RECONSTRUCTING BINARY MATRICES UNDER WINDOW CONSTRAINTS FROM THEIR ROW AND COLUMN SUMS

ANDREAS ALPERS AND PETER GRITZMANN

Abstract. The present paper deals with the discrete inverse problem of reconstructing binary matrices from their row and column sums under additional constraints on the number and pattern of entries in specified minors. While the classical consistency and reconstruction problems for two directions in discrete tomography can be solved in polynomial time, it turns out that these window constraints cause various unexpected complexity jumps back and forth from polynomial-time solvability to $\mathbb{NP}$-hardness.

1. Introduction

The problem of reconstructing binary matrices from their row and column sums is a classical inverse task in combinatorics; see [17]. Even though the term was introduced much later it can be seen as a root of *discrete tomography*; see [19, 16, 11, 12, 1], and the collected editions [14], [15]. Of particular relevance for the present paper are two well-known observations: (1) The question of consistency of the data i.e., the question whether there exists a matrix whose row and column sums coincide with the given data, can be solved in polynomial time. (2) Typically, its row and column sums do not determine the underlying matrix uniquely. See [17, 8, 2, 13] for characterizations of the 'rare' cases of uniqueness.

In the present paper we address the second issue by adding additional window constraints, specifying (or giving bounds on) how many points there are in certain minors of the matrix. These constraints come up naturally in *dynamic discrete tomography* [3]. An application of particular relevance in physics is that of particle tracking; see [5], [20]. Here, the positions of particles over time are to be reconstructed from two (sometimes more) high speed camera images. Window constraints are a natural way of modeling additional physical information for instance on the speed of the particles; see [3] and the background literature given there for further information.

Here we are taking a complexity theoretical view commemorating Observation (1). In fact, when adding various kinds of window constraints, we are interested in the boundary between polynomial-time solvability and $\mathbb{NP}$-hardness. Intuitively speaking, we ask which kinds of additional constraints can be added without imposing a significant extra computational effort. Or, phrased differently, what is the computational price to pay for reducing the number of solutions by utilizing additional window information. We will focus on the effect of three different parameters: k corresponds to the *size* of the window, ν is the *number of* 1*'s* in the nonzero minors, and t specifies the allowed positions of 1's in the windows, referred to as the *pattern* of the window. The choice for these parameters will specify the given problem $\textsc{Rec}(k, \nu, t)$, which is formally introduced in Section 2. Hence k, ν, and t are given beforehand (i.e., are not part of the input). As it will turn out, the problem exhibits various unexpected complexity jumps.

Omitting technical details (which are all given in Section 2) some of these jumps can be summarized as follows; see Table 1.

For $k = 1$ the problems are in $\mathbb{P}$ regardless on how the other parameters are set; see Theorem 1(i). For $k \geq 2$ it depends on ν and t whether the problems are in $\mathbb{P}$ or $\mathbb{NP}$-hard; see Theorems 1(ii),(iii)

Zentrum Mathematik, Technische Universität München, D-85747 Garching bei München, Germany
E-mail address: alpers@ma.tum.de, gritzman@tum.de.

The authors gratefully acknowledge support through the German Research Foundation Grant GR 993/10-2 and the European COST Network MP1207.

and 2. For $k \geq 2$, some values of ν render the problems tractable while others make them $\mathbb{NP}$-hard; see Theorems 1(ii) and 2(ii) even if the patterns are not restricted at all. Adding a pattern constraint may turn an otherwise $\mathbb{NP}$-hard problem into a polynomial time solvable problem; see Theorems 2(ii) and 1(iii). The reverse complexity jump, however, can also be observed; see Theorems 1(ii) and 2(i).

The present paper is organized as follows: Section 2 will introduce our notation and state the main results. The proofs of our tractability results are given in Section 3 while the $\mathbb{NP}$-hardness results are proved in Section 4. Section 5 contains some final remarks. In particular, it provides a (potentially also quite interesting) extension of our main problem and collects related implications of our main results.

2. Notation and Main Results

We begin with some standard notation.

Let $\mathbb{Z}$, $\mathbb{N}$, and $\mathbb{N}_0$ denote the set of integers, natural numbers, and non-negative integers, respectively. For $k \in \mathbb{N}$ set $k\mathbb{N}_0 := \{ki : i \in \mathbb{N}_0\}$, $k\mathbb{N}_0 + 1 := \{ki + 1 : i \in \mathbb{N}_0\}$, $[k] := \{1, \dots, k\}$, and $[k]_0 := [k] \cup \{0\}$. The support of a vector $x := (\xi_1, \dots, \xi_d)^T \in \mathbb{Z}^d$ is defined as $\operatorname{supp}(x) := \{i : \xi_i \neq 0\}$. With $\mathbb{1}$ we denote the all-ones vector of the corresponding dimension. The cardinality of a finite set $F \subseteq \mathbb{Z}^d$ is denoted by $|F|$. We will use the notational convention that specific settings of variables and parameters are signified by a superscript *.

In this paper we often refer to points (or variables) $\xi_{p,q}$ where (p, q) are points of the integer grid $\mathbb{Z}^2$. Then, p and q denote the x- and y-coordinates of (p, q), respectively. In the grid $[m] \times [n]$, the set $\{i\} \times [n]$ and $[m] \times \{j\}$ is called *column* i and *row* j, respectively. For $k \in \mathbb{N}$ we refer to $\{(i-1)k+1, \dots, ik\} \times [n]$ and $[m] \times \{(j-1)k+1, \dots, jk\}$ as *vertical strip* i *(of width* k*)* and *horizontal strip* j *(of width* k*)*, respectively. Of course, the standard matrix notation can be obtained from our notation by the coordinate transformation

$$\begin{pmatrix} \xi_{1,n} & \cdots & \xi_{m,n} \\ \vdots & & \vdots \\ \xi_{1,1} & \cdots & \xi_{m,1} \end{pmatrix} \mapsto \begin{pmatrix} \xi_{1,1} & \cdots & \xi_{1,n} \\ \vdots & & \vdots \\ \xi_{m,1} & \cdots & \xi_{m,n} \end{pmatrix}.$$

Generally, any subset of the $[m] \times [n]$ grid is called a *window*. Windows of the form $([a, b] \times [c, d]) \cap \mathbb{Z}^2$, with $a, b, c, d \in \mathbb{Z}$ and $a \leq b$, $c \leq d$, are called *boxes*. Defining for any $k \in \mathbb{N}$ and $m, n \in k\mathbb{N}$ the set of *(lower-left) corner points* $C(m, n, k) := ([m] \times [n]) \cap (k\mathbb{N}_0 + 1)^2$, we call any box $B_k(i, j) := (i, j) + [k-1]^2$ with $(i, j) \in C(m, n, k)$ a *block*; see Figure 1 for an illustration. (Here we depict the structure both as point set and as pixels. In the following we restrict the figures to pixel images, which we find more intuitive.) The blocks form a partition of $[m] \times [n]$, i.e., $\bigcup_{(i,j) \in C(m,n,k)} B_k(i, j) = [m] \times [n]$. In the main part of this paper, we consider such non-overlapping blocks, which play also a role in super-resolution imaging [4]. In Section 5 we consider also other windows, which may be positioned at other places than those defined by $C(m, n, k)$.

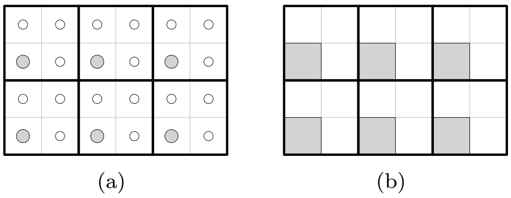


Figure 1. (a) Point set, (b) the same structure depicted as pixels showing the six blocks $B_k(i, j)$, $(i, j) \in C(m, n, k)$ for $m = 6$, $n = 4$, and $k = 2$. The elements of $C(m, n, k)$ are highlighted.

Next we introduce three patterns that we will study in detail since they exhibit already the general complexity jump behavior we are particular interested in.

