - \mathbf{z}^*$ (Lemma 14). So in order to bound $\|\mathbf{x}^* - \mathbf{z}^*\|_1$, it suffices (by triangle inequality) to bound $\|\uparrow \mathbf{x}^* - \uparrow \mathbf{z}^*\|_1$. We do this by adapting the approach of Eisenbrand and Weismantel [11] to bound the length of any cycle \mathbf{r} of $\uparrow \mathbf{x}^* - \uparrow \mathbf{z}^*$.

The remainder of the proof

We say that $|\Gamma|$ is the *size* of the decomposition. Let us show that for each brick, there exists an \hat{f} -optimal decomposition whose coefficients have small encoding length, and its size is small. For any matrix A , define $g_\infty(A) = \max_{\mathbf{g} \in \mathcal{G}(A)} \|\mathbf{g}\|_\infty$.

Lemma 10 *Each brick of \mathbf{x}^* of type i has an \hat{f} -optimal decomposition Γ*

1. of size at most $t + 1$, and
2. $\max_{(\mathbf{c}, \lambda_{\mathbf{c}} = p_{\mathbf{c}}/q_{\mathbf{c}}) \in \Gamma} \{p_{\mathbf{c}}, q_{\mathbf{c}}\} \leq (t + 1)!((2t - 2)g_\infty(E_2^i))^{t+1} \leq (t + 1)^{(t+1)} (g_1(E_2))^{(t+2)} \leq (t + 1)^{(t+1)}(s\|E_2^i\|_\infty + 1)^{(s+1)(t+2)}$.

Proof An \hat{f} -optimal decomposition corresponds to a solution of the LP (6). We will argue that there is a solution whose support is composed of columns which do not differ by much, which corresponds to a solution of an LP with small coefficients, and the claimed bound can then be obtained by Cramer’s rule.

Specifically, we claim that there exists an \hat{f} -optimal decomposition Γ which corresponds to an optimal solution λ of (6) such that there exists a point $\zeta \in \mathbb{Z}^t$ and if $\mathbf{c} \in \text{supp}(\lambda)$, then $\|\mathbf{c} - \zeta\|_\infty \leq (t - 1)g_\infty(E_2^i)$. For a solution λ of (6), define $R' := \max_{\mathbf{c}, \mathbf{c}' \in \text{supp}(\lambda)} \|\mathbf{c} - \mathbf{c}'\|_\infty$ to be the longest side of the bounding box of all $\mathbf{c} \in \text{supp}(\lambda)$. For a point $\zeta \in \mathbb{Z}^t$, say, for $\mathbf{c} \in \text{supp}(\lambda)$, that a coordinate $j \in [t]$ is *tight* if $c_j = \zeta_j - \lceil \frac{R'}{2} \rceil$ or $c_j = \zeta_j + \lceil \frac{R'}{2} \rceil$, and define $S = \sum_{\mathbf{c} \in \text{supp}(\lambda)} \sum_{j=1}^t \lambda_{\mathbf{c}} [j \text{ is tight in } \mathbf{c}]$ (where “[X]” is an indicator of the statement X) to be the weighted number of tight coordinates. Now let $\zeta \in \mathbb{Z}^t$ be any point which is an integer center of the bounding box (i.e., $\|\mathbf{c} - \zeta\|_\infty \leq \lceil \frac{R'}{2} \rceil$ for all $\mathbf{c} \in \text{supp}(\lambda)$) and which minimizes S . For contradiction assume that λ is an optimal solution of (6) which minimizes R' and S (lexicographically in this order) and $R' > (2t - 2)g_\infty(E_2^i)$. Assuming Γ is a decomposition of a brick \mathbf{c} of type i , we have $\mathbf{c}, \mathbf{c}' \in \mathcal{C}^i = \{\tilde{\mathbf{c}} \in \mathbb{Z}^t \mid E_2^i \tilde{\mathbf{c}} = \mathbf{b}^i, \mathbf{l}^i \leq \tilde{\mathbf{c}} \leq \mathbf{u}^i\}$

and thus $\mathbf{c} - \mathbf{c}' \in \text{Ker}_{\mathbb{Z}}(E_2^i)$. By Proposition 1 we may write $\mathbf{c} - \mathbf{c}' = \sum_{j=1}^{2t-2} \gamma_j \mathbf{g}_j$ with $\mathbf{g}_j \in \mathcal{G}(E_2^i)$ and $\mathbf{g}_j \sqsubseteq \mathbf{c} - \mathbf{c}'$ for all $j \in [2t - 2]$. Note that because $\|\mathbf{c} - \mathbf{c}'\|_{\infty} > R := (2t - 2)g_{\infty}(E_2^i)$, we have that there exists $j \in [2t - 2]$ such that $\gamma_j > 1$. Hence $\mathbf{g} := \sum_{j=1}^{2t-2} \lfloor \frac{\gamma_j}{2} \rfloor \mathbf{g}_j$ satisfies $\mathbf{g} \neq \mathbf{0}$. Let $\bar{\mathbf{c}} := \mathbf{c} - \mathbf{g}$, and $\bar{\mathbf{c}}' := \mathbf{c}' + \mathbf{g}$.

First, because $\bar{\mathbf{c}} - \bar{\mathbf{c}}' = (\mathbf{c} - \mathbf{c}') - 2\mathbf{g} = \sum_{j=1}^{2t-2} (\gamma_j - 2\lfloor \frac{\gamma_j}{2} \rfloor) \mathbf{g}_j$, we may bound $\|\bar{\mathbf{c}} - \bar{\mathbf{c}}'\|_{\infty} \leq (2t - 2)g_{\infty}(E_2^i) = R$. Second, by the conformality of the decomposition, $\bar{\mathbf{c}}, \bar{\mathbf{c}}' \in \mathcal{C}^i$. Third, by separable convex superadditivity (Proposition 2), we have that $f(\mathbf{c}) + f(\mathbf{c}') \geq f(\bar{\mathbf{c}}) + f(\bar{\mathbf{c}}')$. Fourth, there exist a coordinate $j \in [t]$ such that $|c_j - c'_j| = R'$ but, since $\|\bar{\mathbf{c}} - \bar{\mathbf{c}}'\|_{\infty} \leq R$, $|\bar{c}_j - \bar{c}'_j| \leq R < R'$ and thus j is no longer a tight coordinate for either $\bar{\mathbf{c}}$ or $\bar{\mathbf{c}}'$ (or both), and no new tight coordinates can be introduced because $R < R'$. Without loss of generality, let $\lambda_{\mathbf{c}} \leq \lambda_{\mathbf{c}'}$. Now initialize $\lambda' := \lambda$ and modify it by setting $\lambda'_{\bar{\mathbf{c}}}, \lambda'_{\bar{\mathbf{c}}'} := \lambda_{\mathbf{c}}, \lambda'_{\bar{\mathbf{c}}} := 0, \lambda'_{\bar{\mathbf{c}}'} := \lambda_{\mathbf{c}'} - \lambda_{\mathbf{c}}$. By our arguments above, λ' is another optimal solution of (6) but the weighted number S of tight coordinates has decreased by the fourth point, a contradiction.

