ON DOUBLE-RESOLUTION IMAGING IN DISCRETE TOMOGRAPHY

ANDREAS ALPERS AND PETER GRITZMANN

Abstract. Super-resolution imaging aims at improving the resolution of an image by enhancing it with other images or data that might have been acquired using different imaging techniques or modalities. In this paper we consider the task of doubling the resolution of tomographic grayscale images of binary objects by fusion with double-resolution tomographic data that has been acquired from two viewing angles. We show that this task is polynomial-time solvable if the gray levels have been reliably determined. The task becomes $\mathbb{NP}$-hard if the gray levels of some pixels come with an error of ± 1 or larger. The $\mathbb{NP}$-hardness persists for any larger resolution enhancement factor. This means that noise does not only affect the quality of a reconstructed image but, less expectedly, also the algorithmic tractability of the inverse problem itself.

1. Introduction

The results presented in this paper are motivated by the task of enhancing the resolution of reconstructed tomographic images obtained from binary objects representing, for instance, crystalline structures, nanoparticles or two-phase samples [1, 34, 6, 37, 3]. The reconstructed tomographic image might contain several gray levels, which, depending on the accuracy of the reconstruction, result from the fact that low-resolution pixels may cover different numbers of black high-resolution pixels. For turning the grayscale image into a higher-resolution binary image we utilize the gray levels and two additional higher-resolution projections, which may have been acquired by different imaging techniques or modalities (e.g., via scanning transmission electron microscopy [34]). More precisely, we study the task of reconstructing binary $m \times n$-images from row and column sums and additional constraints, so-called *block constraints*, on the number of black pixels to be contained in the $k \times k$-blocks resulting from a subdivision of each pixel in the $m/k \times n/k$ low-resolution image. We remark that we do not require that the X-ray data is taken from orthogonal directions. In our context it suffices that the X-ray data has been taken with double-resolution according to the discretization of the lower-resolution image.

Figure 1 illustrates the process. The example given in this figure also shows that the block-constraints can help to narrow down the solution space. In fact, the solution shown in this figure is uniquely determined by the input (the row and column sums and block constraints). The row and column sums alone do not determine the solution uniquely (and, of course, neither do the block constraints).

Apart from super-resolution imaging [27, 31, 22, 26, 35], and its particular applications [15, 36, 7, 29] in discrete tomography [25, 33, 23, 21, 12, 17, 1, 19, 20], the tasks discussed in this paper are also relevant in other contexts.

From a combinatorial point of view, they can be viewed as reconstructing binary matrices from given row and column sums and some additional constraints. Unexpected complexity jumps based on the results of the present paper are discussed in [5]. Background information on such problems involving different kinds of additional constraints can be found in [9, Sect. 4].

Other applications belong to the realm of *dynamic discrete tomography* [6, 37, 4, 5]. For instance, in plasma particle tracking, some particles are reconstructed at time t. Between t and the next time step $t + 1$ the particles may have moved to other positions. The task is to reconstruct their new

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.

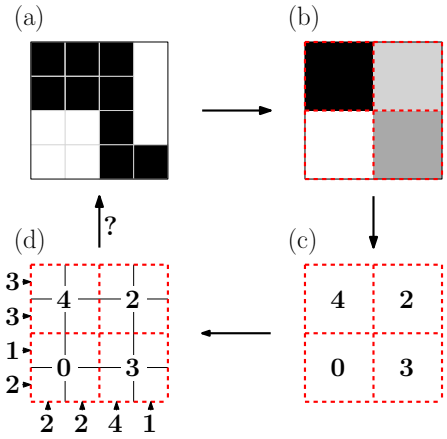


FIGURE 1. The double resolution imaging task DR. (a) Original (unknown) high resolution image, (b) the corresponding half-resolution grayscale image, (c) gray levels converted into block constraints, (d) taken in combination with double-resolution row and column sum data. The task is to reconstruct from (d) the original binary image shown in (a).

positions, again from few projections. One way of incorporating additional prior knowledge about the movement leads to block constraints of the kind discussed in this paper; see [4].

As the task of reconstructing a binary matrix from its row and column sums can be formulated as the task of finding a b-matching in a bipartite graph, we remark that our results can also be seen as results on finding b-matchings that are subject to specific additional constraints. For related (but intrinsically quite different) results on matchings subject to so-called budget constraints; see, e.g., [16].

The two main contributions of this paper are as follows. On the one hand, we show that reliable bimodal information can be utilized efficiently, i.e., the resolution can be doubled in polynomial time if the gray levels of the low resolution image have been determined precisely (Theorem 1). On the other hand, the task becomes already intractable (unless $\mathbb{P} = \mathbb{NP}$) if the gray levels of some pixels come with some small error of ± 1 (Theorem 3 and Corollary 1). This proves that noise does not only affect the quality of the reconstructed image but also the algorithmic tractability of the inverse problem itself. Hence the possibility of compensating noisy imaging by including bimodal information may in practice be jeopardized by its algorithmic complexity.

2. Notation and Main Results

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 $k\mathbb{N}_0 := \{ki : i \in \mathbb{N}_0\}$, $[k] := \{1, \dots, k\}$, and $[k]_0 := \{0, \dots, k\}$. With $\mathbf{1}$ we denote the all-ones vector of the corresponding dimension. The cardinality of a finite set $F \subseteq \mathbb{Z}^d$ is denoted by $|F|$.

In this paper we use Cartesian coordinates (rather than matrix notation) to represent pixels in an image. In particular, the two numbers i and j in a pair (i, j) denote the x- and y-coordinate of a point, respectively. In the following we consider grids $[m] \times [n]$. The set $\{i\} \times [n]$ and $[m] \times \{j\}$ is called *column* i and *row* j, respectively. Let, in the following $k \in \mathbb{N}$. Any set $\{(i-1)k+1, \dots, ik\} \times [n]$ is a *vertical strip (of width k)*. A *horizontal strip (of width k)* is a set of the form $[m] \times \{(j-1)k+1, \dots, jk\}$.

Sets 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*. For $i, j \in \mathbb{N}$ let $B_k(i, j) := B(i, j) := (i, j) + [k-1]_0^2$. 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 $B_k(i, j)$ with $(i, j) \in C(m, n, k)$ a *block*. The blocks form a partition of $[m] \times [n]$, i.e., $\dot{\bigcup}_{(i,j) \in C(m,n,k)} B_k(i, j) = [m] \times [n]$.

For $\varepsilon, k \in \mathbb{N}_0$ with $k \geq 2$ we define the task of (noisy) super-resolution

NSR(k, ε)

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

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

$c_1, \ldots, c_m \in \mathbb{N}_0$, (column sum measurem.)

$R \subseteq C(m, n, k)$, (corner points of reliable gray value measurem.)

$v(i, j) \in [k^2]_0$, $(i, j) \in C(m, n, k)$, (gray value measurem.)

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

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

$$\sum_{q\in[n]} \xi_{p,q} = s_p, \quad p \in [m], \quad \text{(column sums)}$$

$$\sum_{(p,q)\in B_k(i,j)} \xi_{p,q} = v(i, j), \quad (i, j) \in R, \quad \text{(block constraints)}$$

$$\sum_{(p,q)\in B_k(i,j)} \xi_{p,q} \in v(i, j) + [-\varepsilon, \varepsilon], \quad (i, j) \in C(m, n, k) \setminus R, \quad \text{(noisy block constraints)}$$

or decide that no such solution exists.