The first pattern $P(k,0)$ is unconstrained, i.e., does not pose any additional restrictions on the positions of 1's. The second pattern $P(k,1)$ forces all elements in the $k \times k$ block to be 0 except possibly for the two entries in the lower-left and upper-right corner. The third pattern $P(k,2)$ excludes all patterns that admit more than one 1 in each row of the $k \times k$ block (see also Figure 2). Here are the formal definitions.

For $k \in \mathbb{N}$ let $2^{[k-1]_0^2}$ denote the power set of $[k-1]_0^2$. Then we set

$$\begin{aligned} P(k,0) &:= 2^{[k-1]_0^2}, \\ P(k,1) &:= \{\{(0,0)\},\{(k-1,k-1)\}\}, \\ P(k,2) &:= \{M \in 2^{[k-1]_0^2} \ : \ |M \cap ([k-1]_0 \times \{j\})\,| \le 1 \text{ for all } j \in [k]_0.\} \end{aligned}$$

Further, for $(i,j) \in C(m,n,2)$ and $x = (\xi_{i,j})_{i \in [m], j \in [n]}$ we set

$$\operatorname{pat}_k(x,i,j) := \{(p,q) \in B_k(i,j) \ : \ \xi_{p,q} \neq 0\} - (i,j).$$

For $k, \nu \in \mathbb{N}$ and $t \in \{0,1,2\}$ we define now the following problem $\textsc{Rec}(k,\nu,t)$.

$\textsc{Rec}(k,\nu,t)$

Instance: $m,n \in k\mathbb{N}$,

$r_1,\dots,r_n \in \mathbb{N}_0$, (row sum measurements)

$c_1,\dots,c_m \in \mathbb{N}_0$, (column sum measurements)

$v(i,j) \in \{0,\nu\}$, $(i,j) \in C(m,n,k)$, (block measurements)

Task: Find $\xi_{p,q} \in \{0,1\}$, $(p,q) \in [m] \times [n]$, with

$$\begin{aligned} &\sum_{p \in [m]} \xi_{p,q} = r_q, && q \in [n], && \text{(row sums)} \\ &\sum_{q \in [n]} \xi_{p,q} = c_p, && p \in [m], && \text{(column sums)} \\ &\sum_{(p,q) \in B_k(i,j)} \xi_{p,q} \le v(i,j), && (i,j) \in C(m,n,k), && \text{(block constraints)} \\ &\operatorname{pat}_k(x,i,j) \in P(k,t), && (i,j) \in C(m,n,k), && \text{(pattern constraints),} \end{aligned}$$

or decide that no such solution exists.

In other words, we ask for 0/1-solutions that satisfy given row and column sums, *block constraints* of the form $\sum_{(p,q)\in B_k(i,j)} \xi_{i,j} \le v(i,j)$ with given $v(i,j) \in \{0,\nu\}$, and *pattern constraints* that restrict the potential locations of the 1's in each block. We remark that the $v(i,j)$, $(i,j) \in C(m,n,k)$, can be viewed as some part of prior knowledge. In particle tracking [20, 5], for instance, the $v(i,j)$ may reflect prior knowledge about physically meaningful particle trajectories; see also [3].

Our main results show that the computational complexity of $\textsc{Rec}(k,\nu,t)$ may change drastically when k, ν, or t is varied.

Theorem 1.

(i) $\textsc{Rec}(1,\nu,t) \in \mathbb{P}$ *for any* $\nu \in \mathbb{N}$ *and* $t \in \{0,1,2\}$.

(ii) $\textsc{Rec}(k,1,0) \in \mathbb{P}$ *for any* $k \ge 2$.

(iii) $\textsc{Rec}(k,\nu,2) \in \mathbb{P}$ *for any* $k \ge 2$ *and* $\nu \ge k$.

Theorem 2.

(i) $\textsc{Rec}(k,1,1) \in \mathbb{NP}$*-hard for any* $k \ge 2$.

(ii) $\textsc{Rec}(k,2,0) \in \mathbb{NP}$*-hard for any* $k \ge 2$.

The most notable changes are summarized in Table 1. Some of these changes may at first glance seem somewhat counterintuitive. For instance, restricting the solution space via pattern constraints turns the $\mathbb{NP}$-hard problem $\textsc{Rec}(k, 2, 0)$ $(k \geq 2)$, into the polynomial time solvable problem $\textsc{Rec}(k, 2, 2)$; Figure 2 depicts the possible types of blocks in $\textsc{Rec}(k, 2, 2)$. Conversely, additional pattern constraints convert the tractable problem $\textsc{Rec}(k, 1, 0)$ into the $\mathbb{NP}$-hard problem $\textsc{Rec}(k, 1, 1)$ $(k \geq 2)$.

	$\mathbb{P}$	$\mathbb{NP}$-hard
varying k	$\textsc{Rec}(1, 2, 0)$	$\textsc{Rec}(k, 2, 0)$
	$\textsc{Rec}(1, 1, 1)$	$\textsc{Rec}(k, 1, 1)$
varying ν	$\textsc{Rec}(k, 1, 0)$,	$\textsc{Rec}(k, 2, 0)$
varying t	$\textsc{Rec}(k, 2, 2)$	$\textsc{Rec}(k, 2, 0)$
	$\textsc{Rec}(k, 1, 0)$	$\textsc{Rec}(k, 1, 1)$

TABLE 1. Computational complexity of $\textsc{Rec}(1, \nu, t)$ and $\textsc{Rec}(k, \nu, t)$ for $k \geq 2$ under change of a single parameter.

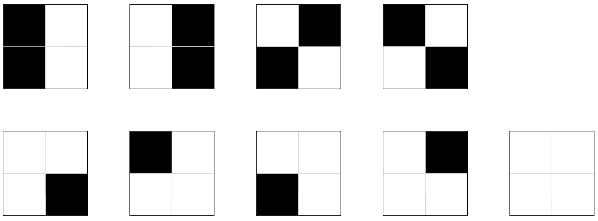


FIGURE 2. The 9 possible types of 1-entries (black pixels) in each block of any solution of a $\textsc{Rec}(2, 2, 2)$ instance.

3. Tractability results: Polynomial-time solvability

This section contains the proofs of our tractability results stated in Theorem 1.

Clearly, a polynomial-time algorithm for $\textsc{Rec}(1, \nu, t)$ can be given along the classical lines for deciding whether given row and column sums of a binary matrix are consistent. In the following we give one of the various arguments.

Proof of Theorem 1(i). The problem obviously reduces to that of reconstructing a 0/1-matrix from given row and column sums with some of the matrix entries fixed to 0. It is well known that the coefficient matrix is totally unimodular, see, e.g., [12, 2], [18, Sect. 19.3]. The problem can thus be solved efficiently, e.g., by linear programming. □

Next we turn to the proof of Theorem 1(ii). For $(i, j) \in I \subseteq C(m, n, k)$, let

$$\sigma_i(j) := |(\{i\} \times [j]) \cap I| \qquad \text{and} \qquad \rho_j(i) := |([i] \times \{j\}) \cap I|$$

denote the number of blocks $B_k(a, b)$ with $(a, b) \in I$, $b \leq j$ and $a \leq i$, that lie in the same vertical and horizontal strip, respectively, as the block $B_k(i, j)$. Moreover, for $I \subseteq C(m, n, k)$ we set

$$G(I) := \bigcup_{(i,j)\in I} B_k(i, j) \subseteq [m] \times [n],$$
$$\Pi_x(I) := \{i \in [m] : \exists j \in [n] : (i, j) \in I\},$$
$$\Pi_y(I) := \{j \in [n] : \exists i \in [n] : (i, j) \in I\}.$$

Note that $\Pi_x(I)$ and $\Pi_y(I)$ denotes the projection of I onto the first and second coordinate, respectively.

We use a (slightly generalized version of a) result of [4] on the following problem.