Thus, there exists a point $\boldsymbol{\zeta} \in \mathbb{Z}^t$ and an optimal solution λ of (6) such that for each $\mathbf{c} \in \text{supp}(\lambda)$, it holds that $\|\mathbf{c} - \boldsymbol{\zeta}\|_{\infty} \leq R/2 = (t - 1)g_{\infty}(E_2^i)$. We obtain the following reduced LP from (6) by deleting all columns \mathbf{c} with $\|\mathbf{c} - \boldsymbol{\zeta}\|_{\infty} > R/2$, and denote the remaining set of columns by $\bar{\mathcal{C}}^i$:

$$\min \sum_{\mathbf{c} \in \bar{\mathcal{C}}^i} \lambda_{\mathbf{c}} f^i(\mathbf{c}) \quad \text{s.t.} \quad \sum_{\mathbf{c} \in \bar{\mathcal{C}}^i} \lambda_{\mathbf{c}} \mathbf{c} = \mathbf{x}^j, \|\lambda\|_1 = 1, \lambda \geq \mathbf{0} . \tag{13}$$

This LP is equivalent to one obtained by subtracting $\boldsymbol{\zeta}$ from all columns and the right hand side:

$$\min \sum_{\mathbf{c} \in \bar{\mathcal{C}}^i} \lambda_{\mathbf{c}} f^i(\mathbf{c}) \quad \text{s.t.} \quad \sum_{\mathbf{c} \in \bar{\mathcal{C}}^i} \lambda_{\mathbf{c}} (\mathbf{c} - \boldsymbol{\zeta}) = (\mathbf{x}^j - \boldsymbol{\zeta}), \|\lambda\|_1 = 1, \lambda \geq \mathbf{0} . \tag{14}$$

Now, this LP has $t + 1$ rows and its columns have the largest coefficient bounded by $R/2$ in absolute value. A basic solution λ has $|\text{supp}(\lambda)| \leq t + 1$ and, by Cramer’s rule, the denominator of each $\lambda_{\mathbf{c}}$ is bounded by $(t + 1)!$ times the largest coefficient to the power of $t + 1$, thus bounded by

$$(t + 1)!(R/2)^{(t+1)} \leq (t + 1)!((t - 1)g_{\infty}(E_2^i))^{(t+1)} \leq (t + 1)^{(t+1)}(g_1(E_2))^{(t+2)}.$$

In the worst case, we can bound this as

$$(t + 1)^{(t+1)}(s\|E_2^i\|_{\infty} + 1)^{s(t+2)},$$

where we use

$$g_{\infty}(E_2^i) \leq \|E_2^i\|_{\infty}(2s\|E_2^i\|_{\infty} + 1)^s$$

[12, Lemma 2].

□

Next, we will need the notion of an Egyptian fraction. For a rational number p/q , $p, q \in \mathbb{N}$, its *Egyptian fraction* is a finite sum of distinct unit fractions such that

$$\frac{p}{q} = \frac{1}{q_1} + \frac{1}{q_2} + \dots + \frac{1}{q_k},$$

for $q_1, \dots, q_k \in \mathbb{N}$ distinct. Call the number of terms k the *length* of the Egyptian fraction. Vose [43] has proven that any p/q has an Egyptian fraction of length $\mathcal{O}(\sqrt{\log q})$. Since our algorithm requires an exact bound, we present the following weaker yet exact result:

Lemma 11 (Egyptian Fractions) *Let $p, q \in \mathbb{N}$, $1 \leq p < q$. Then p/q has an Egyptian fraction of length at most $2(\log_2 q) + 1$ and all denominators are at most q^2 .*

Proof Let $a = 2^k$ be largest such that $a < q$, so $k = \lceil (\log_2 q) - 1 \rceil < \log_2 q$. Write $ap = bq + r$, $0 \leq r < q$. Note that $p < q \implies b < a$ and $q \leq 2a \implies r < 2a$. Now let $(b_{k-1}, \dots, b_1, b_0)$ be the binary representation of $b < a$ so $b = \sum_{i=0}^{k-1} 2^i b_i$ and $e(r_k, \dots, r_1, r_0)$ be that of $r < 2a$ so $r = \sum_{i=0}^k r_i 2^i$. Then we have

$$\frac{p}{q} = \frac{ap}{aq} = \frac{bq + r}{aq} = \frac{b}{a} + \frac{1}{q} \frac{r}{a} = \sum_{i=0}^{k-1} \frac{b_i}{2^{k-i}} + \sum_{i=0}^k \frac{r_i}{q \cdot 2^{k-i}},$$

where $b_i, r_i \in \{0, 1\}$, so a sum of at most $2k + 1 \leq 2(\log_2 q) + 1$ terms with all denominators $d_i \leq q2^k = qa \leq q^2$. Moreover, all denominators in the first sum are distinct and at most 2^k , and all in the second sum are distinct and at least $q > 2^k$, hence all distinct, so this is an Egyptian fraction of p/q of length $2(\log_2 q) + 1$ and denominators are at most q^2 . \square

Recall that our goal is to obtain a configurable cycle. However, for that we also need a special form of a decomposition. Say that Γ is a *scalable decomposition* of a brick $(\mathbf{x}^*)^j$ of type i if it is a \hat{f} -optimal decomposition, and for each $(\mathbf{c}_\gamma, \lambda_\gamma) \in \Gamma$, λ_γ is of the form $1/q_\gamma$ for some $q_\gamma \in \mathbb{N}$. We note that in what follows we do not need an algorithm computing a scalable decomposition, only the following existence statement.

Lemma 12 *Each brick of \mathbf{x}^* has a scalable decomposition of size at most $\kappa_1 \cdot t^3 \log(t \|E_2\|_\infty)$, where $\kappa_1 = 52$.*

Proof Fix $j \in [N]$. Let $\mathbf{x} = (\mathbf{x}^*)^j$ be a brick of \mathbf{x}^* of type i . By Lemma 10, there exists an \hat{f} -optimal decomposition of \mathbf{x} of size $t + 1$ where each coefficient $\lambda_{\mathbf{c}} = p_{\mathbf{c}}/q_{\mathbf{c}}$ satisfies $p_{\mathbf{c}}, q_{\mathbf{c}} \leq (t + 1)^{(t+1)} (s \|E_2^i\|_\infty + 1)^{(s+1)(t+2)}$. For each \mathbf{c} in the decomposition now express $\lambda_{\mathbf{c}}$ as an Egyptian fraction:

$$\lambda_{\mathbf{c}} = \frac{p_{\mathbf{c}}}{q_{\mathbf{c}}} = \frac{1}{a_1} + \frac{1}{a_2} + \dots + \frac{1}{a_e}.$$

By Lemma 11,

$$\begin{aligned} \epsilon &\leq 2(\log_2 q_\epsilon) + 1 = 2 \left(\log \left((t + 1)^{(t+1)} (s \|E_2^i\|_\infty + 1)^{(s+1)(t+2)} \right) \right) + 1 \\ &\leq 25st \log(st \|E_2^i\|_\infty) . \end{aligned}$$

Thus, the resulting decomposition is of size at most $(t + 1)25st \log(st \|E_2^i\|_\infty) \leq 2 \cdot 26t^3 \log(t \|E_2^i\|_\infty)$ (by $s \leq t$ this justifies the deletion of s in the $\log()$ at the cost of a factor of 2, so the last bound holds) and is scalable, since each coefficient is of the form $1/q_\gamma$ for some $q_\gamma \in \mathbb{N}$. \square

We will now show that we are guaranteed a configurable cycle of $\mathbf{x}^* - \mathbf{z}^*$ if there exists an analogue of a regular cycle of a certain “lifting” of \mathbf{x}^* and \mathbf{z}^* .