The numbers $r_1, \ldots, r_n$ and $c_1, \ldots, c_m$ are the row and column sum measurements of the higher-resolution binary $m \times n$ image, $v(i, j) \in [k^2]_0$ corresponds to the gray value of the low-resolution $k \times k$-pixel at (i, j) of the low-resolution $m/k \times n/k$ grayscale image, and R is the set of low-resolution pixel locations for which we assume that the gray values have been determined reliably, i.e., without error. The number ε is an error bound for the remaining blocks. The task is to find a binary high-resolution image satisfying the row and column sums such that the number of black pixels in each block adds up to a gray value for the corresponding $k \times k$-pixel in the lower-resolution image that lies in the specified interval. Clearly, a necessary (and easily verified) condition for feasibility is that

$$\sum_{q\in[n]} r_q = \sum_{p\in[m]} c_p.$$

In the following we will assume without loss of generality that this is always the case.

Our special focus is on double-resolution imaging, i.e., on the case $k = 2$. For $\varepsilon > 0$ we define NDR(ε) =NSR$(2, \varepsilon)$. In the reliable situation, i.e., for $\varepsilon = 0$, we simply speak of *double-resolution* and set DR=NSR$(2, 0)$. (Then, of course, the set R can be omitted from the input.)

In the present paper, we show that the resolution of any reliable grayscale image can in fact be doubled in each dimension in polynomial-time if X-ray data is provided from two viewing angles at double resolution. In other words, we show

Theorem 1. DR $\in \mathbb{P}$.

Figures 2 and 3 illustrate the performance of the algorithm for DR applied to two phantoms. The results have been obtained within a fraction of a second on a standard PC.

In Figure 2, the original phantom is a binary 200×200 image of a crystalline sample taken from [2]. It is assumed that the half-resolution grayscale image shown in Figure 2(a) has been obtained by some imaging method and that double-resolution X-ray information in the two standard directions is also available. Application of the algorithm yields the binary image shown in Figure 2(c).

In Figure 3, the original phantom is a binary 100×100 image, for which the half-resolution grayscale image shown in Figure 3(a) is available. From this image and the row and column sums (counting the black pixels) of the original phantom the algorithm returns the binary image shown in Figure 3(b).

(a) (b) (c)

FIGURE 2. The algorithm for DR (from the proof of Theorem 1) applied to a phantom. (a) Low-resolution 100×100 grayscale image, (b) magnification of the part contained in the red rectangle, (c) magnification of the corresponding double-resolution image obtained by the algorithm.

(a) (b) (c)

FIGURE 3. The algorithm for DR (from the proof of Theorem 1) applied to a phantom. (a) Low-resolution 50×50 grayscale image, (b) high-resolution 100×100 binary image obtained by the algorithm, (c) high-resolution 100×100 binary image minimizing the total variation.

The results shown in Figures 2 and 3 satisfy, of course, the row and column sums and block constraints. However, in both cases the solutions are still not unique. This can also be detected in polynomial time.

Theorem 2. *For every instance of* DR *it can be decided in polynomial time whether the instance admits a unique solution.*

While the additional X-ray information (at double resolution) does in general reduce the ambiguity, typically it will still not lead to uniqueness. Of course, a standard way of dealing with non-uniqueness in practice is regularization [7, 28, 8, 18]. To illustrate the possibility of adapting a regularization scheme to our context, we show in Figure 3(c) a binary solution minimizing the *total variation (TV).* In fact, the solution is obtained from Figure 3(a) by applying 15 so-called local switches, each of which are strictly decreasing the value of the total variation functional

$$J(x) = \sum_{p\in[m]} \sum_{q\in[n]} ||(\nabla x^{(1)}_{p,q}, \nabla x^{(2)}_{p,q})^T||_2,$$

where

$$(\nabla x)^{(1)}_{p,q} = \begin{cases} \xi_{p+1,q} - \xi_{p,q} & : \quad p < m, \\ 0 & : \quad \text{otherwise}, \end{cases}$$

$$(\nabla x)^{(2)}_{p,q} = \begin{cases} \xi_{p,q+1} - \xi_{p,q} & : \quad q < n, \\ 0 & : \quad \text{otherwise}, \end{cases}$$

and $||\cdot||_2$ denotes the Euclidean norm. See, e.g., [10] for some background information. Of course, we obtain the same results for other p-norms (including the 1-norm), because our images are binary.

We now turn to the task $n\mathrm{DR}(\varepsilon)$ where small "occasional" uncertainties in the gray levels are allowed. First observe that a constant number of uncertainties does not increase the complexity. However, if we allow a (as we will see, comparably small but) not constant number of uncertainties the problem becomes hard.

Theorem 3. $\textsc{n}\mathrm{DR}(\varepsilon)$ *is* $\mathbb{NP}$*-hard for any* $\varepsilon > 0$.

As it turns out, the problem remains $\mathbb{NP}$-hard for larger block sizes.

Corollary 1. $\textsc{n}\mathrm{SR}(k, \varepsilon)$ *is* $\mathbb{NP}$*-hard for any* $k \geq 2$ *and* $\varepsilon > 0$.

In other words, noise does not only affect the reconstruction quality. It also affects the algorithmic tractability of the inverse problem. (For background material on complexity theory, see, e.g., [14].)

The present paper is organized as follows. We deal with the case of reliable data in Section 3. Results on $\textsc{n}\mathrm{DR}(\varepsilon)$ and $\textsc{n}\mathrm{SR}(k, \varepsilon)$ involving noisy data are contained in Section 4. Section 5 concludes with some additional remarks on certain extensions and an open problem.

3. Reliable Data

In this section we discuss the task DR, i.e., $k = 2$ and $\varepsilon = 0$. We remark that, in the following, our emphasis is on providing brief and concise arguments for polynomial-time solvability rather than to focus on computationally or practically most efficient algorithms. Recall, however, our comments on the computational performance of our algorithms and the effect of the additional information for the example images depicted in Figures 2 and 3.

The main result of this section is Theorem 1. In its proof we show that DR decomposes into five problems that can be solved independently. The five problems are single-graylevel versions of DR where each block is required to contain the same number $\nu \in [4]_0$ of ones. Moreover, we allow for restricted problem instances (and in particular row and column sums) that are defined only on subsets

$$G(I) := \bigcup_{(i,j)\in I} B(i,j) \subseteq [m] \times [n]$$

of the grid $[m] \times [n]$ defined by means of some $I \subseteq C(m, n, 2)$. Let

$$\Pi_x(I) := \{i \in [m] : \exists j \in [n] : (i,j) \in I\}$$

and

$$\Pi_y(I) := \{j \in [n] : \exists i \in [n] : (i,j) \in I\}.$$

Then we define for $\nu \in [4]_0$:

DR(ν)

Instance: $m, n \in 2\mathbb{N}$,

$I \subseteq C(m,n,2)$, (a set of corner points)

$r_{j+l} \in \mathbb{N}_0$, $j \in \Pi_y(I),\ l \in \{0,1\}$ (row sum measurem.)

$c_{i+l} \in \mathbb{N}_0$, $i \in \Pi_x(I),\ l \in \{0,1\}$ (column sum measurem.)

