Huge multiway table problems
Shmuel Onn
Technion - Israel Institute of Technology, Haifa, Israel

A R T I C L E   I N F O

Article history:
Received 16 May 2014
Accepted 30 July 2014

Keywords:
Integer programming
Multiway table
Graver basis
Integer cone

A B S T R A C T

Optimization over \( l \times m \times n \) integer three-way tables is NP-hard already for fixed \( l = 3 \), but solvable in polynomial time with both \( l, m \) fixed. Here we consider huge tables, where the variable dimension \( n \) is encoded in binary. Combining recent results on Graver bases and recent results on integer cones, we show how to handle such problems in polynomial time. We also show that a harder variant of the problem lies in both NP and coNP. Our treatment goes through the more general class of \( n \)-fold integer programming problems.

1. Introduction

Consider the following optimization problem over three-way tables with line-sums,

\[
\min \left\{ wX : X \in \mathbb{Z}^{l \times m \times n}_+ : \sum_i x_{i,j,k} = e_{j,k}, \sum_j x_{i,j,k} = f_{i,k}, \sum_k x_{i,j,k} = g_{i,j} \right\},
\]

where \( w \in \mathbb{Z}^{l \times m \times n}, e \in \mathbb{Z}^{m \times n}_+, f \in \mathbb{Z}^{l \times n}_+, \) and \( g \in \mathbb{Z}^{l \times m}_+ \). It is NP-hard already for \( l = 3 \), see [1]. But, for fixed \( l, m \) it is solvable in polynomial time [2], and in fact, in time which is cubic in \( n \) and linear in the binary encoding of \( w, e, f, g \), see [3]. Assume throughout then that \( l, m \) are fixed. We call the problem huge if the variable number \( n \) of layers is encoded in binary. In this case, it may not be possible even to write down a single feasible table in polynomial time, let alone solve the above problem. What can be done, then? To describe our results on such huge table problems, let us adjust indexation as follows. We consider each table as a tuple \( x = (x^1, \ldots, x^t) \) consisting of \( n \) many \( l \times m \) layers. Each layer \( x^k \) has a cost matrix \( w^k \in \mathbb{Z}^{l \times m}_+ \), column-sums vector \( e^k \in \mathbb{Z}^m_+ \), and row-sums vector \( f^k \in \mathbb{Z}^l_+ \). Let us first consider table problems which are symmetric with respect to permutations of the layers, that is, where all layers have the same data, namely \( w^k = \tilde{w}, e^k = \tilde{e}, \) and \( f^k = \tilde{f} \) for suitable \( \tilde{w} \in \mathbb{Z}^{l \times m}_+, \tilde{e} \in \mathbb{Z}^m_+, \) and \( \tilde{f} \in \mathbb{Z}^l_+ \). We then have the following.

**Theorem 1.1.** The huge symmetric three-way table problem with fixed \( l, m \) and variable \( n \) can be solved in time which is polynomial in the binary encoding of \( n, w, e, f, g \).

Let us proceed to describe a more general situation. We are now given \( t \) types of layers, where each type \( k \) has its cost matrix \( w^k \in \mathbb{Z}^{l \times m}_+, \) column-sums vector \( e^k \in \mathbb{Z}^m_+, \) and row-sums vector \( f^k \in \mathbb{Z}^l_+ \). Now, in addition, we are given positive integers \( n_1, \ldots, n_t, n \) with \( n_1 + \cdots + n_t = n \), all encoded in binary. A feasible table \( x = (x^1, \ldots, x^t) \) now must have first \( n_1 \) layers of type 1, next \( n_2 \) layers of type 2, and so on, with last \( n_t \) layers of type \( t \). Note that the symmetric case discussed above occurs as the special case of \( t = 1 \), and the standard (non-huge) table problem occurs as the special case of \( t = n \) and \( n_1 = \cdots = n_t = 1 \).

E-mail address: onn@ie.technion.ac.il.

http://dx.doi.org/10.1016/j.disopt.2014.07.003
1572-5286/© 2014 Published by Elsevier B.V.
We can now define the huge table problem formally as follows.