Fix for each brick of \mathbf{x}^* a scalable decomposition Γ^j . Let $\uparrow\mathbf{x}^*$ be the rise of \mathbf{x}^* defined as a vector obtained from \mathbf{x}^* by keeping every integer brick $(\mathbf{x}^*)^j$, and replacing every fractional brick $(\mathbf{x}^*)^j$ with $|\Gamma^j|$ terms $\lambda_\gamma \mathbf{c}_\gamma$, one for each $(\mathbf{c}_\gamma, \lambda_\gamma) \in \Gamma^j$. Observe that each brick of $\uparrow\mathbf{x}^*$ is of the form $\lambda_\mathbf{c} \mathbf{c}$ for some configuration \mathbf{c} and some coefficient $0 \leq \lambda_\mathbf{c} \leq 1$. Thus, for a brick $\lambda_\mathbf{c} \mathbf{c}$ we say that \mathbf{c} is its configuration, $\lambda_\mathbf{c}$ is its coefficient, and its type is identical to the type of brick it originated from; in particular, bricks which originated from an integer brick $\mathbf{p} = (\mathbf{x}^*)^j$ are of the form $\lambda_\mathbf{p} \mathbf{p}$ with $\lambda_\mathbf{p} = 1$. Let N' be the number of bricks of $\uparrow\mathbf{x}^*$ and define a mapping $\nu : [N'] \rightarrow [N]$ such that if a brick $j \in [N']$ of $\uparrow\mathbf{x}^*$ was defined from brick $\ell \in [N]$ of \mathbf{x}^* , then $\nu(j) = \ell$. The natural inverse ν^{-1} is defined such that, for $\ell \in [N]$, $\nu^{-1}(\ell)$ is the set of bricks of $\uparrow\mathbf{x}^*$ which originated from $(\mathbf{x}^*)^\ell$.

Lemma 13 *The vector $\uparrow\mathbf{x}^*$ has at most $\kappa_2 \cdot r \cdot t^3 \log(t \|E_2^1, \dots, E_2^r\|_\infty)$ fractional bricks, where $\kappa_2 = 2\kappa_1$.*

Proof By Lemma 6 there is a conf-optimal \mathbf{x}^* with at most $2r$ fractional bricks. By Lemma 12 for each fractional brick of \mathbf{x}^* of type i there is a scalable decomposition of size at most $\kappa_1 \cdot t^3 \log(t \|E_2^i\|_\infty) \leq \kappa_1 \cdot t^3 \log(t \|E_2^1, \dots, E_2^r\|_\infty)$. Thus, $\uparrow\mathbf{x}^*$ has at most $\kappa_1 \cdot t^3 \log(t \|E_2^1, \dots, E_2^r\|_\infty)$ fractional bricks for each fractional brick of \mathbf{x}^* , of which there are at most $2r$, totaling $2\kappa_1 \cdot r \cdot t^3 \log(t \|E_2^1, \dots, E_2^r\|_\infty)$ fractional bricks. \square

Denote by $\uparrow\mathbf{z}^* \in \mathbb{R}^{N't}$ the rise of \mathbf{z}^* (with respect to \mathbf{x}^*) defined as follows. Let $j \in [N']$, $\ell = \nu(j)$, and λ be the coefficient of the j -th brick of $\uparrow\mathbf{x}^*$. Then the j -th brick of $\uparrow\mathbf{z}^*$ is $(\uparrow\mathbf{z}^*)^j := \lambda(\mathbf{z}^*)^\ell$. Observe that $\|\uparrow\mathbf{x}^* - \uparrow\mathbf{z}^*\|_1 \geq \|\mathbf{x}^* - \mathbf{z}^*\|_1$ by applying triangle inequality to each brick and its decomposition individually and aggregating.

For any vector $\mathbf{x} \in \mathbb{R}^{N't}$, define the fall of \mathbf{x} as a vector $\downarrow\mathbf{x} \in \mathbb{R}^{Nt}$ such that for $\ell \in [N]$, $(\downarrow\mathbf{x})^\ell = \sum_{j \in \nu^{-1}(\ell)} \mathbf{x}^j$. We see that $\downarrow(\uparrow\mathbf{x}^*) = \mathbf{x}^*$ and $\downarrow(\uparrow\mathbf{z}^*) = \mathbf{z}^*$. Say that \mathbf{r} is a cycle of $\uparrow\mathbf{x}^* - \uparrow\mathbf{z}^*$ if $\mathbf{r} \sqsubseteq \uparrow\mathbf{x}^* - \uparrow\mathbf{z}^*$ and $\mathbf{r} \in \text{Ker}_{\mathbb{Z}}(E^{(N')})$.³

Lemma 14 *If \mathbf{r} is a cycle of $\uparrow\mathbf{x}^* - \uparrow\mathbf{z}^*$, then $\downarrow\mathbf{r}$ is a configurable cycle of $\mathbf{x}^* - \mathbf{z}^*$.*

³ Recall that $E^{(N')}$ is the N' -fold matrix formed from blocks E , see Eq. (1).

Proof To show that $\downarrow \mathbf{r}$ is a configurable cycle, we need to show that **(1)** $\downarrow \mathbf{r} \in \text{Ker}_{\mathbb{Z}}(E^{(N)})$ and, **(2)** for each brick $\mathbf{x} = (\mathbf{x}^*)^j$ of \mathbf{x}^* , there is an \hat{f} -optimal decomposition of \mathbf{x} such that $\mathbf{h} = (\downarrow \mathbf{r})^j$ decomposes accordingly. For the first part, $\downarrow \mathbf{r}$ is integral because it is obtained by summing bricks of \mathbf{r} , which is integral. Denote by $i(j)$ the type of a brick j (we abuse this notation; note that $i(j)$ for $j \in [N]$ may differ from $i(j)$ for $j \in [N']$, but context always makes clear what we mean). By the fact that $\mathbf{r} \in \text{Ker}_{\mathbb{Z}}(E^{(N')})$ and the definition of $\downarrow \mathbf{r}$, we have $\mathbf{0} = \sum_{j=1}^{N'} E_1^{i(j)} \mathbf{r}^j = \sum_{j=1}^N E_1^{i(j)} (\downarrow \mathbf{r})^j$, and, for each $\ell \in [N]$, $\mathbf{0} = \sum_{j \in v^{-1}(\ell)} E_2^{i(j)} \mathbf{r}^j = E_2^{i(\ell)} (\downarrow \mathbf{r})^\ell$, thus $\downarrow \mathbf{r} \in \text{Ker}_{\mathbb{Z}}(E^{(N)})$.