Task: Find $\xi_{p,q} \in \{0,1\}$, $(p,q) \in G(I)$ with

$$\sum_{p:(p,j)\in G(I)} \xi_{p,j+l} = r_{j+l}, \qquad j \in \Pi_y(I),\ l \in \{0,1\} \qquad \text{(row sums)}$$

$$\sum_{q:(i,q)\in G(I)} \xi_{i+l,q} = c_{i+l}, \qquad i \in \Pi_x(I),\ l \in \{0,1\} \qquad \text{(column sums)}$$

$$\sum_{(p,q)\in B(i,j)} \xi_{p,q} = \nu, \qquad (i,j) \in I, \qquad \text{(block constraints)}$$

or decide that no such solution exists.

Of course, the tasks DR(ν), $\nu \in \{0,4\}$, are trivial. In fact, the only potential solution satisfying the block constraints is given by $\xi^*_{p,q} = \nu/4$, $(p,q) \in G(I)$, and it is checked easily and in polynomial time whether this satisfies the row and column sums.

The next lemma and corollary deal with DR(ν), $\nu \in \{1,3\}$. Its statement and proof uses the notation $\sigma_i(j)$ and $\rho_j(i)$ for the number of blocks in the same vertical strip as but below $B(i,j)$ or in the same horizontal strip as but left of $B(i,j)$, respectively. More precisely, let

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

for $(i,j) \in I \subseteq C(m,n,2)$.

Lemma 1.

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

$$r_j + r_{j+1} = \rho_j(m) \quad \text{and} \quad c_i + c_{i+1} = \sigma_i(n).$$

(ii) The solution of a feasible instance of DR(1) *is unique if, and only if, for every $(i,j) \in I$ we have*

$$r_j \cdot r_{j+1} = 0 \qquad \text{and} \qquad c_i \cdot c_{i+1} = 0.$$

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

Proof. Clearly, for the feasibility of a given instance of DR(1) the conditions

$$r_j + r_{j+1} = \rho_j(m) \quad \text{and} \quad c_i + c_{i+1} = \sigma_i(n),$$

are necessary.

Now, suppose that the conditions are satisfied. For every $(i,j) \in I$ and $(p,q) \in B(i,j)$ we set

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

Figure 4 gives an illustration.

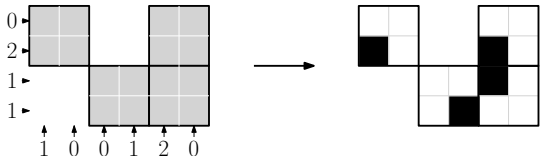


FIGURE 4. Illustration of DR(1). (Left) Row and column sums and blocks $B(i,j)$ with $(i,j) \in I$ in gray color. (Right) Solution defined by (3.1).

By this definition we have satisfied all block constraints. A simple counting argument shows that the row and columns sums are also as required. In fact, for $j \in \Pi_y(I)$ and $i \in \Pi_x(I)$ we have

$$\begin{aligned}
\sum_{p:(p,j)\in G(I)} \xi^*_{p,j} &= |\{p : ((p,j) \in I) \wedge (\rho_j(p) \le r_j)\}| &= r_j, \\
\sum_{p:(p,j)\in G(I)} \xi^*_{p,j+1} &= |\{p : ((p,j) \in I) \wedge (\rho_j(p) > r_j)\}| &= \rho_j(m) - r_j = r_{j+1}, \\
\sum_{q:(i,q)\in G(I)} \xi^*_{i,q} &= |\{q : ((i,q) \in I) \wedge (\sigma_i(q) \le c_i)\}| &= c_i, \\
\sum_{q:(i,q)\in G(I)} \xi^*_{i+1,q} &= |\{q : ((i,q) \in I) \wedge (\sigma_i(q) > c_i)\}| &= \sigma_i(n) - c_i = c_{i+1}.
\end{aligned} \tag{3.2}$$

This concludes the proof of (i).

To show (ii), we consider a solution x^* to DR(1). Suppose that $r_j > 0$ and $r_{j+1} > 0$ for some $j \in \Pi_y(I)$. Then, there exist $i, l \in [m]$ with $\xi^*_{i,j} = \xi^*_{l,j+1} = 1$. By the block constraints we have $\xi^*_{i,j+1} = \xi^*_{l,j} = 0$. Applying a switch to $\xi^*_{i,j}$, $\xi^*_{l,j+1}$, $\xi^*_{i,j+1}$, and $\xi^*_{l,j}$, i.e., setting $\overline{\xi}_{i,j} = \overline{\xi}_{l,j+1} = 0$, $\overline{\xi}_{i,j+1} = \overline{\xi}_{l,j} = 1$, and $\overline{\xi}_{p,q} = \xi^*_{p,q}$ otherwise, yields thus another solution. As the same argument holds also for the column sums we have shown that the condition in (ii) is necessary. The condition is, on the other hand, also sufficient since the 0-values of the row and column sums leave only a single position (p,q) in each block $B(i,j)$, $(i,j) \in I$, for which $\xi_{p,q}$ can take on a non-zero value (in fact, the value needs to be 1 to satisfy the block constraints).

We now turn to (iii). The algorithm checks the condition (i). If it is violated, it reports infeasibility; otherwise, a solution is constructed through (3.1) Clearly, all steps can be performed in polynomial time. □

Corollary 2.

(i) An instance $\mathcal{I}$ of DR(3) *is feasible if, and only if, for every $(i,j) \in I$ we have*

$$r_j + r_{j+1} = 3\rho_j(m) \quad \text{and} \quad c_i + c_{i+1} = 3\sigma_i(n).$$

(ii) The solution of a feasible instance of DR(3) *is unique if, and only if, for every $(i,j) \in I$ we have*

$$(2\rho_j(m) - r_j) \cdot (2\rho_j(m) - r_{j+1}) = 0 \qquad \text{and} \qquad (2\sigma_i(n) - c_i) \cdot (2\sigma_i(n) - c_{i+1}) = 0.$$

(iii) DR(3) $\in \mathbb{P}$.

Proof. The results follow directly from the results for DR(1) in Lemma 1 as the former are obtained from the latter by inversion and vice versa, i.e., by replacing r_{j+l} by $2\rho_j(m) - r_{j+l}$ and c_{i+l} by $2\sigma_i(n) - c_{i+l}$, $(i,j) \in I$, $l \in \{0,1\}$, and, in the solutions, all 1's by 0's and 0's by 1's. □

Next we prove polynomial-time solvability of DR(2). To this end we need two lemmas. The first can be viewed as dealing with a certain "two-color version" of discrete tomography. The second deals with specific switches or interchanges (in the sense of [30]).

Lemma 2. *The problem, given $m, n \in 2\mathbb{N}$, $I \subseteq C(m,n,2)$, and $r_j, c_i \in \mathbb{N}_0$, $(i,j) \in I$, decide whether there exist non-negative integers $\zeta_{i,j}$, $\eta_{i,j}$, $(i,j) \in I$, such that*

$$\begin{aligned} \sum_{p:(p,j)\in I} \zeta_{p,j} &= r_j, \qquad j \in \Pi_y(I), \\ \sum_{q:(i,q)\in I} \eta_{i,q} &= c_i, \qquad i \in \Pi_x(I), \\ \zeta_{i,j} + \eta_{i,j} &\leq 1, \qquad (i,j) \in I, \end{aligned}$$

and, if so, determine a solution, can be solved in polynomial time.