**Huge table problem.** Given \( t \) types, costs \( w^k \in \mathbb{Z}^{t \times m} \), column-sums \( e^k \in \mathbb{Z}^m_+ \), and row-sums \( f^k \in \mathbb{Z}^n_+ \) for \( k = 1, \ldots, t \), line-sums \( g \in \mathbb{Z}^{t \times m}_+ \), and positive integers \( n_1, \ldots, n_t \) with \( n_1 + \cdots + n_t = n \), find an optimal table or assert that none exists.

Note that while \( t \) may be small, the set of possible layers of type \( k \) is
\[
\left\{ x^k \in \mathbb{Z}_+^{t \times m} : \sum_i x^k_{i,j} = e^k_j, \sum_j x^k_{i,j} = f^k_i \right\},
\]
and may have cardinality exponential in the binary encoding of \( e^k, f^k \), so again it is not off hand clear how to even write down a single table. But we have the following.

**Theorem 1.2.** Consider the huge three-way table problem with all data but \( t \) encoded in binary. For fixed \( t \) it can be solved in polynomial time. For \( t \) variable and encoded in unary, we can test in polynomial time if a given table is optimal and if not obtain a better one, and the feasibility problem is in both complexity classes \( \text{NP} \) and \( \text{coNP} \).

Since problems in both \( \text{NP} \) and \( \text{coNP} \) are typically in \( \text{P} \), it is particularly interesting whether the feasibility problem with variable \( t \) is indeed solvable in polynomial time.

These results follow from broader results which we proceed to describe. The class of \( n \)-fold integer programming problems is defined as follows. Let \( A \) be an \( (r, s) \times d \) bimatrix, by which we mean a matrix having an \( r \times d \) block \( A_1 \) and \( s \times d \) block \( A_2 \),
\[
A = \begin{pmatrix} A_1 & A_2 \\ A_1 & A_2 \end{pmatrix}.
\]
Its \( n \)-fold product is the following \((r + sn) \times (dn)\) matrix,
\[
A^{(n)} := \begin{pmatrix} A_1 & A_1 & \cdots & A_1 \\ A_2 & 0 & \cdots & 0 \\ 0 & A_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_2 \end{pmatrix}.
\]
The \( n \)-integer programming problem is then the following.
\[
\min \left\{ wx : x \in \mathbb{Z}^{dn}, A^{(n)} x = b, 1 \leq x \leq u \right\},
\]
where \( w \in \mathbb{Z}^{dn}, b \in \mathbb{Z}^{r + sm}, \) and \( l, u \in \mathbb{Z}^{dn} \) with \( \mathbb{Z}_n := \mathbb{Z} \cup \{ \pm \infty \} \). For example, multiway table problems of any dimension are \( n \)-fold programs, as explained before.

It was shown in [2,4], building on [5–7], that \( n \)-fold integer programming for fixed bimatrix \( A \) can be solved in polynomial time. More recently, it was shown that for fixed \( A \) it can be solved in time which is cubic in \( n \) and linear in the binary encoding of \( w, b, l, u, \) and that if only the dimensions \( r, s, d \) of \( A \) are fixed but \( A \) is part of the input, then it can be solved in time cubic in \( n \), polynomial in the unary encoding of \( A \), and linear in the binary encoding of \( w, b, l, u \). See [8] for a detailed treatment of the theory and applications of \( n \)-fold integer programming.

The vector ingredients of an \( n \)-fold integer program are naturally arranged in bricks, where \( w = (w^1, \ldots, w^n) \) with \( w^k \in \mathbb{Z}^d \) for \( k = 1, \ldots, n \), and likewise for \( l, u, b \), with \( b = (b^0, b^1, \ldots, b^d) \) with \( b^0 \in \mathbb{Z}^d \) and \( b^k \in \mathbb{Z}^d \) for \( k = 1, \ldots, n \). Call an \( n \)-fold integer program huge if \( n \) is encoded in binary. More precisely, we are now given \( t \) types of bricks, where each type \( k = 1, \ldots, t \) has its cost \( w^k \in \mathbb{Z}^d \), lower and upper bounds \( l^k, u^k \in \mathbb{Z}^d \), and right-hand side \( b^k \in \mathbb{Z}^d \). Also given are \( b^0 \in \mathbb{Z}^d \) and positive integers \( n_1, \ldots, n_t \) with \( n_1 + \cdots + n_t = n \), all encoded in binary. A feasible point \( x = (x^1, \ldots, x^n) \) now must have first \( n_1 \) bricks of type 1, next \( n_2 \) bricks of type 2, and so on, with last \( n_t \) bricks of type \( t \). Standard \( n \)-fold integer programming occurs as the special case of \( t = n \) and \( n_1 = \cdots = n_t = 1 \), and symmetric \( n \)-fold integer programming occurs as the special case of \( t = 1 \).