DR(1)

$$\begin{array}{llll}
\text{Instance:} & m, n \in k\mathbb{N}, & & \\
& I \subseteq C(m,n,k), & & \text{(a set of corner points)} \\
& r_{j+l} \in \mathbb{N}_0, & j \in \Pi_y(I),\ l \in [k-1]_0, & \text{(row sum measurem.)} \\
& c_{i+l} \in \mathbb{N}_0, & i \in \Pi_x(I),\ l \in [k-1]_0, & \text{(column sum measurem.)} \\
\text{Task:} & \text{Find } \xi_{p,q} \in \{0,1\},\ (p,q) \in G(I) \text{ with} & & \\
& \sum\limits_{p:(p,j)\in G(I)} \xi_{p,j+l} = r_{j+l}, & j \in \Pi_y(I),\ l \in [k-1]_0, & \text{(row sums)} \\
& \sum\limits_{q:(i,q)\in G(I)} \xi_{i+l,q} = c_{i+l}, & i \in \Pi_x(I),\ l \in [k-1]_0, & \text{(column sums)} \\
& \sum\limits_{(p,q)\in B_k(i,j)} \xi_{p,q} \quad = 1, & (i,j) \in I, & \text{(block constraints),} \\
& \text{or decide that no such solution exists.} & &
\end{array}$$

Proposition 1 ([4])**.**

(i) An instance $\mathcal{I}$ of DR(1) *is feasible if, and only if, for every $(i,j) \in I$ we have*

$$\sum_{l=0}^{k-1} r_{j+l} = \rho_j(m) \quad \text{and} \quad \sum_{l=0}^{k-1} c_{i+l} = \sigma_i(n).$$

(ii) DR(1) $\in \mathbb{P}$.

As a service to the reader we remark that a solution $\xi^*_{p,q}$, $(p,q) \in G(I)$, for a given instance of DR(1) is obtained by setting for every $(i,j) \in I$ and $(p,q) \in B_k(i,j)$:

$$\begin{aligned}
a_{i,j} &:= i + \min\{l \in \{0,1\} : \sigma_i(j) \le \sum_{h=0}^{l} c_{i+h}\}, \\
b_{i,j} &:= j + \min\{l \in \{0,1\} : \rho_j(i) \le \sum_{h=0}^{l} r_{j+h}\}, \\
\xi^*_{p,q} &:= \begin{cases} 1 & : \quad (p,q) = (a_{i,j}, b_{i,j}), \\ 0 & : \quad \text{otherwise.} \end{cases}
\end{aligned} \tag{3.1}$$

An illustration is given in Figure 3 (taken from [4]).

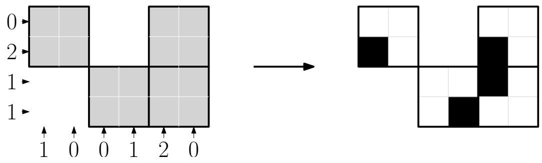


FIGURE 3. ([4]) Illustration of DR(1). (Left) Row and column sums and blocks $B_k(i,j)$ with $(i,j) \in I$ in gray color. (Right) Solution defined by (3.1).

Now we provide a polynomial-time algorithm for REC$(k,1,0)$.

Proof of Theorem 1(ii). Let $\mathcal{I} = (m, n, r_1, \ldots, r_n, c_1, \ldots, c_m, v(1,1), \ldots, v(m-k+1, n-k+1))$ denote an instance of REC$(k,1,0)$. We proceed in two steps.

First, we solve the following reconstruction-from-row-and-column-sums instance of finding $\eta_{i,j} \in \{0,1\}$, $(i,j) \in C(m,n,k)$, satisfying the constraints

$$\begin{aligned} \sum_{i:(i,j)\in I} \eta_{i,j} &= \sum_{l=0}^{k-1} r_{j+l}, && j \in [n] \cap (k\mathbb{N}_0 + 1), \\ \sum_{j:(i,j)\in I} \eta_{i,j} &= \sum_{l=0}^{k-1} c_{i+l}, && i \in [m] \cap (k\mathbb{N}_0 + 1), \\ \eta_{i,j} &= 0, && (i,j) \in \{(a,b) \in C(m,n,k) : v(a,b) = 0\}. \end{aligned} \tag{3.2}$$

If no solution exists, we report infeasibility of $\mathcal{I}$. The idea behind solving (3.2) first is that in this way we locate boxes that we want to contain a 1.

In a second step, if (3.2) is feasible, we invoke DR(1) from Proposition 1 for the instance $\mathcal{I}'$ where

$$\mathcal{I}' := (m, n, I, r_1, \ldots, r_n, c_1, \ldots, c_m)$$

and

$$I := \{(i,j) \in C(m,n,k) : \eta^*_{i,j} = 1\}. \tag{3.3}$$

If there is no solution we report infeasibility of $\mathcal{I}$; otherwise, we return the solution of DR(1). It remains to be shown that this solves $\mathcal{I}$ correctly.

To this end, suppose that $\mathcal{I}$ has a solution $\xi^*_{i,j}$, $(i,j) \in [m] \times [n]$. Then, (3.2) is also feasible (a solution is given by $\eta^*_{i,j} := \sum_{(p,q)\in B_k(i,j)} \xi^*_{p,q}$, $(i,j) \in C(m,n,k)$) and $\xi^*_{i,j}$ satisfies the corresponding instance of DR(1). Conversely, let $\eta^*_{i,j}$, $(i,j) \in C(m,n,k)$, denote a solution to (3.2) and let I be defined as in (3.3). For every $(i,j) \in I$ we have clearly

$$\sum_{l=0}^{k-1} r_{j+l} = \rho_j(m) \quad \text{and} \quad \sum_{l=0}^{k-1} c_{i+l} = \sigma_i(n).$$

By Proposition 1 there is thus a solution to DR(1). This solution satisfies all row and column sum constraints of $\mathcal{I}$ by definition. Further since $\{(i,j) \in C(m,n,k) : v(i,j) = 0\} \cap I = \emptyset$ by definition of (3.2), there are 1's in the solution only in those blocks $B_k(i,j)$, $(i,j) \in C(m,n,k)$, for which $v(i,j) = 1$ holds. The block constraints are thus also satisfied, which proves the claim.

The integer linear program (3.2) can be solved in polynomial time as the coefficient matrix is well known to be totally unimodular; see, e.g., [12, 2], [18, Sect. 19.3]. The set I in (3.3) is determined in polynomial time, and DR(1) $\in \mathbb{P}$ by Proposition 1(ii). Hence, Rec$(k,1,0) \in \mathbb{P}$. □

Next we turn to the proof of Theorem 1(iii). It relies on the following result.

Lemma 1. *Let $A = (a_1, \ldots, a_{m_1})^T \in \{0,1\}^{m_1 \times n_1}$ be the node-edge incidence matrix of a (simple) bipartite graph, and let $H = (h_1, \ldots, h_{m_2})^T$ denote a binary $m_2 \times n_1$ matrix with the following properties.*

(i) For every $l \in [m_2]$ there exists an index $i \in [m_1]$ with supp$(h_l) \subseteq$ supp(a_i);
(ii) The supports of any two distinct vectors h_i and h_l of H with supp$(h_i) \cup$ supp$(h_l) \subseteq$ supp(a_k) for some $k \in [m_1]$ do not intersect.

Then, $\begin{pmatrix} A \\ H \end{pmatrix}$ is totally unimodular.

Proof. It suffices to prove the assertion under the additional assumption that

$$|\text{supp}(h_l)| \geq 2, \qquad \text{for all } l \in [m_2]. \tag{3.4}$$

All vectors h_l whose support is a singleton can be added later since appending any subset of rows of the $m_1 \times n_1$-identity matrix to a totally unimodular matrix yields again a totally unimodular matrix.