Proof. We assemble the variables $\zeta_{i,j}$, $\eta_{i,j}$ for $(i,j) \in I$ into an $|I|$-dimensional vector and rephrase the constraints in matrix form $Ax = r$, $A'x = c$, and $(E\,E)x \leq \mathbb{1}$, where r contains the r_j's, c contains the c_i's, and E denotes the $|I| \times |I|$-identity matrix. The problem is then to determine an integer solution of $\{x : Mx \leq b,\ x \geq 0\}$, where

$$M := \begin{pmatrix} A & 0 \\ -A & 0 \\ 0 & A' \\ 0 & -A' \\ E & E \end{pmatrix} \quad \text{and} \quad b := \begin{pmatrix} r \\ -r \\ s \\ -s \\ \mathbb{1} \end{pmatrix}.$$

The submatrix

$$M' := \begin{pmatrix} A & 0 \\ 0 & A' \\ E & E \end{pmatrix}$$

is totally unimodular as it is the node-edge incidence matrix of a bipartite graph (one of the two parts of the partition is the set of the last $|I|$ rows of M'). But then M is also totally unimodular as it results from appending a subset of rows of $-M'$ to the totally unimodular matrix M'. In other words, $\{x : Mx \leq b,\ x \geq 0\}$ is an integral polyhedron, i.e., it coincides with the convex hull of its integral vectors. Hence, a vertex of this polyhedron can be found in polynomial time by linear programming. See [32, Sect. 16&19] for background material. □

In a solution of DR we can have 16 different block types. They are shown in Figure 5. Several of them have the same label as they cannot be distinguished by the row sums.

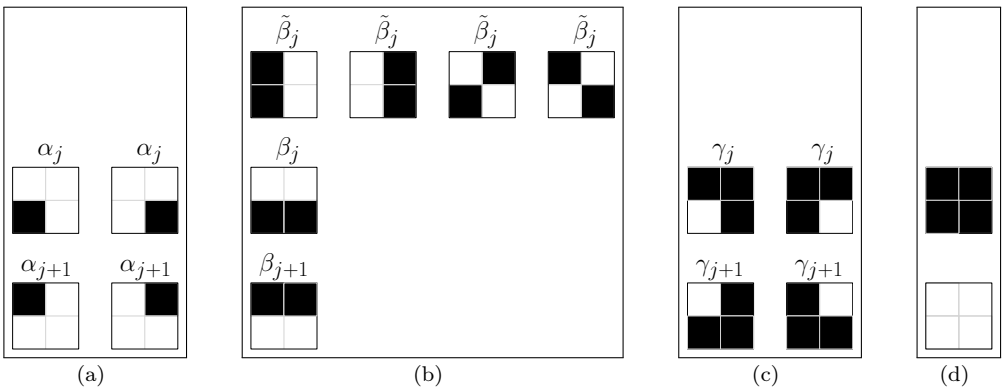


FIGURE 5. Block types (value 1 and 0 indicated in black and white, respectively). The labels correspond to the variables that count the number of occurrences of the respective block type in the horizontal strip $[m] \times \{j, j+1\}$. The block types in (d) are not further considered as they can be eliminated in a preprocessing step.

We introduce now the concept of local switches. For this we remark that if we consider a horizontal strip with a block labeled α_j and a block labeled β_{j+1} (labeling as in Figure 5), we can perform a switch that turns the blocks into blocks labeled α_{j+1} and $\tilde{\beta}_j$, respectively. We call such a switch a

horizontal local switch of class (1). In a similar way we define the *horizontal local switches of class* (2)-(7); see Table 1, column 1-3, and column 4 for an illustration. Note that the illustration does not show all horizontal local switches but rather gives a typical example.

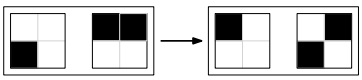
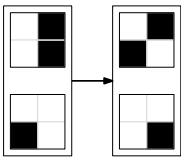
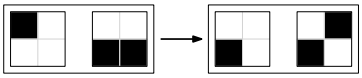
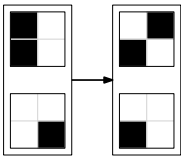
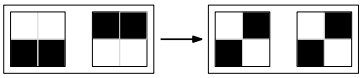
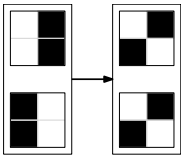
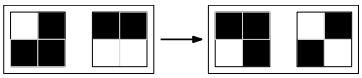
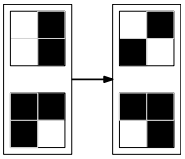
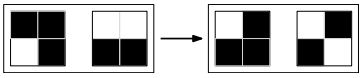
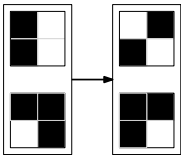
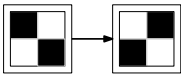
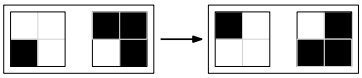
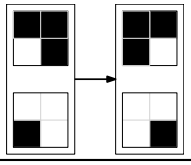
Class	turns	into	Illustration (horizontal local switch)	Illustration (vertical local switch)
(1)	(α_j, β_{j+1})	$(\alpha_{j+1}, \tilde{\beta}_j)$		
(2)	(α_{j+1}, β_j)	$(\alpha_j, \tilde{\beta}_j)$		
(3)	(β_j, β_{j+1})	$(\tilde{\beta}_j, \tilde{\beta}_j)$		
(4)	$(\gamma_{j+1}, \beta_{j+1})$	$(\gamma_j, \tilde{\beta}_j)$		
(5)	(γ_j, β_j)	$(\gamma_{j+1}, \tilde{\beta}_j)$		
(6)	$\tilde{\beta}_j$	$\tilde{\beta}_j$		
(7)	(α_j, γ_j)	$(\alpha_{j+1}, \gamma_{j+1})$		

TABLE 1. Local switches.

Let φ denote the coordinate transformation

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

A switch in a solution x^* is called a *vertical local switch of class* (t), $t \in [7]$, if it is a horizontal local switch of class (t) in $\varphi(x^*)$. See Table 1, column 5, for an illustration.

A switch is called *local* if it is a horizontal or vertical local switch of some class (t), $t \in [7]$. A solution is called *reduced* if no local switch of any class (t), $t \in [7]$, can be applied.

Lemma 3.

(i) The set of solutions of a given instance of DR *is invariant under local switches.*

(ii) If an instance of DR *has a solution, then there is also a reduced solution.*

Proof. To prove (i) just observe that the local switches do neither change the row and column sums nor the number of ones contained in each block.

Let us now turn to (ii). Suppose the instance has a solution. The local switches of class (t), $t \in [6]$, increase the number of blocks $B(i,j)$ with

$$\xi^*_{i,j} = \xi^*_{i+1,j+1} = 1 \qquad \text{and} \qquad \xi^*_{i+1,j} = \xi^*_{i,j+1} = 0.$$

The process of applying local switches of class (t), $t \in [6]$, thus needs to terminate since the number of blocks in each problem instance is finite. Further, note that local switches of class (7) applied in horizontal strips increase the number of blocks labeled α_{j+1}, the vertically applied switches of that class do not decrease the number of such blocks, but increase the number of blocks containing at least a one in the rightmost column of the block. □

Lemma 4. $\mathrm{DR}(2) \in \mathbb{P}$.