For \( k = 1, \ldots, t \), the set of all possible bricks of type \( k \) is the following.
\[
S^k := \{ z \in \mathbb{Z}^d : A_2 z = b^k, l^k \leq z \leq u^k \}.
\]
We assume for simplicity that \( S^k \) is finite for all \( k \), which is the case in most applications, such as in multiway table problems. Let \( \lambda^k := (\lambda^k_z : z \in S^k) \) be a nonnegative integer tuple with entries indexed by points of \( S^k \). Each feasible point \( x = (x^1, \ldots, x^n) \) gives rise to \( \lambda^1, \ldots, \lambda^t \) satisfying \( \sum \lambda^k_z = n_i \), where \( \lambda^k_z \) is the number of bricks of \( x \) of type \( k \) which are equal to \( z \). Let the support of \( \lambda^k \) be \( \text{supp}(\lambda^k) := \{ z \in S^k : \lambda^k_z \neq 0 \} \). Then a compact presentation of \( x \) consists of the restrictions of \( \lambda^k \) to \( \text{supp}(\lambda^k) \) for all \( k \). However, the cardinality of \( S^k \) may be exponential in the binary encoding of the data \( b^k, l^k, u^k \), so off hand this presentation might be exponential as well. Nonetheless, we show the following.
Theorem 1.3. Consider data for the huge n-fold integer programming problem with t types over \((r, s) \times d\) bimatrix \(A\), with \(r, s, d\) fixed, and with \(u^k, b^k, \ell^k, u^k, n_1, \ldots, n_t, n\) all encoded in binary. Then the following three statements hold:

1. If the problem is feasible, then there is an optimal solution which admits a compact presentation \(\lambda^1, \ldots, \lambda^t\) satisfying \(|\text{supp}(\lambda^k)| \leq 2^d\) for \(k = 1, \ldots, t\).
2. For \(t\) fixed, the problem can be solved in polynomial time even if the bimatrix \(A\) is a variable part of the input and encoded in binary.
3. For \(A\) fixed and \(t\) variable and encoded in unary, the augmentation problem can be solved in polynomial time, namely, given a feasible point presented compactly, we can either assert that it is optimal or find a better feasible point.

We proceed as follows. In Section 2 we prove the above theorems. In Section 3 we discuss extensions to tables of any dimension and quad n-fold integer programming.

2. Proofs

We begin by proving the three parts of Theorem 1.3 one by one. First, note that point \(x = (x^1, \ldots, x^n)\) is feasible in the huge n-fold integer program only if each brick \(x^i\) lies in some \(S^k = \{z \in \mathbb{Z}^d : A_d z = b^k, \ell^k \leq z \leq u^k\}\), and \(A_1 \sum_{i=1}^n x^i = b^k\).

So our assumption that each \(S^k\) is finite implies that the set of feasible points is finite as well. Therefore, if the program is feasible, then it has an optimal solution.

The proof of part (1) makes use of a nice argument of Eisenbrand–Shmonin [9].

Proof of Theorem 1.3 part (1). Suppose the huge n-fold program is feasible. Then, as explained above, there is an optimal solution. Let \(x\) be an optimal solution with minimum value \(\sum_{i=1}^n \|x^i\|^2\) with \(\|z\|^2 = \sum_{j=1}^d z_{j}^2\) the Euclidean norm squared. Let \(\lambda^1, \ldots, \lambda^t\) be a compact presentation of \(x\). Suppose indirectly that we have \(|\text{supp}(\lambda^j)| > 2^d\). Then there are two vectors \(y' \neq y''\) in \(\text{supp}(\lambda^j)\) having the same parity on each coordinate, implying \(\bar{y} := \frac{1}{2}(y' + y'') \in S^j\). For \(k = 1, \ldots, t\) define \(\mu^k\) on \(S^k\) to be the same as \(\lambda^k\) except that \(\mu^j_{y'} := \lambda^j_{y'} - 1, \mu^j_{y''} := \lambda^j_{y''} - 1\) and \(\mu^j := \lambda^j + 2\). Let \(\bar{x}\) be the vector whose compact presentation is given by \(\mu^k\). Then