As the matrix A is the node-edge incidence matrix of a bipartite graph, let (R_1, R_2) denote a corresponding partition of (the indices of) the vertices of the graph i.e., a partition of $[m_1]$. Clearly, for every $i \neq l$ with $(i,l) \subseteq (R_1 \times R_1) \cup (R_2 \times R_2)$ we have

$$\operatorname{supp}(a_i) \cap \operatorname{supp}(a_l) = \emptyset. \tag{3.5}$$

It suffices to prove that every collection of rows of

$$M := \begin{pmatrix} A \\ H \end{pmatrix}$$

can be split into two parts such that the sum of the rows in one part minus the sum of the rows in the other part is a vector with entries in $\{0, \pm 1\}$; see e.g. [18, Theorem 19.3]. Hence, let $I \in [m_1]$ and $L \in [m_2]$ denote such a collection of row indices of A and H, respectively. For $R \subseteq [m_1]$ we set

$$L(R) := \{l \in L \,:\, \exists i \in R \text{ with } \operatorname{supp}(h_l) \subseteq \operatorname{supp}(a_i)\}.$$

Of course, the sets $(R_1 \cap I)$ and $(R_2 \cap I)$ form a partition of I, i.e.,

$$I = (R_1 \cap I) \ \cup \ (R_2 \cap I) \quad \text{and} \quad (R_1 \cap I) \ \cap \ (R_2 \cap I) = \emptyset. \tag{3.6}$$

Also, it follows from (i), (ii), (3.4), (3.5), and the fact that the graph is simple that

$$\begin{aligned} &L = L(R_1 \cap I) \ \cup \ L(R_1 \setminus I) \ \cup \ L(R_2 \cap I) \ \cup \ L(R_2 \setminus I) \quad \text{and} \\ &L(R_1 \cap I), \ \ L(R_1 \setminus I), \ L(R_2 \cap I), \ L(R_2 \setminus I) \quad \text{are pairwise disjoint,} \end{aligned} \tag{3.7}$$

i.e., the sets constitute a partition of L.

By (i), (ii), (3.5), (3.6) and (3.7) we have

$$\sum_{i \in R_1 \cap I} a_i + \sum_{l \in L(R_1 \setminus I)} h_l \in \{0,1\}^{n_1},$$

hence

$$\sum_{i \in R_1 \cap I} a_i + \sum_{l \in L(R_1 \setminus I)} h_l - \sum_{l \in L(R_1 \cap I)} h_l \in \{0,1\}^{n_1}$$

Similarly,

$$\sum_{i \in R_2 \cap I} a_i + \sum_{l \in L(R_2 \setminus I)} h_l - \sum_{l \in L(R_2 \cap I)} h_l \in \{0,1\}^{n_1}.$$

Therefore,

$$\sum_{i \in R_1 \cap I} a_i + \sum_{l \in L(R_1 \setminus I)} h_l - \sum_{l \in L(R_1 \cap I)} h_l - \left(\sum_{i \in R_2 \cap I} a_i + \sum_{l \in L(R_2 \setminus I)} h_l - \sum_{l \in L(R_2 \cap I)} h_l \right) \in \{0, \pm 1\}^{n_1},$$

i.e., the collection I and L of rows of A and H, respectively, is split into two parts such that the sum of the rows in one part minus the sum of the rows in the other part is a vector with entries in $\{0, \pm 1\}$. □

Now we present a polynomial-time algorithm for Rec$(k, \nu, 2)$ when $2 \leq k \leq \nu$.

Proof of Theorem 1(iii). We claim that the problem can be formulated as that of finding an integer solution to $\{x \,:\, Mx \leq z,\ x \geq 0\}$ with a totally unimodular matrix M and integral right-hand side z, showing that the problem is solvable in polynomial time; see [12], [18, Thm. 16.2].

Assembling the variables $\xi_{i,j}$, for $(i,j) \in [m] \times [n]$ into an mn-dimensional vector and rephrasing the row and column sum constraints in matrix form $Ax = (r^T, c^T)^T$, where r contains the r_j's and c contains the c_i's, A is the node-edge incidence matrix of a bipartite graph, and hence totally unimodular.

As the pattern constraints ensure that no row of a block contains two 1's, and since $v(i,j) = 0$ or $v(i,j) \geq k$, we can (equivalently) replace the block and pattern constraints by the *box constraints*

$$\sum_{(p,q) \in W(i,j+l)} \xi_{p,q} \leq \min\{1, v(i,j)\}, \qquad (i,j) \in C(m,n,k), \quad l \in [k-1]_0, \tag{3.8}$$

where $W(i,j+l) := B_k(i,j) \cap ([m] \times \{j+l\})$.

We can rephrase the box constraints in matrix form $Hx \leq v$, with a binary matrix H and a vector v containing the respective right-hand sides.

Since each box $W(i, j+l)$ is contained in a row of $[m] \times [n]$ the condition in Lemma 1(i) is satisfied. Also, any two of these boxes are disjoint since the blocks are disjoint. Hence the condition in Lemma 1(ii) is satisfied. Therefore, by Lemma 1, the matrix

$$\begin{pmatrix} A \\ H \end{pmatrix}$$

is also totally unimodular. Appending $-A$ and the identity matrix E yields again a totally unimodular matrix, hence

$$M := \begin{pmatrix} A \\ H \\ -A \\ E \end{pmatrix}$$

is totally unimodular. Since the corresponding right-hand side vector $z = (r^T, c^T, v^T, -r^T, -c^T, \mathbf{1}^T)^T$ is integral, the instance can be solved by linear programming. □

4. Intractability results: $\mathbb{NP}$-hardness

This section contains the proof of Theorem 2. We begin with the $\mathbb{NP}$-hardness of Rec$(k, 1, 1)$ for $k \geq 2$.

Proof of Theorem 2(i). It suffices to show the result for $k = 2$ since the $\mathbb{NP}$-hardness for larger k can be inferred from that for $k = 2$ by setting the row and column sums r_{j+l} and c_{i+l} to zero for every $(i, j) \in C(m, n, k)$ and $l \in \{3, \ldots, k-1\}$; see [4].

So, let $k = 2$. We use a transformation from the following problem 3-color tomography, which is $\mathbb{NP}$-hard by [7]. (Note that the third color is just "blank.")

3-color tomography

Instance: $m, n \in \mathbb{N}$, $r_1^{(a)}, \ldots, r_n^{(a)}, c_1^{(a)}, \ldots, c_m^{(a)} \in \mathbb{N}_0$ for $a \in [2]$.

Task: Find $\xi_{p,q}^{(a)} \in \{0, 1\}$, $(p, q) \in [m] \times [n]$, $a \in [2]$, with

$$\sum_{p \in [m]} \xi_{p,q}^{(a)} = r_q^{(a)}, \qquad q \in [n], a \in [2], \qquad \text{(row sums)}$$

$$\sum_{q \in [n]} \xi_{p,q}^{(a)} = c_p^{(a)}, \qquad p \in [m], a \in [2], \qquad \text{(columns sums)}$$

$$\xi_{p,q}^{(1)} + \xi_{p,q}^{(2)} \leq 1, \qquad (p, q) \in [m] \times [n] \qquad \text{(disjointness condition),}$$

or decide that no such solution exists.

Let $\mathcal{I} = (m, n, r_1^{(1)}, \ldots, r_n^{(1)}, c_1^{(1)}, \ldots, c_m^{(1)}, r_1^{(2)}, \ldots, r_n^{(2)}, c_1^{(2)}, \ldots, c_m^{(2)})$ denote an instance of 3-color tomography. We are setting up an instance $\mathcal{I}'$ of our reconstruction problem by defining the row and column sums $r_1, \ldots, r_{2n}$ and $c_1, \ldots, c_{2m}$, respectively.

The block and pattern constraints ensure that there are only three possibilities for setting the 1's in each block. They are shown in Figure 4.

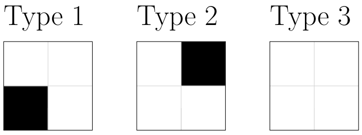


Figure 4. The three possible types of blocks in solutions of Rec$(2, 1, 1)$ instances.

Note that the types can be viewed as representing three colors, which are counted by different row and column sums. The row and column sums with odd indices count type 1 blocks, the other sums count type 2 blocks.