To see the second part, fix a brick $j \in [N]$ of type i and let $\mathbf{x} = (\mathbf{x}^*)^j$, $\mathbf{z} = (\mathbf{z}^*)^j$ and $\mathbf{h} = (\downarrow \mathbf{r})^j$. We need to show that $\mathbf{h} = \sum_{\gamma \in v^{-1}(j)} \mathbf{h}_\gamma$ can be written as $\sum_{\mathbf{c} \in \mathcal{C}^i} \lambda_{\mathbf{c}} \mathbf{h}_{\mathbf{c}}$ with $\mathbf{h}_{\mathbf{c}} \sqsubseteq \mathbf{c} - \mathbf{z}$ and $\mathbf{h}_{\mathbf{c}} \in \text{Ker}_{\mathbb{Z}}(E_2^i)$. By definition of $\uparrow \mathbf{x}^*$ and \mathbf{r} , there is a scalable decomposition Γ of \mathbf{x} (namely the one used to define $\uparrow \mathbf{x}^*$) such that for each $\gamma \in v^{-1}(j)$, $\mathbf{h}_\gamma \sqsubseteq \lambda_\gamma (\mathbf{c}_\gamma - \mathbf{z})$ and $\mathbf{h}_\gamma \in \text{Ker}_{\mathbb{Z}}(E_2^i)$. Thus we may write $\mathbf{h} = \sum_{\gamma \in v^{-1}(j)} \lambda_\gamma \cdot (\lambda_\gamma^{-1} \mathbf{h}_\gamma)$ with $\lambda_\gamma^{-1} \mathbf{h}_\gamma \sqsubseteq \mathbf{c}_\gamma - \mathbf{z}$ and $\lambda_\gamma^{-1} \mathbf{h}_\gamma$ integral by the fact that $\lambda_\gamma = 1/q_\gamma$ with $q_\gamma \in \mathbb{N}$, concluding the proof. \square

We are finally ready to use the Steinitz Lemma to derive a bound on $\|\mathbf{x}^* - \mathbf{z}^*\|_1$.

Theorem 2 *Let \mathbf{x}^* be a conf-optimal solution of (HugeCP) with at most $2r$ fractional bricks. Then there exists an optimal solution \mathbf{z}^* of (HugeIP) such that*

$$\begin{aligned} \|\mathbf{z}^* - \mathbf{x}^*\|_1 &\leq \left(\kappa_2 t^4 \log(t \|E_2^1, \dots, E_2^\tau\|_\infty) \right) (2r \|E_1\|_\infty g_1(E_2))^{r+2} \\ &\leq \left(\kappa_2 t^4 \log(t \|E_2^1, \dots, E_2^\tau\|_\infty) \right) (2r)^{r+2} (\|E\|_\infty s)^{3rs} . \end{aligned}$$

Proof Denote by \bar{E}_1 the first r rows of the matrix $E^{(N)}$. Let \mathbf{z}^* be an optimal integer solution such that $\|\mathbf{z}^* - \mathbf{x}^*\|_1$ is minimal, let $\uparrow \mathbf{x}^*$ be the rise of \mathbf{x}^* with at most $\kappa_2 \cdot r \cdot t^3 \log(t \|E_2^1, \dots, E_2^\tau\|_\infty)$ fractional bricks (see Lemma 13), let $\uparrow \mathbf{z}^*$ be the rise of \mathbf{z}^* with respect to \mathbf{x}^* , and let $\mathbf{q} = \uparrow \mathbf{x}^* - \uparrow \mathbf{z}^*$.

We want to get into the setting of the Steinitz Lemma, that is, to obtain a sequence of vectors with small ℓ_1 -norm and summing up to zero. To this end, we shall decompose $\bar{E}_1 \mathbf{q}$ in the following way; we stress that we have $\bar{E}_1 \mathbf{q} = \mathbf{0}$. For every integral brick \mathbf{q}^i of type $\ell \in [\tau]$ we have its decomposition $\mathbf{q}^i = \sum_j \mathbf{g}_j^i$ into elements of $\mathcal{G}(E_2^\ell)$ by the Positive Sum Property (Proposition 1); for each \mathbf{g}_j^i append $E_1^\ell \mathbf{g}_j^i$ into the sequence. For every fractional brick \mathbf{q}^i of type $\ell \in [\tau]$ we have its decomposition $\mathbf{q}^i = \sum_{j=1}^t \alpha_j \mathbf{g}_j^i$, $\alpha_j \geq 0$ for each j , into elements of $\mathcal{C}(E_2^\ell)$; for each \mathbf{g}_j^i append $\lfloor \alpha_j \rfloor$ copies of $E_1^\ell \mathbf{g}_j^i$ into the sequence, and finally append $E_1^\ell \{\alpha_j\} \mathbf{g}_j^i$. Observe that since $\uparrow \mathbf{x}^*$ has at most $\kappa_2 \cdot r \cdot t^3 \log(t \|E_2^1, \dots, E_2^\tau\|_\infty)$ fractional bricks (Lemma 13), so does \mathbf{q} , and thus we have appended $\leq t \cdot \kappa_2 \cdot r \cdot t^3 \log(t \|E_2^1, \dots, E_2^\tau\|_\infty) \leq \kappa_2 \cdot r \cdot t^4 \log(t \|E_2^1, \dots, E_2^\tau\|_\infty)$ fractional vectors into the sequence. Now we have a sequence

$$\mathbf{0}_1, \dots, \mathbf{0}_m, \mathbf{p}_{m+1}, \dots, \mathbf{p}_{m+f} \tag{15}$$

with m integer vectors $\mathbf{o}_1, \dots, \mathbf{o}_m$ and f fractional vectors $\mathbf{p}_{m+1}, \dots, \mathbf{p}_{m+f}$. Moreover, since, for each $i \in [\tau]$, $\mathcal{C}(E_2^i) \subseteq \mathcal{G}(E_2^i)$,

each vector has ℓ_∞ -norm of $\|E_1^1, \dots, E_1^\tau\|_\infty \cdot g_1(E_2)$ and they sum up to $\mathbf{0}$. Observe that $(m+f) \cdot g_1(E_2) \geq \|\mathbf{q}\|_1 = \|\uparrow \mathbf{x}^* - \uparrow \mathbf{z}^*\|_1 \geq \|\mathbf{x}^* - \mathbf{z}^*\|_1$. We now focus on bounding $m+f$. The Steinitz Lemma (Lemma 1) implies that there exists a permutation π such that the sequence (15) can be re-arranged as

$$\mathbf{v}_1, \dots, \mathbf{v}_{m+f}, \tag{16}$$

where \mathbf{v}_i is $\mathbf{o}_{\pi^{-1}(i)}$ if $i \in [1, m]$ and $\mathbf{p}_{\pi^{-1}(i)}$ if $i \in [m+1, m+f]$, respectively, and for each $1 \leq k \leq m+f$ the prefix sum $\mathbf{t}_k := \sum_{i=1}^k \mathbf{v}_i$ satisfies

$$\|\mathbf{t}_k\|_\infty \leq r \|E_1\|_\infty g_1(E_2).$$

We will now argue that there cannot be indices $1 \leq k_1 < \dots < k_{f+2} \leq f+m$ with

$$\mathbf{t}_{k_1} = \dots = \mathbf{t}_{k_{f+2}}, \tag{17}$$

which implies that $f+m$ is bounded by $f+1$ times the number of integer points of norm at most $r \|E_1\|_\infty g_1(E_2)$ and therefore,

$$\begin{aligned} \|\mathbf{x}^* - \mathbf{z}^*\|_1 &\leq \|\uparrow \mathbf{x}^* - \uparrow \mathbf{z}^*\|_1 \leq (f+1) (2r \|E_1\|_\infty g_1(E_2) + 1)^r \cdot g_1(E_2) \\ &\leq \kappa_2 \cdot t^4 \log(t \|E_2^1, \dots, E_2^\tau\|_\infty) \cdot r (2r \|E_1\|_\infty g_1(E_2) + 1)^r \cdot g_1(E_2) \\ &\leq \left(\kappa_2 t^4 \log(t \|E_2^1, \dots, E_2^\tau\|_\infty) \right) (2r \|E_1\|_\infty g_1(E_2))^{r+2}. \end{aligned}$$