\[
\sum_{i=1}^n \|x^i\|^2 - \sum_{i=1}^n \|\bar{x}^i\|^2 = \sum_{k=1}^t \sum_{z \in S^k} (\lambda^k_z - \mu^k_z)z^2 = y^2 + y'' - 2y = 0
\]

and therefore \(A_1 \sum_{i=1}^n \bar{x}^i = A_1 \sum_{i=1}^n x^i = b^0\) so \(\bar{x}\) is also feasible. Furthermore,

\[
w x - w \bar{x} = \sum_{k=1}^t w_k \sum_{z \in S^k} (\lambda^k_z - \mu^k_z)z = w^j(y' + y'' - 2y) = 0
\]

and therefore \(\bar{x}\) is also optimal. But now we have

\[
\sum_{i=1}^n \|x^i\|^2 - \sum_{i=1}^n \|\bar{x}^i\|^2 = \sum_{k=1}^t \sum_{z \in S^k} (\lambda^k_z - \mu^k_z)z^2 = \|y'\|^2 + \|y''\|^2 - 2\|y\|^2
\]

\[
= \|y'\|^2 + \|y''\|^2 - 2 \left\| \frac{1}{2} (y' + y'') \right\|^2 = \frac{1}{2} \|y' - y''\|^2 > 0
\]

which is a contradiction to the choice of \(x\). This completes the proof. 

The proof of part (2) uses the following beautiful result of [10] building on [9].

Proposition 2.1 (Goemans–Rothvoss). Fix \(d, t\). Let \(S^k = \{z \in \mathbb{Z}^d : A_d z \leq a^k\}\) be finite for \(k = 1, \ldots, t\), and let \(T = \{z \in \mathbb{Z}^d : B z \leq b\}\). Then, in polynomial time, we can decide if there are nonnegative integer tuples \(\lambda^k = (\lambda^k_{z} : z \in S^k)\) such that \(\sum_{k=1}^t \sum_{z \in S^k} \lambda^k_z z \in T\), in which case we can compute such \(\lambda^k\) of polynomial support.

Proof of Theorem 1.3 part (2). We make use of points in \(z \in \mathbb{Z}^{(d+1)}\) and index each such point by \(z = (z_0^1, z_1^1, \ldots, z_0^t, z_1^t)\) with \(z_0^k \in \mathbb{Z}\) and \(z_i^k \in \mathbb{Z}^2\) for \(k = 1, \ldots, t\). Let \(L \leq U\) be two integers. Define the following sets \(S^1, \ldots, S^t\) and \(T\) in \(\mathbb{Z}^{(d+1)}\),

\[S^k := \{z \in \mathbb{Z}^{(d+1)} : z_0^k = 1, A_2 z^k = b^k, \ell^k \leq z^k \leq u^k, z_0^k = 0, z_i^k = 0, i \neq k\},\]

\[T := \{y \in \mathbb{Z}^{(d+1)} : y_1^0 = n_1, \ldots, y_1^t = n_t, A_1 \sum_{k=1}^t y^k = b^0, L \leq \sum_{k=1}^t w^k y^k \leq U\} .\]
Now suppose that $x = (x^1, \ldots, x^n)$ is a feasible point in the huge $n$-fold integer program, with objective function value $wx = v$ which satisfies $L \leq v \leq U$. Note that $\{z^k \in \mathbb{Z}^d : z \in S^k\}$ is the set of possible bricks of $x$ of type $k$, and let $\lambda_k = (\lambda^k_z : z \in S^k)$ for $k = 1, \ldots, t$ be nonnegative integer tuples with $\lambda^k_z$ the number of bricks of $x$ of type $k$ which is equal to $z^k$. Let $y := \sum_{i=1}^t \sum_{z \in S^i} \lambda^i_z z \in \mathbb{Z}^{d(t+1)}$.