Setting

$$\begin{aligned} r_{2q+a-2} &:= r_q^{(a)}, \quad q \in [n], \;\; a \in [2], \\ c_{2p+a-2} &:= c_p^{(a)}, \quad p \in [m], \;\; a \in [2], \end{aligned}$$

and using the correspondences

$$\begin{aligned} \xi_{p,q}^{(a)^*} = 1 \quad &\Leftrightarrow \quad (\xi^*_{2p-1,2q-1}, \xi^*_{2p-1,2q}, \xi^*_{2p,2q-1}, \xi^*_{2p,2q}) \text{ is of type a,} \qquad \text{for } a \in [2], \\ \xi_{p,q}^{(1)^*} = \xi_{p,q}^{(2)^*} = 0 \quad &\Leftrightarrow \quad (\xi^*_{2p-1,2q-1}, \xi^*_{2p-1,2q}, \xi^*_{2p,2q-1}, \xi^*_{2p,2q}) \text{ is of type 3,} \end{aligned}$$

we conclude that $\mathcal{I}$ admits a solution if, and only if, $\mathcal{I}'$ admits a solution. □

Next we turn to the $\mathbb{NP}$-hardness of REC$(k, 2, 0)$ for $k \geq 2$. Again it suffices to prove the result for $k = 2$.

Proof of Theorem 2(ii). The general structure of the proof follows that of the $\mathbb{NP}$-hardness result under data uncertainty given in [4]. However, some adaptations and additions are required as will be detailed below.

As in [4] we give a transformation from the $\mathbb{NP}$-hard problem 1-IN-3-SAT [6], which asks for a satisfying truth assignment that sets exactly one literal true in each clause of a given Boolean formula in conjunctive normal form where all clauses contain three literals (involving three different variables).

For a given instance of 1-IN-3-SAT a "circuit board" is constructed that contains an initializer, several connectors, and clause chips. The general structure of the circuit board from the proof of [4] and its modification, which contains additional connectors and initializers, are outlined in Figure 5.

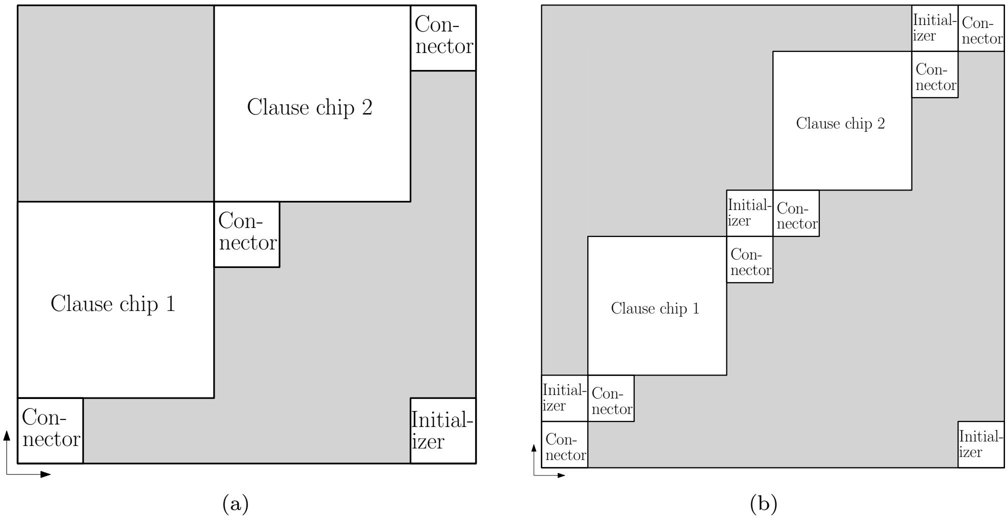


FIGURE 5. The general layout of (a) the original circuit board and (b) the modified circuit board, here for two clauses.

The initializers contain, for every variable τ_t, $t \in [T]$, so-called *τ_t-chips*, while the connectors contain *$\neg\tau_t$-chips.* The clause chips are more complex as they consist of two *collectors*, two *verifiers*, and a *transmitter*. The initializer holds a truth assignment for the variables $\tau_1, \ldots, \tau_T$ of the given instance $\mathcal{I}$ of 1-IN-3-SAT. The Boolean values TRUE and FALSE are encoded in τ_t-chips, $t \in [T]$, by the type 1 and 2 blocks, respectively, that are shown in Figure 6.

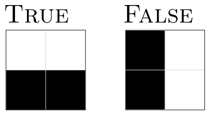
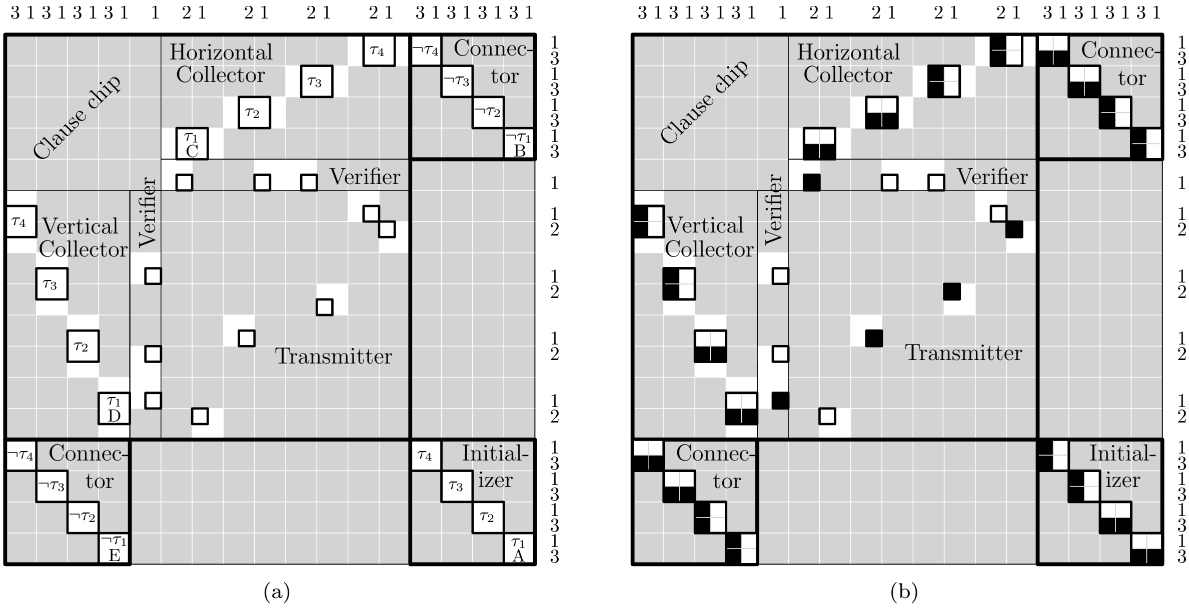


FIGURE 6. Boolean values in τ_t-chips.

The proof in [4] shows how the truth assignment is transmitted through the circuit board, and how the verifier chips indeed check the feasibility of the given 1-IN-3-SAT instance. Figure 7 depicts a specific problem instance and a corresponding solution for the original construction of [4] .



(a)

(b)

FIGURE 7. ([4]) Original transformation from 1-IN-3-SAT for the instance that involves the single clause $\tau_1 \vee \neg\tau_2 \vee \tau_3$. (a) The circuit board. By setting to zero suitable row and column sums it is ensured that the non-zero components $\xi^*_{p,q}$ of a solution are only possible in the bold-framed boxes within the white blocks in the clause chip, connectors, and the initializer. (b) A solution x^* (non-zero components $\xi^*_{p,q}$ are depicted as black pixels), representing the Boolean solution $(\tau^*_1, \tau^*_2, \tau^*_3, \tau^*_4) =$ (TRUE, TRUE, FALSE, FALSE).