Assume for contradiction that there exist $f+2$ indices $1 \leq k_1 < \dots < k_{f+2} \leq f+m$ satisfying (17). By the pigeonhole principle, there is an index k_ℓ such that all the vectors $\mathbf{v}_{k_\ell+1}, \dots, \mathbf{v}_{k_{\ell+1}}$ from the rearrangement (16) correspond to integer vectors $\mathbf{o}_{\pi^{-1}(p)}$ for $p \in [k_\ell+1, k_{\ell+1}]$. We will show that this collection of vectors corresponds to a cycle \mathbf{h} of $\uparrow \mathbf{x}^* - \uparrow \mathbf{z}^*$ which by the minimality of $\|\mathbf{x}^* - \mathbf{z}^*\|_1$ and Lemmas 9 and 14 is impossible. To obtain the cycle, for each $p \in [k_\ell+1, k_{\ell+1}]$, let $i(p), j(p)$, and $\ell(p)$ be such that $\mathbf{o}_{\pi^{-1}(p)} = E_1^{\ell(p)} \mathbf{g}_{j(p)}^{i(p)}$. Initialize $\mathbf{h} := \mathbf{0} \in \mathbb{Z}^{N't}$ and, for each $p \in [k_\ell+1, k_{\ell+1}]$, let $\mathbf{h}^{i(p)} := \mathbf{h}^{i(p)} + \mathbf{g}_{j(p)}^{i(p)}$. Now we check that \mathbf{h} is, in fact, a cycle. First, to see that $E^{(N')}\mathbf{h} = \mathbf{0}$, we have $E_2^\ell \mathbf{h}^i = \mathbf{0}$ for every brick $i \in [N']$ of type ℓ by the fact that \mathbf{h}^i is a sum of $\mathbf{g}_j^i \in \mathcal{G}(E_2^\ell) \subseteq \text{Ker}_{\mathbb{Z}}(E_2^\ell)$, and we have $\bar{E}_1 \mathbf{h} = \mathbf{0}$ by the fact that $\mathbf{t}_{k_\ell} = \mathbf{t}_{k_{\ell+1}}$ and thus $\sum_{p \in [m+f]} E_1^{\ell(p)} \mathbf{g}_{j(p)}^{i(p)} = \mathbf{0}$. Second, $\mathbf{h} \sqsubseteq \mathbf{q}$ because, for every brick $i \in [N']$, \mathbf{h}^i is a sign-compatible sum of elements $\mathbf{g}_j^i \sqsubseteq \mathbf{q}^i$. \square

3.4 Improving the proximity theorem when I has identical columns

In this section we will show how to construct a huge n -fold instance I' from any input instance I such that the number of columns of I' per brick is at most $(2\|E\|_\infty + 1)^{r+s}$,

and in some sense I and I' are equivalent. Specifically, we will show a mapping between the solutions of I and I' which maps integer or configurable optima of I to integer or configurable optima of I' and vice versa, respectively, and such that proximity bounds from I' can be transferred to I . This will eventually allow us to show that even if I has very large t , we can bound the distance between a configurable optimum and some integer optimum of I by a function independent of t .

Construction of I' .

Note that $(2\|E\|_\infty + 1)^{r+s}$ is the number of distinct $(r + s)$ -dimensional integer vectors with entries bounded by $\|E\|_\infty$ in absolute value, hence the number of possible distinct columns per brick. We will show how to “join” variables corresponding to identical columns. Consider any IP with a separable convex objective where columns corresponding to variables x_1 and x_2 are identical. Let f_1 and f_2 be the objective functions corresponding to x_1 and x_2 , and l_1, l_2 and u_1, u_2 be their lower and upper bounds, respectively. Let x_{12} be a new variable which replaces x_1, x_2 in I' . Set the lower bound of x_{12} to be $l_{12} = l_1 + l_2$, upper bound $u_{12} = u_1 + u_2$, and define its objective function as the (min, +)-convolution of f_1 and f_2 :

$$f_{12}(x_{12}) = \min_{\substack{x_1, x_2 \in \mathbb{Z}, x_{12} = x_1 + x_2 \\ (l_1, l_2) \leq (x_1, x_2) \leq (u_1, u_2)}} f_1(x_1) + f_2(x_2) . \tag{18}$$

Note that if f_1 and f_2 are convex, then f_{12} is also convex. Extend f_{12} to fractional values as a linear interpolation, that is, for $x_{12} = \lfloor x_{12} \rfloor + \{x_{12}\}$ fractional, let $f_{12}(x_{12})$ be $f_{12}(\lfloor x_{12} \rfloor) + \{x_{12}\}(f_{12}(\lceil x_{12} \rceil) - f_{12}(\lfloor x_{12} \rfloor))$. The value $f_{12}(x_{12})$ can be obtained by binary search on x_1 (which determines $x_2 = x_{12} - x_1$) in $\mathcal{O}(\log(u_{12} - l_{12}))$ calls to evaluation oracles for f_1 and f_2 . When merging a set S of more than 2 variables, one would compute $f_S(x_S)$ as the solution of the corresponding integer program whose objective is $\sum_{i \in S} f_i(x_i)$ and its constraints are $\sum_{i \in S} x_i = x_S$ and appropriate lower and upper bounds; by [13], this is solvable in time $\text{poly}(|S|) \log(f_{\max}, u_S - l_S)$. However, our goal here is to strengthen our proximity result for I by studying I' , without actually attempting to solve I' .

For a solution \mathbf{x} of I (not necessarily integral), we define $\sigma(\mathbf{x})$ to be a solution of I' where x_1 and x_2 are replaced by $x_{12} = x_1 + x_2$. Clearly, for integer \mathbf{x} , the value of $\sigma(\mathbf{x})$ under the objective of I' is at most the value of \mathbf{x} under f , and if \mathbf{x} is an integer optimum of I , then $\sigma(\mathbf{x})$ will be an integer optimum of I' because we then have $f_{12}(x_{12}) = f_1(x_1) + f_2(x_2)$. We abuse the notation and for an integer \mathbf{x}' define $\sigma^{-1}(\mathbf{x}')$ to be some integral member \mathbf{x} of the set $\sigma^{-1}(\mathbf{x}')$ which satisfies $f_1(x_1) + f_2(x_2) = f_{12}(x'_{12})$. For a configurable solution \mathbf{x}' we define $\sigma^{-1}(\mathbf{x}')$ by taking an \hat{f} -optimal decomposition Γ' of the brick of \mathbf{x}' containing x_{12} and applying σ^{-1} to the configurations in Γ' ; this defines a decomposition Γ and thus a brick $\sum \Gamma$ of a solution \mathbf{x} of I . The next lemma shows that this construction preserves the value of the solution.

Lemma 15 *If \mathbf{x} is an integer optimum of I , then $\sigma(\mathbf{x})$ is an integer optimum of I' , respectively. Similarly, if \mathbf{x} is a configurable optimum of I , then $\sigma(\mathbf{x})$ is a configurable*

optimum of I' . Analogously, if \mathbf{x}' an integer optimum of I' , then $\sigma^{-1}(\mathbf{x}')$ is an integer optimum of I , and if \mathbf{x}' is a configurable optimum of I' , then $\sigma^{-1}(\mathbf{x}')$ is a configurable optimum of I .