Proof. Our general strategy is as follows. First we show that a given instance is feasible if, and only if, there exists a specific reduced solution that contains only three block types. Then we give a polynomial-time algorithm that finds such a solution or determines infeasibility.

In the following we will again use the notational convention that specific settings of variables and parameters are signified by a superscript *.

By possibly rearranging the rows and columns, we assume that

$$r_j \geq r_{j+1} \qquad \text{and} \qquad c_i \geq c_{i+1}, \quad (i,j) \in I.$$

Now, suppose that the given instance has a solution. By Lemma 3 there exists a reduced solution. In the following, we consider such a reduced solution.

The potential block types, and corresponding variables β_j, β_{j+1}, and $\tilde{\beta}_j$ counting the number of occurrences of the respective block types in the strip $[m] \times \{j, j+1\}$ of a reduced solution, are shown in Figure 5(b). Block types counted by the same variable have equal row sums.

Consider now for fixed $(i,j) \in I$ the blocks in $G(I) \cap ([m] \times \{j, j+1\})$. Counting the block types, any solution needs to satisfy

$$\begin{array}{rcrcl} 2\beta_j & & + & \tilde{\beta}_j & = & r_j, \\ & 2\beta_{j+1} & + & \tilde{\beta}_j & = & r_{j+1}, \end{array}$$

which implies

$$\beta_j - \beta_{j+1} = (r_j - r_{j+1})/2. \tag{3.3}$$

As the solution is (3)-reduced we need to have $\beta^*_j = 0$ or $\beta^*_{j+1} = 0$. This, together with $r_j \geq r_{j+1}$ and (3.3), implies

$$\beta^*_{j+1} = 0, \qquad \beta^*_j = (r_j - r_{j+1})/2, \quad \text{and} \quad \tilde{\beta}^*_j = r_{j+1}.$$

The same argument applies to the other horizontal strips, and a similar argument holds for the vertical strips. Hence, we conclude that DR(2) amounts to the task of assigning one of the three block types shown in Figure 6 to each of the blocks $B(i,j)$, $(i,j) \in I$, in such a way that the number of type 1 and type 2 blocks in each horizontal and vertical strip, respectively, equals some prescribed numbers.

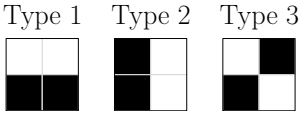


FIGURE 6. The three potential block types in solutions of DR(2) instances considered in the proof of Lemma 4. Value 1 and 0 are indicated in black and white, respectively.

Now, we introduce 0/1-variables $\zeta_{i,j}$, $\eta_{i,j}$, $(i,j) \in I$, that should be interpreted as follows: $\zeta^*_{i,j} = 1$ and $\eta^*_{i,j} = 0$ if, and only if, $B(i,j)$ is of type 1; $\zeta^*_{i,j} = 0$ and $\eta^*_{i,j} = 1$ if, and only if, $B(i,j)$ is of type 2, and $\zeta^*_{i,j} = \eta^*_{i,j} = 0$ if, and only if, $B(i,j)$ is of type 3. Then we ask for 0/1-solutions that satisfy

$$\begin{aligned}
\sum_{p:(p,j)\in I} \zeta_{p,j} &= (r_j - r_{j+1})/2, \qquad (i,j) \in I,\\
\sum_{q:(i,q)\in I} \eta_{i,q} &= (c_i - c_{i+1})/2, \qquad (i,j) \in I,\\
\zeta_{i,j} + \eta_{i,j} &\leq 1.
\end{aligned}$$

Such solutions can be found in polynomial time by Lemma 2. □

Now with these preliminary results we can turn to the proof of Theorem 1.

Proof of Theorem 1. We show that the task of finding a solution to a given problem instance can be decomposed into five independent subtasks of solving instances of DR(ν), $\nu \in \{0,\dots,4\}$. The subtasks are then polynomial time solvable by Lemma 1, Corollary 2 and 4 (and trivially for DR(0) and DR(4)). The key step in our proof is to deduce from the data the row and column sums for the particular subtasks. This deduction is possible for reduced solutions. Their existence for feasible problem instances is guaranteed by Lemma 3.

Further, we will, if needed apply some preprocessing. First, if the problem instance contains block constraints of the form $v(i,j) = \nu$, $\nu \in \{0,4\}$, we need to have $\xi^*_{i,j} = \xi^*_{i+1,j} = \xi^*_{i,j+1} = \xi^*_{i+1,j+1} = \nu/4$. In this case, we fix the values of these variables, reduce the row and column-sums accordingly, and consider only the remaining blocks. For notational convenience, we assume from now on that our instance does not contain such block constraints.

Second, by possibly rearranging the rows and columns, we assume that

$$r_j \geq r_{j+1} \qquad \text{and} \qquad c_i \geq c_{i+1}, \quad (i,j) \in C(m,n,2). \tag{3.4}$$

Suppose now that the problem instance has a solution.

Considering a horizontal strip $[m] \times \{j, j+1\}$ with $j \in [n] \cap (2\mathbb{N}_0 + 1)$, we compute

$$\begin{aligned}
a_j &:= |\{i \in [m] \cap (2\mathbb{N}_0 + 1) : v(i,j) = 1\}|,\\
b_j &:= |\{i \in [m] \cap (2\mathbb{N}_0 + 1) : v(i,j) = 2\}|,\\
c_j &:= |\{i \in [m] \cap (2\mathbb{N}_0 + 1) : v(i,j) = 3\}|.
\end{aligned}$$

A count of the block types of the solution in that strip (the variables for the block types are indicated in Figure 5), yields

$$\begin{aligned}
\alpha_j \quad &+ 2\beta_j \quad + \tilde{\beta}_j + \gamma_j + 2\gamma_{j+1} = r_j, \\
\alpha_{j+1} + \quad &2\beta_{j+1} + \tilde{\beta}_j + 2\gamma_j + \gamma_{j+1} = r_{j+1}, \\
\alpha_j + \alpha_{j+1} \quad &= a_j, \qquad (3.5)\\
\beta_j + \beta_{j+1} + \tilde{\beta}_j \quad &= b_j, \qquad (3.6)\\
\gamma_j + \gamma_{j+1} &= c_j. \qquad (3.7)
\end{aligned}$$

From the system we obtain

$$\alpha_j + \beta_j - \beta_{j+1} + \gamma_{j+1} = r_j - b_j - c_j, \tag{3.8}$$
$$-\alpha_j - \beta_j + \beta_{j+1} + \gamma_j = r_{j+1} - a_j - b_j - c_j. \tag{3.9}$$

By Lemma 3 there is a reduced solution; the corresponding variable setting will be signified by the superscript *. Suppose now in such a reduced solution we have $\beta^*_{j+1} > 0$. As we cannot apply any further local switch of class (t), $t \in \{1,3,4\}$, we need to have $\alpha^*_j = \beta^*_j = \gamma^*_{j+1} = 0$, hence by (3.5) and (3.7) $\alpha^*_{j+1} = a_j$, and $\gamma^*_j = c_j$. Using (3.8) and (3.9) we obtain

$$\begin{aligned} r_j &= b_j + c_j - \beta^*_{j+1}, \\ r_{j+1} &= a_j + b_j + 2c_j + \beta^*_{j+1}, \end{aligned}$$

which imply $r_j < b_j + c_j \le r_{j+1}$, a contradiction to (3.4). Hence, we have $\beta^*_{j+1} = 0$.