Since $x$ is feasible, we have

$$y^k_0 = \sum_{i=1}^t \sum_{z \in S^i} \lambda^i_z z^k_0 = \sum_{z \in S^k} \lambda^k_z = n_k, \quad k = 1, \ldots, t,$$

$$A_1 \sum_{k=1}^t y^k = A_1 \sum_{k=1}^t \sum_{i=1}^t \sum_{z \in S^i} \lambda^i_z z^k = A_1 \sum_{k=1}^t \sum_{z \in S^k} \lambda^k_z z^k = A_1 \sum_{j=1}^n x^j = b^0,$$

and

$$\sum_{k=1}^t w^k y^k = \sum_{k=1}^t w^k \sum_{z \in S^k} \lambda^k_z z^k = \sum_{k=1}^t w^k \sum \{x^j : x^j \text{ has type } k\} = wx = v.$$

So $y$ is a nonnegative integer combination of points of $\bigcup_{k=1}^t S^k$ which lies in $T$.

Conversely, suppose $y = \sum_{i=1}^t \sum_{z \in S^i} \lambda^i_z z$ is a nonnegative integer combination of points of $\bigcup_{k=1}^t S^k$ and $y \in T$, and let $v := \sum_{k=1}^t w^k y^k$. Then

$$\sum_{z \in S^k} \lambda^k_z = \sum_{i=1}^t \sum_{z \in S^i} \lambda^i_z z^k = y^k_0 = n_k, \quad k = 1, \ldots, t,$$

so we can construct a vector $x = (x^1, \ldots, x^n)$ with $\lambda^k_z$ bricks of type $k$ which are equal to $z^k$ for $k = 1, \ldots, t$ and all $z \in S^k$. We then have

$$A_1 \sum_{j=1}^n x^j = A_1 \sum_{k=1}^t \sum_{z \in S^k} \lambda^k_z z^k = A_1 \sum_{i=1}^t \sum_{z \in S^i} \lambda^i_z z^k = A_1 \sum_{k=1}^t y^k = b^0,$$

so $x$ is feasible in the huge $n$-fold program, and has objective function value

$$wx = \sum_{k=1}^t w^k \sum \{x^j : x^j \text{ has type } k\} = \sum_{k=1}^t w^k \sum_{z \in S^k} \lambda^k_z z^k = \sum_{k=1}^t w^k y^k = v.$$
\[ h = (h^1, \ldots, h^k) \in g(A^g). \] Let \( \bigcup_{k=1}^{t} \text{supp}(\lambda^k) \) be the disjoint union of the supports of the \( \lambda^k \) (so a point which happens to be in the support of more than one \( \lambda^k \) appears more than once). Consider a mapping
\[ \phi: \{h^1, \ldots, h^k\} \rightarrow \bigcup_{k=1}^{t} \text{supp}(\lambda^k): h^i \mapsto z^i \in \text{supp}(\lambda^{k(i)}). \]

Such a mapping provides a compact way of prescribing an \( n \)-lifting of \( h \). For such a lifting and any \( \alpha \in \mathbb{Z}_+ \), we will have that \( x + \alpha y \) is feasible and better than \( x \) if the following conditions hold: (1) \( |\phi^{-1}(z)| \leq \lambda^k \) for \( k = 1, \ldots, t \) and all \( z \in \text{supp}(\lambda^k) \); (2) \( k^{(i)} \leq z^i + \alpha h^i \leq u^{(i)} \) for \( i = 1, \ldots, g \); (3) \( \sum_{k=1}^{t} u^{(k)} h^i \leq 0 \). (Note that each \( h^i \) satisfies \( A^g h^i = 0 \) and hence \( A_2(z^i + \alpha h^i) = A^g z^i = b^{(i)} \) holds automatically.) Now, it can be checked if these conditions hold, say, with \( \alpha = 1 \), and if they do, the maximum \( \alpha \) for which \( h \) can be computed, easily in polynomial time. Moreover, a compact presentation \( \mu^1, \ldots, \mu^t \) of the new better point \( x + \alpha y \) can be obtained as follows. Begin by defining \( \mu^{k} := \lambda^k \) for \( k = 1, \ldots, t \). Now, for \( i = 1, \ldots, g \), set
\[ \mu^{k(i)} + \alpha h^i := \mu^{k(i)} + \alpha h^i + 1. \]

This provides a compact presentation of the new feasible and better point \( x + \alpha y \).