Proof It follows from the definition of f_{12} that for any integer solution of I we get an integer solution of I' which is at least as good, and for any integer solution of I' we get an integer solution of I with the same value. For configurable solutions we apply the observation above to each configuration in some \hat{f} -optimal decomposition and use the fact that \hat{f} is defined via f_{12} . \square

This approach generalizes readily to any number of variables. For the sake of simplicity we continue with the example of “joining” two variables whose columns in $E^{(N)}$ are identical.

We are left to argue about proximity. While we believe that it holds in general that any proximity bound between integer and configurable optima of I' transfers to I , we only need this for our specific bound, so we take a less general route.

Lemma 16 *Let \mathbf{x} be a configurable optimum of I with at most $2r$ fractional bricks, $\mathbf{x}' = \sigma(\mathbf{x})$ a configurable optimum of I' , \mathbf{z}' an ℓ_1 -closest integer optimum of I' , and $\mathbf{z} = \sigma^{-1}(\mathbf{z}')$ an integer optimum of I . Let P be the bound of Theorem 2 on $\|\mathbf{x}' - \mathbf{z}'\|_1$. Then $\|\mathbf{x} - \mathbf{z}\|_1 \leq P$.*

Proof Consider the proof of Theorem 2. In it, we create a sequence of vectors v_1, \dots, v_{m+f} . Each of these vectors corresponds to some $E_1^\ell \lambda_j g_j^i$. The crucial observation is that the sequence $(v_i)_i$ obtained from \mathbf{x} , \mathbf{z} is identical to the sequence obtained from \mathbf{x}' , \mathbf{z}' , so if $\|\mathbf{x}' - \mathbf{z}'\|_1 \leq P$, then also $\|\mathbf{x} - \mathbf{z}\|_1 \leq P$. \square

The next corollary is now immediate:

Corollary 1 *Let \mathbf{x}^* be a conf-optimal solution of (HugeCP) with at most $2r$ fractional bricks. Then there is an optimal solution \mathbf{z}^* of (HugeIP) such that*

$$\begin{aligned} \|\mathbf{z}^* - \mathbf{x}^*\|_1 &\leq \left(\kappa_2(r+s)(2\|E\|_\infty + 1)^{4s} \right) (2r\|E_1\|_\infty g_1(E_2))^{6r} \\ &\leq (2\|E\|_\infty + 1)^{\mathcal{O}(s)} (2r\|E\|_\infty g_1(E_2))^{\mathcal{O}(r)} \leq (\|E\|_\infty r s)^{\mathcal{O}(rs)}. \end{aligned}$$

3.5 Algorithm

Recall the statement of the theorem we are proving: **Theorem 1**. Huge N -fold IP with any separable convex objective can be solved in time

$$(\|E\|_\infty r s)^{\mathcal{O}(r^2 s + r s^2)} \text{poly}(\tau, t, \log \|\mathbf{l}, \mathbf{u}, \mathbf{b}, N, f_{\max}\|_\infty).$$

Proof We first give a description of the algorithm which solves huge N -fold IP, then show its correctness, and finally give a time complexity analysis.

Description of the algorithm. First, obtain an optimal solution \mathbf{y} of (ConfLP) and from it a conf-optimal solution $\mathbf{x}^* = \varphi(\mathbf{y})$ with at most $2r$ fractional bricks by Lemma 6.

Applying Corollary 1 to \mathbf{x}^* guarantees the existence of an integer optimum \mathbf{z}^* satisfying

$$\|\mathbf{x}^* - \mathbf{z}^*\|_1 \leq P := \left((r + s)(2\|E\|_\infty + 1)^{4s} \right) (2r\|E_1\|_\infty g_1(E_2))^{6r} . \quad (19)$$

Together with the fact that there are at most $2r$ fractional bricks, this implies that \mathbf{z}^* differs from \mathbf{x}^* in at most $P' = P + 2r$ bricks. The idea of the algorithm is to “fix” the value of the solution on “almost all” bricks and compute the rest using an auxiliary \bar{N} -fold IP problem with a polynomial \bar{N} .

Formally, our goal is to compute an optimal solution \mathbf{z} of (HugeIP) represented succinctly by multiplicities of configurations, or in other words, as a solution ζ of (ConfILP). Denote by $\mathbf{y}_{-P'}$ the vector whose coordinates are defined by setting, for every type $i \in [\tau]$ and every configuration $\mathbf{c} \in \mathcal{C}^i$, $\mathbf{y}_{-P'}(i, \mathbf{c}) = \max\{0, \lfloor y(i, \mathbf{c}) \rfloor - P'\}$. This leaves us with $\|\mathbf{y}\|_1 - \|\mathbf{y}_{-P'}\|_1 \leq |\text{supp}(\mathbf{y})|P' \leq (r + \tau)P' =: \bar{P}$ bricks to determine. Let $\bar{\zeta} = \mathbf{y} - \mathbf{y}_{-P'}$, define $\bar{\mu}$ by setting, for each $i \in [\tau]$, $\bar{\mu}_i := \sum_{\mathbf{c} \in \mathcal{C}^i} \bar{\zeta}(i, \mathbf{c})$, let $\bar{\mathbf{x}} = \varphi(\bar{\zeta})$, and let $\bar{N} = \|\bar{\zeta}\|_1 = \|\bar{\mu}\|_1 \leq \bar{P}$. Construct an auxiliary \bar{N} -fold IP instance with the same blocks $E_1^i, E_2^i, i \in [\tau]$, by, for each brick $\bar{\mathbf{x}}^j$ of type i , setting

$$- \bar{f}^j = f^i, \quad - \bar{\mathbf{b}}^j = \mathbf{b}^i, \quad - \bar{\mathbf{l}}^j = \mathbf{l}^i, \quad - \bar{\mathbf{u}}^j = \mathbf{u}^i.$$

We say that such a brick was *derived from type i* . Lastly, let $\bar{\mathbf{b}}^0 = \mathbf{b}^0 - \sum_{i=1}^\tau \sum_{\mathbf{c} \in \mathcal{C}^i} \zeta(i, \mathbf{c}) E_1^i \mathbf{c}$.

After obtaining an optimal solution $\bar{\mathbf{z}}$ of this instance we update ζ as follows. For each brick $\bar{\mathbf{z}}^j$ derived from type i , increment $\zeta(i, \bar{\mathbf{z}}^j)$ by one.

Correctness. By (19), it is correct to assume that there exists a solution ζ of (ConfILP) which has $\zeta(i, \mathbf{c}) \geq \max\{0, \lfloor y(i, \mathbf{c}) \rfloor - P'\}$ for each $i \in [\tau]$ and $\mathbf{c} \in \mathcal{C}^i$. Thus we may do a variable transformation of (ConfILP) $\zeta = \bar{\zeta} + \mathbf{y}_{-P'}$, obtaining an auxiliary (ConfILP) instance

$$\min \mathbf{v}(\bar{\zeta} + \mathbf{y}_{-P'}) : B(\bar{\zeta} + \mathbf{y}_{-P'}) = \mathbf{d}, \mathbf{0} \leq \bar{\zeta} .$$

The auxiliary huge \bar{N} -fold instance is simply the instance corresponding to the above, and the final construction of ζ corresponds to the described variable transformation.