In the following we distinguish three cases. First, suppose that in every reduced solution we have $\beta_j > 0$. As we cannot apply any further local switch of class (t), $t \in \{2,5\}$, we must have $\alpha^*_{j+1} = \gamma^*_j = 0$, hence $\alpha^*_j = a_j$, and $\gamma^*_{j+1} = c_j$. Eq. (3.9) implies $\beta^*_j = b_j + c_j - r_{j+1}$. As $0 < \beta^*_j \le b_j$, we thus have $c_j \le r_{j+1} < b_j + c_j$. Further, all values of the variables are determined since (3.6) yields $\tilde{\beta}^*_j = r_{j+1} - c_j$.

For the second case, suppose that there is a reduced solution with $\beta^*_{j+1} = \beta^*_j = 0$, and in any such solution we have $\alpha_j > 0$. Then, $\tilde{\beta}^*_j = b_j$, and as we cannot apply any further local switch of class (7), we need to have $\gamma^*_j = 0$, hence $\gamma^*_{j+1} = c_j$. Eq. (3.9) implies $\alpha^*_j = a_j + b_j + c_j - r_{j+1}$. As $0 < \alpha^*_j \le a_j$, we thus have $b_j + c_j \le r_{j+1} < a_j + b_j + c_j$. Further, all values of the variables are determined since (3.5) yields $\alpha^*_{j+1} = r_{j+1} - b_j - c_j$.

Finally, for the third case, suppose that there is a solution with $\beta^*_{j+1} = \beta^*_j = \alpha^*_j = 0$. Then $\tilde{\beta}^*_j = b_j$ and $\alpha^*_{j+1} = a_j$. Eq. (3.9) implies $\gamma^*_j = r_{j+1} - a_j - b_j - c_j$. As $0 \le \gamma^*_j \le c_j$ we thus have $a_j + b_j + c_j \le r_{j+1} \le a_j + b_j + 2c_j$. Further, all values of the variables are determined since (3.7) yields $\gamma^*_{j+1} = a_j + b_j + 2c_j - r_{j+1}$.

Since, by assumption, the given instance has a solution and, hence a reduced one, exactly one of the three cases above must occur. Which one it is, can be determined from the data, simply by testing whether we have

$$\begin{array}{lcccll} c_j & \le & r_{j+1} & < & b_j + c_j, & \\ b_j + c_j & \le & r_{j+1} & < & a_j + b_j + c_j, & \text{or} \\ a_j + b_j + c_j & \le & r_{j+1} & \le & a_j + b_j + 2c_j. & \end{array} \tag{3.10}$$

In all three cases we know the particular values of the α^*_j, α^*_{j+1}, β^*_j, β^*_{j+1}, $\tilde{\beta}^*_j$, γ^*_j, and γ^*_{j+1}. Therefore we know for each $\nu \in [3]$ the row sums $\sum_{p:(p,j)\in I(\nu)} \xi_{p,j}$ and $\sum_{p:(p,j)\in I(\nu)} \xi_{p,j+1}$, where $I(\nu)$ denotes the set

$$I(\nu) := \bigcup_{\substack{(a,b)\in C(m,n,2) \\ \wedge\, v(a,b)=\nu}} B(a,b).$$

In particular, we have

$$\sum_{p:(p,j)\in I(1)} \xi_{p,j} = \alpha^*_j, \qquad \sum_{p:(p,j)\in I(2)} \xi_{p,j} = 2\beta^*_j + \tilde{\beta}^*_j, \qquad \sum_{p:(p,j)\in I(3)} \xi_{p,j} = \gamma^*_j + 2\gamma^*_{j+1},$$
$$\sum_{p:(p,j)\in I(1)} \xi_{p,j+1} = \alpha^*_{j+1}, \qquad \sum_{p:(p,j)\in I(2)} \xi_{p,j+1} = \tilde{\beta}^*_j, \qquad \sum_{p:(p,j)\in I(3)} \xi_{p,j+1} = 2\gamma^*_j + \gamma^*_{j+1}.$$

A similar argument applies to the vertical strips. We can therefore decompose the reconstruction problem into the three independent subproblems DR(ν), $\nu \in [3]$, which, by Lemma 1, Corollary 2, and Lemma 4, are solvable in polynomial time. □

Note that the above proof is constructive, fully detailing the steps of a polynomial-time algorithm for solving DR.

We also remark that the solutions returned by the proposed algorithm are always reduced solutions. This is easily seen by (i) verifying that the values α_j^*, α_{j+1}^*, β_j^*, β_{j+1}^*, $\tilde{\beta}_j^*$, γ_j^*, and γ_{j+1}^*, $j \in [n]$, in each of the three cases in (3.10) (and correspondingly for the variables in the vertical strips) do not allow an application of any of the local switches of class (t), $t \in [7] \setminus \{6\}$, because for each of the local switches there is always a corresponding variable of value zero; and (ii) by noting that the proposed algorithm for DR(2) returns, by definition, only solutions where no local switch of class (6) can be applied.

Now we turn to uniqueness.

Proof of Theorem 2. By Theorem 1, infeasibility is detected or a solution of an instance is determined in polynomial time. We can thus assume that the problem instance has a solution.

Clearly, a solution is unique if, and only if (i) there is only one reduced solution, and (ii) every solution is a reduced solution.

The solution returned by the algorithm is a reduced solution. Any two reduced solutions define the same instances of the subproblems DR(ν), $\nu \in [3]$, because the problem instances are defined by means of (3.10), i.e., the definition depends only on the particular values of the $r_1, \ldots, r_n$, and $c_1, \ldots, c_m$. Hence, there is only one reduced solution if, and only if, the solution to each of the subproblems DR(ν), $\nu \in [3]$, is unique.

The conditions in Lemma 1(ii) and Corollary 2(ii), and hence uniqueness for the instances of DR(1) and DR(3), can be checked in polynomial time. For DR(2) we have by Lemma 2, a vertex solution v^* of the linear program at our disposal. Now we minimize the linear objective function $f(x) = x^T v^*$ over the same feasible region, which again can be done in polynomial time. The solution v^* is unique if, and only if, $f(x^*) = {v^*}^T v^*$; hence uniqueness for DR(2), and therefore (i), can be checked in polynomial time.

Suppose now that the reduced solution x^* returned by the algorithm is unique among all reduced solutions, i.e., (i) is satisfied. We have shown in Lemma 3 that every solution can be reduced by applying a sequence of local switches. Reversing the sequence and the local switches (the *reversed local switch* of the local switch that changes $(\xi_{p,q}^*, \xi_{a,q}^*, \xi_{p,b}^*, \xi_{a,b}^*)$ into $\mathbf{1} - (\xi_{p,q}^*, \xi_{a,q}^*, \xi_{p,b}^*, \xi_{a,b}^*)$ is the switch that reverts these changes), we thus see that (ii) holds if, and only if, there is no reversed local switch that can be applied to x^*. There are $O(m^2n^2)$ possible tuples $(\xi_{p,q}^*, \xi_{a,q}^*, \xi_{p,b}^*, \xi_{a,b}^*)$ that need to be checked whether they form a reversed local switch, hence (ii) can also be checked in polynomial time. □

4. Data Uncertainty

In the proof of Theorem 3 we use a transformation from the following $\mathbb{NP}$-hard problem (see [11])

1-In-3-SAT

Instance: Positive integers S, T, and a set $\mathcal{C}$ of S clauses over T variables $\tau_1, \ldots, \tau_T$, where each clause consists of three literals involving three different variables.