The proof in [4] employs three different types of block constraints. For every block $B_2(i,j)$, $(i,j) \in C(m,n,2)$, there is one of the following constraints:

$$\sum_{(p,q)\in B_2(i,j)} \xi_{p,q} = 0, \qquad ((=,0)\text{-block constraint})$$

$$\sum_{(p,q)\in B_2(i,j)} \xi_{p,q} = 2, \qquad ((=,2)\text{-block constraint})$$

$$\sum_{(p,q)\in B_2(i,j)} \xi_{p,q} \leq 2, \qquad ((\approx,1)\text{-block constraint});$$

the names in parenthesis are those from [4] (even though $(\leq, 2)$ may seem more natural here in the third case).

As for binary variables, the constraint $\sum_{(p,q)\in B_2(i,j)} \xi_{p,q} = 0$ is equivalent to $\sum_{(p,q)\in B_2(i,j)} \xi_{p,q} \leq 0$, it suffices to adapt the construction in such a way that no $(=,2)$-block constraints are required. In fact, the $(=,2)$-block constraints in the original construction are only employed for the $\neg\tau_t$-chips, $t \in [T]$, contained in the connectors.

For each $\neg\tau_t$-chip, $t \in [T]$, in a connector, we replace the $(=, 2)$-block constraints by a $(\approx, 1)$-block constraint. Then, between any two clause chips we insert an additional copy of a connector and an initializer each. These copies are placed as indicated in Figure 5(b). The additional row and column sums are prescribed again to values 1 and 3 in alternation. The new copies ensure that each $\neg\tau_t$-chip, $t \in [T]$, is contained in a horizontal or vertical strip that contains a τ_t-chip of an initializer. The corresponding row or column sums have values 1 and 3, and since we have $(\approx, 1)$-block constraints for both chips, we can distribute the $1 + 3 = 4$ ones only in such a way that each of the two chips contains exactly two 1's. In other words, each $\neg\tau_t$-chip, $t \in [T]$, is required to contain exactly two 1's. Figure 8 shows the solution of our correspondingly adapted circuit board from Figure 7.

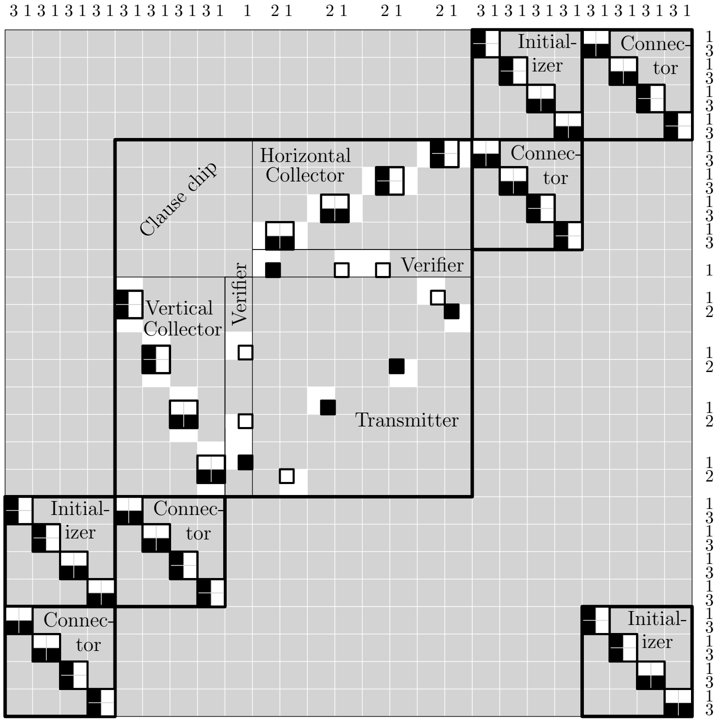


FIGURE 8. Transformation from 1-IN-3-SAT for the instance that involves the single clause $\tau_1 \vee \neg\tau_2 \vee \tau_3$. A solution x^* is shown (non-zero components $\xi^*_{p,q}$ are depicted as black pixels), representing the Boolean solution $(\tau_1^*, \tau_2^*, \tau_3^*, \tau_4^*) = (\text{TRUE}, \text{TRUE}, \text{FALSE}, \text{FALSE})$.

With this adaptation the proof from [4] carries over. As a service to the reader we explain how the truth assignment is transmitted in our particular example; see Figure 9. The formal proof that, in particular, this transmission and, in general, the complete transformation works as needed, follows exactly as in [4].

Suppose the τ_1-chip of the initializer in the lower right corner of the circuit board shown in Figure 9 is of type 1. The truth assignment is vertically transmitted from the initializer to a connector, which negates the Boolean value as the X-rays are set to 1 and 3 in alternation. Then, this Boolean value is horizontally transmitted to an initializer (again negated, so that we have the original values back), then vertically to a connector (again negated). From here the Boolean value is transmitted horizontally to the first clause chip, where it enters and exits, yet again negated, through a collector. In the part between the two collectors, the Boolean values for the literals appearing in the first clause take a path through the verifiers that ensure that exactly one literal is set to true; the values for the other literals are just transmitted (again with negation). This process continues in a similar way for the remaining clauses. We end up with feasible solution for our instance of REC$(k, 2, 0)$ if, and only if, there is a satisfying truth assignment for our given instance of 1-IN-3-SAT. □

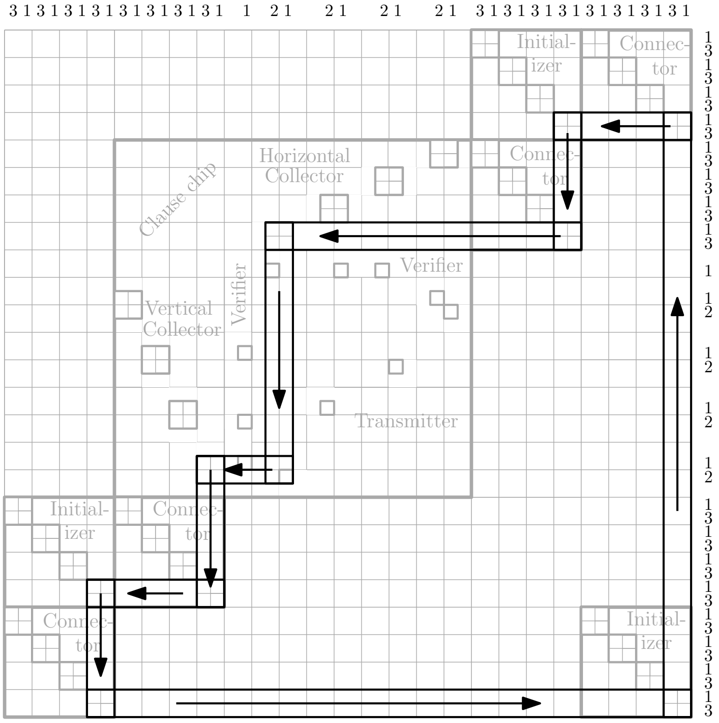


FIGURE 9. Transmission of the Boolean values for τ_1 (for the example shown in Figure 8).

5. Final Remarks

Let us first point out that the question of inferring information about an otherwise unknown object from observations “under the microscope,” i.e., through certain windows, is fundamental for a wide range of applications. For corresponding results on the scanning of binary or integer matrices; see e.g. [9, 10]. In our terminology, [9] studies the task of reconstructing binary matrices from their number of 1’s in each windows of a fixed size.

We will close by introducing a more general problem and stating some implications of our main Theorems 1 and 2.