Complexity. Since $\|\bar{\zeta}\|_1 \leq \bar{P}$, we can obtain an optimal solution $\bar{\mathbf{z}}$ of the auxiliary instance in time $(\|E\|_\infty r s)^{\mathcal{O}(r^2 s + r s^2)} (t \bar{P}) \log(t \bar{P}) \log \|f_{\max}, \bar{\mathbf{b}}, \bar{\mathbf{l}}, \bar{\mathbf{u}}\|_\infty^2$ [13, Corollary 91]. Let us now compute the time needed altogether. To solve (ConfLP), we need time (Lemma 6)

$$\|E\|_\infty^{\mathcal{O}(s^2)} \text{poly}(r t \tau \log \|f_{\max}, \mathbf{l}, \mathbf{u}, \mathbf{b}, \mu\|_\infty) .$$

To solve the auxiliary instance above, we need time

$$(\|E\|_\infty r s)^{\mathcal{O}(r^2 s + r s^2)} (t \bar{P}) \log(t \bar{P}) \log \|f_{\max}, \bar{\mathbf{b}}, \bar{\mathbf{l}}, \bar{\mathbf{u}}\|_\infty^2,$$

where $\bar{P} = (r + \tau)P' \leq \tau(2\|E\|_\infty + 1)^{\mathcal{O}(s)}(2r\|E\|_\infty g_1(E_2))^{\mathcal{O}(r)}$.

Hence we can solve huge N -fold IP in time at most

$$(\|E\|_\infty r s)^{\mathcal{O}(r^2 s + r s^2)} \text{poly}(t \tau \log \|f_{\max}, \mathbf{l}, \mathbf{u}, \mathbf{b}, \boldsymbol{\mu}\|_\infty).$$

□

4 Concluding remarks

At this point one may wonder why bother with the ConfLP rather than solving HugeCP and showing that its optima are close to those of HugeIP. The reason is that even though handling optima of HugeCP is much easier than handling conf-optimal solutions, and even though solving HugeCP is easier than solving ConfLP,⁴ a HugeCP optimum can be very far from a HugeIP optimum [8, Proposition 1]. In other words, ConfLP is a stronger relaxation than HugeCP: consider a brick \mathbf{p} of a HugeCP optimum and a brick \mathbf{q} of a conf-optimal solution; then

$$\mathbf{q} \in \text{conv}\{\mathbf{c} \in \mathbb{Z}^d \mid E_2 \mathbf{c} = \mathbf{0}, \mathbf{l}^i \leq \mathbf{c} \leq \mathbf{u}^i\} \subset \{\mathbf{p}' \in \mathbb{R}^d \mid E_2 \mathbf{p}' = \mathbf{0}, \mathbf{l}^i \leq \mathbf{p}' \leq \mathbf{u}^i\}.$$

In plain language, while \mathbf{q} lies in the integer hull of all configurations, \mathbf{p} only lies in the fractional relaxation of this hull.

Another obstacle is that even though Configuration LP is a standard tool, it is typical that the separation problem is merely approximated rather than solved exactly, leading to approximate solutions of ConfLP. But, we require an exact solution, and so we use a parameterized exact algorithm for IP to solve the separation problem. It is an interesting question when a k -approximate solution of ConfLP, i.e., a solution whose value is at most $k \cdot OPT$, may be used to obtain an $h(k)$ -accurate configurable solution of HugeCP, i.e., a configurable solution which is at ℓ_1 -distance at most $h(k)$ from a configurable optimum. An approximate solution of ConfLP might be much easier to obtain, and yet it may be almost as good as an exact solution for our purposes here.

Another interesting question is a tight complexity bound for the algorithm of Lemma 6. It seems likely that the recent approach of Cslovjecsek et al. [8] could also apply in our high-multiplicity setting, which would yield a near-linear fixed-parameter algorithm. Notice that the iterative augmentation algorithms for standard

⁴ We only outline the reason. We claim that an optimal solution of HugeCP can be obtained in the following way: construct an auxiliary τ -fold CP with bricks $\bar{\mathbf{x}}^1, \dots, \bar{\mathbf{x}}^\tau$ where the i -th brick $\bar{\mathbf{x}}^i$, $i \in [\tau]$, represents the μ^i bricks of type i in the original instance. This is achieved by setting the bounds to $\mu^i \mathbf{l}^i$ and $\mu^i \mathbf{u}^i$, the right hand side to $\mu^i \mathbf{b}^i$, and the objective to $\mu^i f^i(\bar{\mathbf{x}}^i / \mu^i)$. Given an optimum $\bar{\mathbf{x}}$ of this τ -fold CP, we set each brick of type \mathbf{x}^i to the value $\bar{\mathbf{x}}^i / \mu^i$, and claim that \mathbf{x} is an optimum of HugeCP. Thus, while HugeCP can be solved in polynomial time, the separation subproblem needed to solve ConfLP can be NP-hard, in this sense making ConfLP harder.

N -fold IP have a strong combinatorial flavor and use no “black boxes”. Could the ellipsoid method behind Lemma 6 be replaced by a (more) combinatorial algorithm, at least for some important problems which have huge N -fold IP models, such as the scheduling problems studied by Knop et al. [31]?

Funding Open Access funding enabled and organized by Projekt DEAL. D.K. partially supported by OP VVV MEYS project CZ.02.1.01/0.0/0.0/16_019/0000765 “Research Center for Informatics”. M.K. partially supported by Charles University project UNCE/SCI/004, and by project 19-27871X of Grantová agentura České republiky (GA ČR). A.L. partially supported by Israel Science Foundation grant 308/18. M.M. supported by Deutsche Forschungsgemeinschaft (DFG) grant MN 59/4-1. S.O. partially supported by the Dresner chair and Israel Science Foundation grant 308/18.

Availability of data and material (data transparency) Not applicable.

Code Availability Not applicable.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Alon, N., Azar, Y., Woeginger, G.J., Yadid, T.: Approximation schemes for scheduling on parallel machines. *J. Sched.* **1**(1), 55–66 (1998)
2. Altmanová, K., Knop, D., Koutecký, M.: Evaluating and tuning n -fold integer programming. *ACM J. Exp. Algorithmics* **24**(1), 1–22 (2019)
3. Aykanat, C., Pinar, A., Çatalyürek, Ü.V.: Permuting sparse rectangular matrices into block-diagonal form. *SIAM J. Scientific Comput.* **25**(6), 1860–1879 (2004)
4. Bergner, M., Caprara, A., Ceselli, A., Furini, F., Lübbecke, M.E., Malaguti, E., Traversi, E.: Automatic Dantzig-Wolfe reformulation of mixed integer programs. *Math. Prog.* **149**(1–2), 391–424 (2015)
5. Borndörfer, R., Ferreira, C.E., Martin, A.: Decomposing matrices into blocks. *SIAM J. Optim.* **9**(1), 236–269 (1998)
6. Chen, L., Marx, D.: Covering a tree with rooted subtrees—parameterized and approximation algorithms. In *Proc. SODA 2018*, 2801–2820 (2018)
7. Cosmadakis, S.S., Papadimitriou, C.H.: The traveling salesman problem with many visits to few cities. *SIAM J. Comput.* **13**(1), 99–108 (1984)
8. Cslovjecssek, J., Eisenbrand, F., Hunkenschroder, C., Rohwedder, L., Weismantel, R.: Block-structured integer and linear programming in strongly polynomial and near linear time. In: *Proc. SODA 2021*, pp. 1666–1681 (2021)
9. De Loera, J.A., Hemmecke, R., Onn, S., Weismantel, R.: n -fold integer programming. *Discrete Optim.* **5**(2), 231–241 (2008)
10. De Loera, J. A., Hemmecke, R., Köppe, M.: Algebraic and geometric ideas in the theory of discrete optimization, volume 14 of MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Optimization Society, Philadelphia, PA (2013)