Task: Find a satisfying truth assignment for $\mathcal{C}$ that sets exactly one literal true in each clause.

For a given instance of 1-In-3-SAT we will construct a circuit board that contains an initializer, several connectors, and clause chips. A truth assignment is transmitted through the circuit board, the clause chips ensure that the clauses are satisfied.

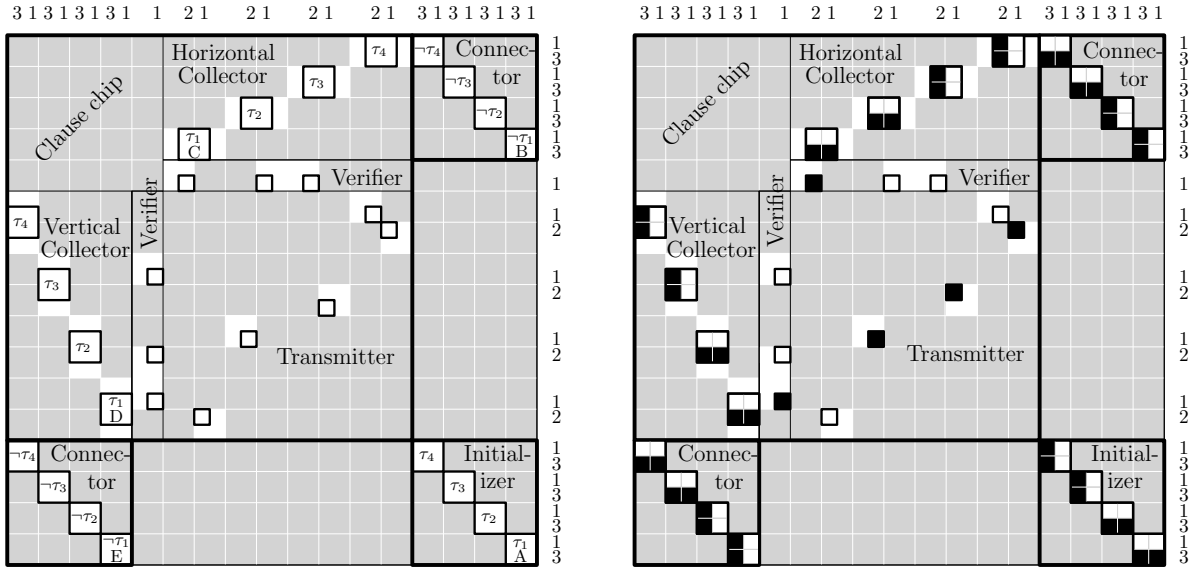
The structure of our proof is in some ways similar to the proof from [13] that establishes the $\mathbb{NP}$-hardness of the task of reconstructing lattice sets from X-rays taken in three or more directions. However, we need to deviate from [13] in several key aspects, because the block constraints do not give a direct way of controlling points over large distances, and the elimination of "unwanted solutions" inside individual blocks seems also problematic. A fundamental difference to [13] is, for instance, that we encode the Boolean values for the variables by specific types of blocks, and the satisfiability of the clauses is verified via row and column sums. At first glance it seems that part of the variable assignment information

is lost after verification, but it turns out that we can recover it from "redundant" information in our encoding.

Key ideas of the proof of Theorem 3. Before we start with the detailed proof, we illustrate the general ideas by means of an example. Suppose, the instance $\mathcal{I}$ of 1-In-3-SAT is given by $\mathcal{I} := (S, T, \mathcal{C}) := (1, 4, \{\tau_1 \vee \neg\tau_2 \vee \tau_3\})$. We will define an instance $\mathcal{I}'$ of nDR(ε) such that there is a solution for $\mathcal{I}'$ if, and only if, there is one for $\mathcal{I}$. To this end, we construct a circuit board contained in the box $[34] \times [34]$. With each $(p, q) \in [34] \times [34]$ we associate a variable $\xi_{p,q}$.

The circuit board consists of three major types of components: An *initializer*, *connectors*, and *clause chips*. The initializer and the connectors are rather similar. The initializer contains for every variable τ_t, $t \in [T]$, a so-called τ_t*-chip*, the connectors contain for every variable a $\neg\tau_t$*-chip*. The clause chips are more complex as they consist of two *collectors*, two *verifiers*, and a *transmitter*.

Figure 7(a) shows the circuit board for our instance.



(a) (b)

FIGURE 7. Transformation from 1-In-3-SAT for the instance $\mathcal{I} := (S, T, \mathcal{C}) := (1, 4, \{\tau_1 \vee \neg\tau_2 \vee \tau_3\})$. (a) The circuit board. By setting to zero suitable row and column sums we make sure 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 solution $(\tau^*_1, \tau^*_2, \tau^*_3, \tau^*_4) = (\text{TRUE}, \text{TRUE}, \text{FALSE}, \text{FALSE})$ of $\mathcal{I}$.

The gray areas indicate blocks that are set to zero via block constraints. By setting to zero suitable row and column sums we make sure that non-zero components $\xi^*_{p,q}$ are only possible in the white bold-framed boxes. The non-zero row and column sums are shown on the top and on the right of the figure.

The bold-framed 2×2 boxes are *Boolean chips*, which are, according to their position on the circuit board, further classified into τ_t and $\neg\tau_t$-chips, $t \in [T]$. In these chips we will encode the respective Boolean values for the variables τ_t and $\neg\tau_t$, $t \in [T]$. All blocks, except for those set to zero, are allowed to contain at most two ones. The blocks containing the $\neg\tau_t$-chips are required to contain precisely two ones.

In each clause chip there are two strips playing a special role. One is a horizontal strip containing a so-called *horizontal verifier*, the other is a vertical strip containing a so-called *vertical verifier* (see rows 25 and 26, and columns 9 and 10 in Figure 7(a)). A key property of our construction is that the

truth assignments will be in 1-to-1 correspondence with the solutions of the NDR(ε) instance if the row and column sum constraints related to the verifiers are left unspecified. The row and column sums constraints related to the verifiers will ensure that the truth assignments are in fact satisfying truth assignments.

We can already begin to see how a truth assignment for our particular instance $\mathcal{I}$ will provide a solution for $\mathcal{I}'$. Utilizing the type 1 and type 2 blocks shown in Figure 8, we set each τ_t-chip, $t \in [T]$, of the initializer to type 1 if $\tau_t^* = \textsc{True}$, otherwise we set it to type 2.

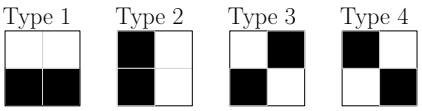


FIGURE 8. The four possible types of blocks for τ_t- and $\neg\tau_t$-chips, $t \in [T]$. In τ_t-chips, the types 1 and 2 represent the Boolean values TRUE and FALSE, respectively.