Now, since the bimatrix \( A^g \) is fixed, so is its Graver complexity \( g = g(A) \) and hence so is the number of elements \( h \in g(A^g) \). Moreover, the number of possible lifting mappings \( \phi \) of \( h \) is \( |\bigcup_{k=1}^{t} \text{supp}(\lambda^k)|^g \) which is polynomial in the size of the input which includes the compact presentation \( \lambda^1, \ldots, \lambda^t \) of \( x \). So by going over all \( h \in g(A^g) \) and \( \phi \) we can either find that there is no feasible better point of the form \( x + \alpha \) and conclude that \( x \) is optimal, or find \( h \in g(A^g) \) mapping \( \phi, \alpha \in \mathbb{Z}_+ \), and compact presentation \( \mu^1, \ldots, \mu^t \) of that \( x + \alpha y \) which gives best improvement. \( \square \)

We proceed to establish Theorems 1.1 and 1.2.

**Proof of Theorems 1.1 and 1.2.** The huge three-way table problem can be formulated as a huge \( n \)-fold integer programming problem as follows. Let \( t = d = lm \) and \( s = l + m \), and let \( A_1 = lm \) be the \( lm \times lm \) identity matrix and \( A_2 \) be the \( (l + m) \times lm \) incidence matrix of the complete bipartite graph \( K_{lm} \). (So \( A_2 \) is itself an \( m \)-fold product \( A = B(m) \) with the bimatrix having \( B_1 = I_l \) and \( B_2 = \{I| \} \) of \( l \) ones.) Index \( i \times m \times n \) tables as \( x = (x^1, \ldots, x^n) \) with \( x^k = (x_{11}, \ldots, x_{1m}, \ldots, x_{lm}, \ldots, x_{nm}) \), let the cost \( u^k \) be as given, set all lower bounds \( k_{ij} := 0 \) and all upper bounds \( k_{ij} := \infty \), and arrange the row-sums, column-sums, and line-sums suitably in the right-hand side vector \( b \), with \( b^0 = g \) and \( b^k = (\phi, e^k) \) for \( k = 1, \ldots, t \). This encodes the huge table problem as a huge \( n \)-fold integer program with a fixed bimatrix \( A \).

**Theorem 1.1** now follows from Theorem 1.3 part (2) with \( t = 1 \).

The statements of Theorem 1.2 regarding the optimization problem with fixed and variable \( t \) now follow from Theorem 1.3 part (2) and part (3) respectively.

We proceed to prove the last statement of Theorem 1.2, asserting that the feasibility problem with variable number \( t \) of types is in both NP and coNP. First note that if the problem is feasible then, by Theorem 1.2 part (1) it has an optimal solution (say, with respect to the identically 0 objective function) with compact presentation \( \lambda^1, \ldots, \lambda^t \) satisfying \( |\text{supp}(\lambda^k)| \leq 2^{ln} \) for \( k = 1, \ldots, t \). This presentation provides a polynomial certificate for feasibility, showing the problem is in NP.

We proceed to show that the problem is in coNP. Let \( c := 2r + s = 2lm + (l + m) \) and define an \( (r, s) \times c \) bimatrix \( C \) with blocks \( C_1 := [A_1, lmm, \theta_{lm 	imes (l + m)}] \) and \( C_2 := [A_2, \theta_{((l + m) 	imes lm), l + m}] \) with \( A_1, A_2 \) as above. Define a huge \( n \)-fold program over the bimatrix \( C \) as follows. As before, set all lower bounds to 0 and all upper bounds to \( \infty \). Without loss of generality assume that all row-sums, column-sums, and line-sums are nonnegative, else there is no feasible table, and arrange them in the right-hand side vector \( b \). Now index the variables as \( v = (v^1, \ldots, v^0) \) with each brick of the form \( v^i = (x^1, y^1, z^1) \) with \( x^1, y^1 \in Z_{lm}^+ \) and \( z^i \in Z_{lm}^+ \), with \( x^1, y^1 \) as a layer of the sought table and \( y^i, z^i \) as slacks. There are again \( t \) types where a brick \( v \) of each type \( k = 1, \ldots, t \) must satisfy \( C_2 v = b^k \), and the multiplicities are \( n_1, \ldots, n_t \) with \( n_1 + \cdots + n_t = n \) as given for the table problem.