As before, $m, n, k \in \mathbb{N}$. Further, for $V_{\le}, V_{\ge}, V_{=} \subseteq \mathbb{N}_0$ we set

$$R(V_{\le}, V_{\ge}, V_{=}) := (\{\le\} \times V_{\le}) \cup (\{\ge\} \times V_{\ge}) \cup (\{=\} \times V_{=}).$$

Also, let for $(i, j) \in \mathbb{N}_0^2$

$$W_k(i, j) := (i, j) + [k-1]_0^2,$$

and define for $L \subseteq \mathbb{N}^2$

$$C(m, n, k, L) := \{(p, q) \in L \ : \ W_k(p, q) \subseteq [m] \times [n]\}\,.$$

For $L \subseteq \mathbb{N}^2$, $(i, j) \in C(m, n, k, L)$, and $x = (\xi_{i,j})_{i\in[m],j\in[n]}$ we set

$$\mathrm{pat}_k(x, i, j) := \{(p, q) \in W_k(i, j) \ : \ \xi_{p,q} \neq 0\} - (i, j).$$

In addition to the patterns $P(k, 0), P(k, 1)$, and $P(k, 2)$ we consider, for $k \ge 2$, a fourth one

$$P(k, 3) := \{M \in 2^{[k-1]_0^2} \ : \ |M \cap ([k-1]_0 \times \{j\})\,| \ge k-1 \text{ for all } j \in [k]_0\}.$$

Note that $P(k, 3)$ can be viewed as the colored-inverted version of $P(k, 2)$.

Now, let $V_{\le}, V_{\ge}, V_{=} \subseteq \mathbb{N}_0$, $t \in [3]_0$, and $L \subseteq \mathbb{N}^2$. Then we define the following problem.

$\mathrm{WRec}(k,V_{\leq},V_{\geq},V_{=},t,L)$

$$\begin{array}{llll}
\text{Instance:} & m,n\in k\mathbb{N}, & & \\
 & r_1,\dots,r_n\in\mathbb{N}_0, & & \text{(row sum measurem.)}\\
 & c_1,\dots,c_m\in\mathbb{N}_0, & & \text{(col. sum measurem.)}\\
 & (\sim_{i,j},v_{i,j})\in R(V_{\leq},V_{\geq},V_{=}), & (i,j)\in C(m,n,k,L), & \text{(window measurem.)}\\[2ex]
\text{Task:} & \text{Find } \xi_{p,q}\in\{0,1\},\ (p,q)\in[m]\times[n], \text{ with} & & \\
 & \displaystyle\sum_{p\in[m]}\xi_{p,q} \quad = r_q, & q\in[n], & \text{(row sums)}\\
 & \displaystyle\sum_{q\in[n]}\xi_{p,q} \quad = c_p, & p\in[m], & \text{(column sums)}\\
 & \displaystyle\sum_{(p,q)\in W_k(i,j)}\xi_{p,q} \quad \sim_{i,j} v(i,j), & (i,j)\in C(m,n,k,L), & \text{(window constraints)}\\
 & \mathrm{pat}_k(x,i,j)\in P(k,t), & (i,j)\in C(m,n,k,L), & \text{(pattern constraints),}\\[1ex]
 & \text{or decide that no such solution exists.} & &
\end{array}$$

Tables 2 and 3 below list tractability and intractability results for $\mathrm{WRec}(k,V_{\leq},V_{\geq},V_{=},t,L)$, respectively, which are simple corollaries to our main Theorems 1 and 2. In fact, they are either just reinterpretations of these results or rely, in addition, on one or two of the following simple transformation principles (T1), (T2), or (T3).

(T1) is the trivial observation that rather than reconstructing the ones in an image, we may reconstruct the zeros instead. More precisely, this process of "color inversion" works as follows. Let α denote the number of positions in the window $W_k(0,0)$. Then, we associate a given instance $\mathcal{I}$ of $\mathrm{WRec}(k,V_{\leq},V_{\geq},V_{=},t,L)$ with the instance $\mathcal{I}'$ of $\mathrm{WRec}(k,V'_{\leq},V'_{\geq},V'_{=},t,L)$ where $V'_{\leq}:=\{\alpha-l\ :\ l\in V_{\leq}\}$, $V'_{\geq}:=\{\alpha-l\ :\ l\in V_{\geq}\}$, and $V'_{=}:=\{\alpha-l\ :\ l\in V_{=}\}$, $r'_j:=m-r_j$, $j\in[n]$, $c'_i:=n-c_i$, $i\in[m]$, $P'(k,t):=\{W_k(0,0)\setminus M\ :\ M\in P(k,(t)\}$, and, for all $(i,j)\in C$, $(\sim_{i,j},v(i,j))$ is replaced by

$$\left\{\begin{array}{lcl}
(\geq,\alpha-v(i,j)) & : & \sim_{i,j}\in\{\leq\},\\
(\leq,\alpha-v(i,j)) & : & \sim_{i,j}\in\{\geq\},\\
(=,\alpha-v(i,j)) & : & \sim_{i,j}\in\{=\}.
\end{array}\right.$$

Clearly, the problems $\mathrm{WRec}(k,V_{\leq},V_{\geq},V_{=},t,L)$ and $\mathrm{WRec}(k,V'_{\leq},V'_{\geq},V'_{=},t,L)$ lie in the same complexity class.

The transformation (T2) adds empty rows and columns to an instance $\mathcal{I}$ of $\mathrm{WRec}(2,V_{\leq},V_{\geq},V_{=},0,L)$ to extend $\mathbb{NP}$-hardness results for $k=2$ to higher values of k. More precisely, let $m':=mk/2$, $n':=nk/2$, and set for $(i,j)\in([m]\times[n])\cap C(m,n,2)$ and $l\in[k]$

$$\begin{aligned}
r'_{\frac{k}{2}(j-1)+l} &:= \left\{\begin{array}{lcl} r_{j+l} & : & l\in[2],\\ 0 & : & \text{otherwise,}\end{array}\right.\\
c'_{\frac{k}{2}(i-1)+l} &:= \left\{\begin{array}{lcl} c_{i+l} & : & l\in[2],\\ 0 & : & \text{otherwise,}\end{array}\right.\\
(\sim'_{i,j},v'(k(i-1)/2+1,k(j-1)/2+1)) &:= (\sim_{i,j},v(i,j)).
\end{aligned}$$

This defines an instance $\mathcal{I}'$ of $\mathrm{WRec}(k,V_{\leq},V_{\geq},V_{=},0,L)$ with row sums $r'_1,\dots,r'_{n'}$, column sums $c'_1,\dots,c'_{m'}$, and block constraints $(\sim'_{i,j},v'(i,j))$, $(i,j)\in C(m',n',3,k,L)$. Clearly, the instance $\mathcal{I}$ admits a solution if, and only if, $\mathcal{I}'$ admits a solution (by filling/extracting the 2×2-blocks of $\mathcal{I}$ into/from the $k\times k$-blocks of $\mathcal{I}'$). As this is a polynomial-time transformation, $\mathbb{NP}$-hardness of $\mathrm{WRec}(2,V_{\leq},V_{\geq},V_{=},0,L)$ implies $\mathbb{NP}$-hardness of $\mathrm{WRec}(k,V_{\leq},V_{\geq},V_{=},0,L)$.

In the final transformation (T3) additional rows and columns are filled by ones. For a given instance $\mathcal{I}$ of $\mathrm{WRec}(2,V_{\leq},V_{\geq},V_{=},0,L)$, and $k\geq 2$, let $m':=mk/2$, $n':=nk/2$, and set for $(i,j)\in([m]\times[n])\cap$