11. Eisenbrand, F., Weismantel, R.: Proximity results and faster algorithms for integer programming using the Steinitz lemma. *ACM Trans. Algorithms* **16**(1), 5:1-5:14 (2020)
12. Eisenbrand, F., Hunkenschroder, C., Klein, K.-M.: Faster algorithms for integer programs with block structure. In *Proc. ICALP 2018*, volume 107 of *Leibniz Int. Proc. Informatics*, pp. 49:1–49:13 (2018)
13. Eisenbrand, F., Hunkenschroder, C., Klein, K., Koutecký, M., Levin, A., Onn, S.: An algorithmic theory of integer programming. Technical report (2019). <http://arXiv.org/abs/1904.01361>
14. Fernandez de la Vega, W., Lueker, G.S.: Bin packing can be solved within $1 + \varepsilon$ in linear time. *Combinatorica* **1**(4), 349–355 (1981)
15. Ferris, M.C., Horn, J.D.: Partitioning mathematical programs for parallel solution. *Math. Prog.* **80**(1), 35–61 (1998)
16. Gamrath, G., Lübbecke, M. E.: Experiments with a generic Dantzig–Wolfe decomposition for integer programs. In: *Proc. SEA 2010, Lecture Notes in Computer Science*, vol. 6049, pp. 239–252. Springer (2010)
17. Gilmore, P.C., Gomory, R.E.: A linear programming approach to the cutting-stock problem. *Oper. Res.* **9**, 849–859 (1961)
18. Grötschel, M., Lovász, L., Schrijver, A.: *Geometric algorithms and combinatorial optimization*, vol. 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, second edition (1993)
19. Hemmecke, R., Köppe, M., Weismantel, R.: Graver basis and proximity techniques for block-structured separable convex integer minimization problems. *Math. Prog.* **145**(1–2, Ser. A), 1–18 (2014)
20. Hochbaum, D.S., Shamir, R.: Strongly polynomial algorithms for the high multiplicity scheduling problem. *Oper. Res.* **39**(4), 648–653 (1991)
21. Hochbaum, D.S., Shantikumar, J.G.: Convex separable optimization is not much harder than linear optimization. *J. ACM* **37**(4), 843–862 (1990)
22. Hochbaum, D.S., Shmoys, D.B.: Using dual approximation algorithms for scheduling problems: theoretical and practical results. *J. Assoc. Comput. Mach.* **34**(1), 144–162 (1987)
23. Jansen, K., Rohwedder, L.: On integer programming and convolution. In *Proc. ITCS 2019*, vol. 124 of *Leibniz Int. Proc. Informatics*, pp. 43:1–43:17 (2019)
24. Jansen, K., Solis-Oba, R.: A polynomial time $OPT + 1$ algorithm for the cutting stock problem with a constant number of object lengths. *Math. Oper. Res.* **36**(4), 743–753 (2011)
25. Jansen, K., Klein, K., Maack, M., Rau, M.: Empowering the Configuration-IP — new PTAS results for scheduling with setups times. In *Proc. ITCS 2019*, volume 124 of *Leibniz Int. Proc. Informatics*, pp. 44:1–44:19 (2019)
26. Jansen, K., Lassota, A., Rohwedder, L.: Near-linear time algorithm for n -fold ilps via color coding. *SIAM J. Discret. Math.* **34**(4), 2282–2299 (2020)
27. Karmarkar, N., Karp, R.M.: An efficient approximation scheme for the one-dimensional bin-packing problem. In *Proc. FOCS 1982*, pp. 312–320 (1982)
28. Khaniyev, T., Elhedhli, S., Erenay, F.S.: Structure detection in mixed-integer programs. *INFORMS J. Comput.* **30**(3), 570–587 (2018)
29. Knop, D., Koutecký, M.: Scheduling meets n -fold integer programming. *J. Sched.* **21**(5), 493–503 (2018)
30. Knop, D., Koutecký, M.: Scheduling kernels via configuration LP. Technical report, (2020). [arXiv:2003.02187](https://arxiv.org/abs/2003.02187)
31. Knop, D., Koutecký, M., Levin, A., Mnich, M., Onn, S.: Multitype integer monoid optimization and applications. Technical report (2019) [arXiv:1909.07326](https://arxiv.org/abs/1909.07326)
32. Knop, D., Koutecký, M., Mnich, M.: Combinatorial n -fold integer programming and applications. *Math. Program.* **184**(1), 1–34 (2020)
33. Koutecký, M., Levin, A., Onn, S.: A parameterized strongly polynomial algorithm for block structured integer programs. In *Proc. ICALP 2018*, vol. 107 of *Leibniz Int. Proc. Informatics*, pp. 85:1–85:14 (2018)
34. Onn, S.: *Nonlinear discrete optimization—an algorithmic theory*. Zurich Lectures in Advanced Mathematics. European Mathematical Society (EMS), Zürich (2010)
35. Onn, S.: Huge multiway table problems. *Discrete Optim.* **14**, 72–77 (2014)
36. Onn, S.: Huge tables and multicommodity flows are fixed-parameter tractable via unimodular integer Carathéodory. *J. Comput. Syst. Sci.* **83**(1), 207–214 (2017)
37. Psaraftis, H.N.: A dynamic programming approach for sequencing groups of identical jobs. *Oper. Res.* **28**(6), 1347–1359 (1980)

38. Scheithauer, G., Terno, J.: The modified integer round-up property of the one-dimensional cutting stock problem. *European J. Oper. Res.* **84**(3), 562–571 (1995)
39. Sevast'janov, S., Banaszczyk, W.: To the Steinitz lemma in coordinate form. *Discrete Math.* **169**(1–3), 145–152 (1997)
40. Steinitz, E.: Bedingt konvergente Reihen und konvexe Systeme. *J. Reine Angew. Math.* **146**, 1–52 (1916)
41. van den Akker, J.M., Hoogeveen, J.A., van de Velde, S.L.: Parallel machine scheduling by column generation. *Oper. Res.* **47**(6), 862–872 (1999)
42. Vanderbeck, F., Wolsey, L. A.: Reformulation and decomposition of integer programs. In: *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*, pp. 431–502. Springer (2010)
43. Vose, M.D.: Egyptian fractions. *Bull. London Math. Soc.* **17**(1), 21–24 (1985)
44. Wang, J., Ralphs, T.: Computational experience with hypergraph-based methods for automatic decomposition in discrete optimization. In: *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pp. 394–402. Springer (2013)
45. Weil, R.L., Kettler, P.C.: Rearranging matrices to block-angular form for decomposition (and other) algorithms. *Mgmt. Sci.* **18**(1), 98–108 (1971)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.