Now, for this auxiliary program we can always write down a compact presentation of a feasible point \( v \) defined as follows. We use the brick \( a^0 := [0, b^0, b^1] \) of type 1 with multiplicity \( \lambda^1_{a^0} := 1 \), the brick \( a^t := [0, b^0, b^1] \) of type 1 with multiplicity \( k_1 := n_1 - 1 \), and for \( k = 2, \ldots, t \), the brick \( a^k := [0, b^0, b^1] \) of type \( k \) with multiplicity \( \lambda^k_{a^k} := n_k \). Clearly, these bricks are nonnegative and each brick of type \( k \) indeed satisfies \( C_2 v = b^k \). Moreover, \( C_1 \sum_{i=1}^{n} v^i = C_1 \left( a^0 + \sum_{k=1}^{t} \lambda^k_{a^k} a^k \right) = b^0 \).

Now, we consider the problem of minimizing the sum of all slack variables. Note that the value of this sum will be always nonnegative, and will be 0 if and only if all slacks are 0, which holds if and only if the restriction \( x := (x^1, \ldots, x^n) \) of \( v \) is a feasible table, which holds if and only if the table problem is feasible.

Suppose now the table problem is infeasible. By Theorem 1.2 part (1), the auxiliary program has an optimal solution, i.e. \( v \) minimizing the sum of slack variables, with compact presentation \( \lambda^k \) satisfying \( |\text{supp}(\lambda^k)| \leq 2^{2lm + l + m} \) for \( k = 1, \ldots, t \). Now, using this compact presentation, we can compute the sum of slacks and verify that it is positive, and using Theorem 1.2 part (3), we can verify that \( v \) is indeed an optimal solution, in polynomial time. This proves that the problem is in coNP. \( \square \)
3. Discussion

It was shown in [1] that every bounded integer program can be isomorphically represented in polynomial time as some $3 \times m \times n$ table problems for some $m$ and $n$. By the above results, for any fixed $m$ we can handle integer programs with huge $n$.

The results on three-way tables with line-sums can be extended to tables of arbitrary fixed dimension and margins of any dimension. (A $k$-margin of a $d$-way table is the sum of entries in some $(d-k)$-way subtable.) We have the following.

**Theorem 3.1.** Consider the huge multiway table problem over $m_1 \times \cdots \times m_k \times n$ tables with $t$ types, with given margins of any dimension, with $k, m_1, \ldots, m_k$ fixed. For $t$ fixed, the optimization problem is solvable in polynomial time, and for $t$ variable and encoded in unary, the augmentation problem is solvable in polynomial time.

The quad $n$-fold integer programming problem was introduced in [12] as a common generalization of $n$-fold integer programming and stochastic integer programming. The problem is defined as follows. Let $A$ be an $(r, s) \times (c, d)$ quadmatrix, by which we mean a matrix having the following block structure,

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix}. $$

Its $n$-fold product is the following $(r + sn) \times (c + dn)$ matrix,

$$A^{(n)} := \begin{pmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,2} \\ A_{2,1} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_{2,1} & 0 & \cdots & A_{2,2} \end{pmatrix}. $$

The quad $n$-fold integer programming problem is then the following,

$$\min \{ wx : x \in \mathbb{Z}^{c+dn}, A^{(n)}x = b, l \leq x \leq u \}, $$

where $w \in \mathbb{Z}^{c+dn}$, $b \in \mathbb{Z}^{r+sn}$, and $l, u \in \mathbb{Z}^{c+dn}$ with $\mathbb{Z}_\infty := \mathbb{Z} \cup \{ \pm \infty \}$. Stochastic integer programming arises with empty blocks $A_{1,1}, A_{1,2}$, and $n$-fold integer programming arises with empty blocks $A_{1,1}, A_{2,1}$. Extending the polynomial solvability of $n$-fold programming [2] and stochastic integer programming [4], it was shown in [12] that for the fixed quadmatrix this problem can be solved in polynomial time as well.

A huge version of this problem can be defined as before. Here a compact presentation of $x = (x^0, x^1, \ldots, x^n)$ consists of $x^0 \in \mathbb{Z}^c$ and the restrictions of $\lambda^k$ to $\text{supp}(\lambda^k)$ for $k = 1, \ldots, t$. **Theorem 1.3** part (1) can be easily extended as follows.

**Theorem 3.2.** Consider the huge quad $n$-fold integer programming problem with $t$ types over an $(r, s) \times (c, d)$ quadmatrix. If it is feasible, then it has an optimal solution with compact presentation $\lambda^1, \ldots, \lambda^t$ satisfying $|\text{supp}(\lambda^k)| \leq 2^d$ for $k = 1, \ldots, t$.

References