$C(m,n,2)$ and $l \in [k]$

$$r'_{\frac{k}{2}(j-1)+l} := \begin{cases} r_{j+l} & : \quad l \in [2], \\ m' & : \quad \text{otherwise}, \end{cases}$$

$$c'_{\frac{k}{2}(i-1)+l} := \begin{cases} c_{i+l} & : \quad l \in [2], \\ n' & : \quad \text{otherwise}, \end{cases}$$

$$(\sim'_{i,j}, v'(k(i-1)/2+1, k(j-1)/2+1)) := (\sim_{i,j}, k^2+v(i,j)-4).$$

This defines an instance $\mathcal{I}'$ of $\text{WREC}(k, V'_{\leq}, V'_{\geq}, V'_{=}, 0, L)$ with row sums $r'_1, \ldots, r'_{n'}$, column sums $c'_1, \ldots, c'_{m'}$, block constraints $(\sim'_{i,j}, v'(i,j))$, $(i,j) \in C(m', n', 3, k, L)$, $V'_{\leq} := \{k^2+v-4 : v \in V_{\leq}\}$, $V'_{\geq} := \{k^2+v-4 : v \in V_{\geq}\}$, and $V'_{=} := \{k^2+v-4 : v \in V_{=}\}$. Clearly, the instance $\mathcal{I}$ admits a solution if, and only if, $\mathcal{I}'$ admits a solution (by filling/extracting the 2×2-blocks of $\mathcal{I}$ into/from the $k \times k$-blocks of $\mathcal{I}'$). As this is a polynomial-time transformation, $\mathbb{NP}$-hardness of $\text{WREC}(2, V_{\leq}, V_{\geq}, V_{=}, 0, L)$ implies $\mathbb{NP}$-hardness of $\text{WREC}(k, V'_{\leq}, V'_{\geq}, V'_{=}, 0, L)$.

k	$V_{\leq}$	$V_{\geq}$	$V_{=}$	t	L	Reference
1	$\mathbb{N}_0$	$\emptyset$	$\emptyset$	$[2]_0$	$\mathbb{N}^2$	Thm. 1(i)
2	$\{0\}$	$\{4\}$	$[4]_0$	0	$(2\mathbb{N}_0+1)^2$	[4]
≥ 2	$\{0,1\}$	$\{k^2\}$	$\{0,k^2\}$	0	$(k\mathbb{N}_0+1)^2$	Thm. 1(ii)
≥ 2	$\{0\}$	$\{k^2-1,k^2\}$	$\{0,k^2\}$	0	$(k\mathbb{N}_0+1)^2$	Thm. 1(ii) & (T1)
≥ 2	$\{0\} \cup \{\nu : \nu \geq k\}$	$\emptyset$	$\{0\}$	2	$(k\mathbb{N}_0+1)^2$	Thm. 1(iii)
≥ 2	$\emptyset$	$\{k^2\} \cup [k(k-1)]_0$	$\{k^2\}$	3	$(k\mathbb{N}_0+1)^2$	Thm. 1(iii) & (T1)

TABLE 2. Polynomial-time solvable problems $\text{WREC}(k, V_{\leq}, V_{\geq}, V_{=}, t, L)$.

k	$V_{\leq}$	$V_{\geq}$	$V_{=}$	t	L	Reference
≥ 2	$\emptyset$	$\{0\}$	$\{0,1,2\}$	0	$\mathbb{N}^2$	[4]
≥ 2	$\{0,1\}$	$\emptyset$	$\emptyset$	1	$(k\mathbb{N}_0+1)^2$	Thm. 2(i)
≥ 2	$\{0,2\}$	$\emptyset$	$\emptyset$	0	$(k\mathbb{N}_0+1)^2$	Thm. 2(ii)
≥ 2	$\{k^2-4,k^2-2\}$	$\emptyset$	$\emptyset$	0	$(k\mathbb{N}_0+1)^2$	Thm. 2(ii) & (T3)
≥ 2	$\emptyset$	$\{k^2-2,k^2\}$	$\emptyset$	0	$(k\mathbb{N}_0+1)^2$	Thm. 2(ii) & (T1)
≥ 2	$\emptyset$	$\{2,4\}$	$\emptyset$	0	$(k\mathbb{N}_0+1)^2$	Thm. 2(ii) & (T1) & (T2)

TABLE 3. $\mathbb{NP}$-hard problems $\text{WREC}(k, V_{\leq}, V_{\geq}, V_{=}, t, L)$.

Let us finally point out, that the Tables 2 and 3 can be largely extended. In particular, we can consider reconstruction problems for different types of windows which are even allowed to vary for each $(i,j) \in [m] \times [n]$. While some of the complexity results in this more general setting may seem rather marginal, other may be interesting for certain applications; see [3]. As this would go far beyond the scope of the present paper we refrain, however, from carrying the results to the extremes here.

REFERENCES

[1] S. Van Aert, K. J. Batenburg, M. D. Rossell, R. Erni, and G. Van Tendeloo. Three-dimensional atomic imaging of crystalline nanoparticles. *Nature*, 470(7334):374–376, 2011.

[2] R. Aharoni, G. T. Herman, and A. Kuba. Binary vectors partially determined by linear equation systems. *Discrete Math.*, 171(1-3):1–16, 1997.

[3] A. Alpers and P. Gritzmann. Dynamic discrete tomography. manuscript, 2017.

[4] A. Alpers and P. Gritzmann. On double-resolution imaging and discrete tomography. submitted, http://arxiv.org/abs/1701.04399, 2017.

[5] A. Alpers, P. Gritzmann, D. Moseev, and M. Salewski. 3D particle tracking velocimetry using dynamic discrete tomography. *Comput. Phys. Commun.*, 187(1):130–136, 2015.

[6] S. A. Cook. The complexity of theorem-proving procedures. In *Proc. 3rd Ann. ACM Symp. Theory of Computing*, pages 151–158. ACM, 1971.

[7] C. Dürr and M. Matamala F. Guiñez. Reconstructing 3-colored grids from horizontal and vertical projections is NP-hard. *SIAM J. Discrete Math*, 26(1):330–352, 2012.

[8] P. Fishburn, J. Lagarias, J. Reeds, and L. Shepp. Sets uniquely determined by projections on axes. II: discrete case. *Discrete Math.*, 91(2):149–159, 1991.

[9] A. Frosini and M. Nivat. Binary matrices under the microscope: A tomographical problem. *Theor. Comput. Sci.*, 370(1-3):201–217, 2007.

[10] A. Frosini, M. Nivat, and S. Rinaldi. Scanning integer matrices by means of two rectangular windows. *Theor. Comput. Sci.*, 406(1-2):90–96, 2008.

[11] R. J. Gardner and P. Gritzmann. Discrete tomography: Determination of finite sets by x-rays. *Trans. Amer. Math. Soc.*, 349(6):2271–2295, 1997.

[12] P. Gritzmann. On the reconstruction of finite lattice sets from their x-rays. In *Discrete Geometry for Computer Imagery (Eds.: E. Ahronovitz and C. Fiorio), DCGI'97, Lecture Notes on Computer Science 1347, Springer*, pages 19–32, 1997.

[13] P. Gritzmann, B. Langfeld, and M. Wiegelmann. Uniqueness in discrete tomography: Three remarks and a corollary. *SIAM J. Discrete Math.*, 25(4):1589–1599, 2011.

[14] G. T. Herman and A. Kuba (eds.). *Discrete Tomography: Foundations, Algorithms and Applications.* Birkhäuser, Boston, 1999.

[15] G. T. Herman and A. Kuba (eds.). *Advances in Discrete Tomography and its Applications.* Birkhäuser, Boston, 2007.

[16] C. Kisielowski, P. Schwander, F. H. Baumann, M. Seibt, Y. Kim, and A. Ourmazd. An approach to quantitative high-resolution transmission electron microscopy of crystalline materials. *Ultramicroscopy*, 58(2):131–155, 1995.

[17] H. J. Ryser. Combinatorial properties of matrices of zeros and ones. *Canad. J. Math.*, 9(1):371–377, 1957.

[18] A. Schrijver. *Theory of Linear and Integer Programming.* John Wiley & Sons, Chichester, UK, 1986.

[19] P. Schwander, C. Kisielowski, F. H. Baumann, Y. Kim, and A. Ourmazd. Mapping projected potential, interfacial roughness, and composition in general crystalline solids by quantitative transmission electron microscopy. *Phys. Rev. Lett.*, 71(25):4150–4153, 1993.

[20] J. Zhu, J. Gao, A. Ehn, M. Aldén, Z. Li, D. Moseev, Y. Kusano, M. Salewski, A. Alpers, P. Gritzmann, and M. Schwenk. Measurements of 3D slip velocities and plasma column lengths of a gliding arc discharge. *Appl. Phys. Lett.*, 106(4):044101–1–4, 2015.