Let us first consider $t = 1$ and the sequence of Boolean chips marked A, B, C, D, and E. See also Figure 9. The chips A and B lie in the same vertical strip, and since A is of type 1, B needs to be of type 2 to satisfy the column sums in the vertical strip. Chip C needs to be of type 1 to satisfy the row sums in the horizontal strip. Then, the remaining points in the two vertical strips intersecting chip C are uniquely determined by the column sums. As chip D needs to contain two ones (since the vertical strip containing chip D contains four ones, two of which need to be contained in chip E via block constraints) we conclude from the row sums that D is of type 1. Considering now the vertical strip intersecting D we deduce that chip E is of type 2, and therefore, together with chip A, satisfies the row sums in the horizontal strip. Thus, the τ_1-chips are consistently of type 1, the $\neg\tau_1$-chips are of type 2. Note that we have satisfied all row and column sums for the strips intersecting one of the chips A,B,C,D, and E. Moreover, the block constraints in these chips are also satisfied.

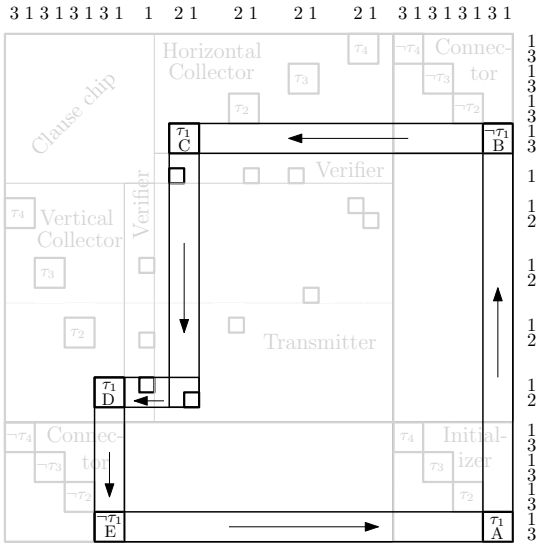


FIGURE 9. The chain of reasoning (vertical transmission) for deducing the types of the Boolean chips marked A, B, C, D, and E from Figure 7.

Turning to the Boolean chips for $t = 2$, we conclude with a similar reasoning that the τ_2- and $\neg\tau_2$-chips are consistently of type 1 and type 2, respectively. We have now satisfied the block constraints for the τ_2- and $\neg\tau_2$-chips, Additionally, we satisfied the row and column sums for the strips intersecting one of these chips. Clearly, we have not violated any of the previously satisfied constraints as there is no row or column that intersects both a Boolean chip for $t = 1$ and $t = 2$.

We say that we have deduced the types of the Boolean chips for $t \in \{1, 2\}$ by *vertical transmission*, because in our previous chain of arguments we started by considering the vertical strip containing the respective Boolean chip of the initializer (see also Figure 9).

Note that such a vertical transmission does not directly work for the assignment FALSE of the variables τ_3, τ_4 since the column sums do not allow us to deduce a unique solution for the respective $\neg\tau_t$-chip. Hence for the Boolean chips for $t \in \{3, 4\}$ we resort to *horizontal transmission* to deduce the types of the respective τ_t- and $\neg\tau_t$-chips. In this way we conclude that the τ_t- and $\neg\tau_t$-chips are consistently type 2 and type 1, respectively.

Figure 7(b) shows the solution that we obtain by the previous arguments. All constraints, in particular also the row and column sums in the verifiers, are satisfied. We will later see in general that there is a single one in each of the two verifiers of the s-clause chip, $s \in S$, if, and only if, exactly one of the three literals appearing in the s-th clause is TRUE.

Now, consider the reverse direction for the proof, i.e., suppose a solution x^* to our instance $\mathcal{I}'$ of Figure 7(a) is given. Note that if we can ensure that the types of the Boolean chips in the initializer are either 1 or 2 then by the previous arguments we have a satisfying truth assignment for $\mathcal{I}$ by setting $\tau_t^* :=$ TRUE, $t \in [T]$, if, and only if, the corresponding τ_t-chip of the initializer is of type 1.

The respective block, row and column sum constraints allow only the four possible types of blocks for the Boolean chips of the initializer shown in Figure 8. The types 3 and 4, however, are ruled out by the following "global" argument: Suppose, for some $t \in [T]$, the τ_t-chip of the initializer is of type $\ell \in \{3, 4\}$. By horizontal transmission, we then conclude that the $\neg\tau_t$-chips need to be of type 1 (by transmitting, as before in a unique way, through each clause chip). By vertical transmission, however, we conclude that the $\neg\tau_t$-chips are of type 2, which is a contradiction.

We remark that effectively we utilize the noisy block constraints at three different places in our construction. They are needed, because we can not prescribe the exact number of ones in the τ_t-chips of the horizontal and vertical collectors, and in the (s, t)-configurations that are introduced later (see also Figures 11, 12, and 13).

Having illustrated the key ideas behind the construction, we now turn to the formal proof.

Proof of Theorem 3. Let in the following $\mathcal{I} := (S, T, \mathcal{C})$ denote an instance of 1-IN-3-SAT. We will define an instance $\mathcal{I}'$ of NDR(ε) such that there is a solution for $\mathcal{I}'$ if, and only if, there is one for $\mathcal{I}$. The circuit board will be contained in the box

$$[m] \times [n] := [S(6T + 2) + 2T]^2.$$

With each $(p, q) \in [m] \times [n]$ we associate a variable $\xi_{p,q}$. Figure 10 illustrates the general layout of the construction.

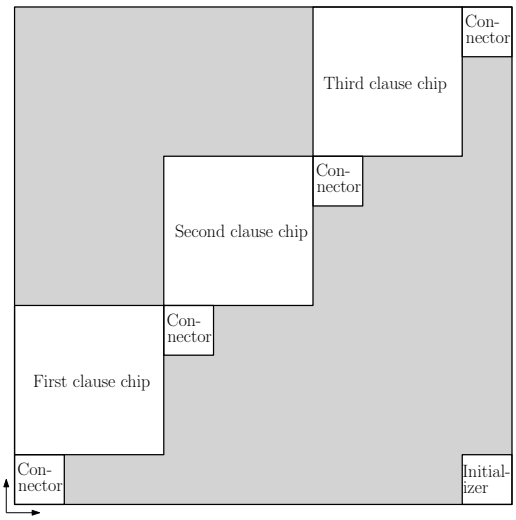


FIGURE 10. The general layout of the circuit board (here for $S = 3$). Non-zero components $\xi_{p,q}^*$ of a solution will be possible only for those (p, q) contained in the white area.

We define the instance $\mathcal{I}'$ of NDR(ε) by specifying the row sums $r_1, \ldots, r_n$, the column sums $c_1, \ldots, c_m$, and the block constraints. For notational convenience we define the block constraints via a function $f : C(m, n, 2) \to \{(=, 0), (=, 2), (\approx, 1)\}$. In this notation, $f(i, j) = (=, \nu)$, $\nu \in \{0, 2\}$, denotes the block constraint

$$\sum_{(p,q)\in B(i,j)} \xi_{p,q} = \nu,$$

while $f(i, j) = (\approx, 1)$ signifies the block constraint

$$\sum_{(p,q)\in B(i,j)} \xi_{p,q} \in 1 + ([-\varepsilon, \varepsilon] \cap \mathbb{Z}).$$

The set R of reliable block constraints for $\mathcal{I}'$ is then $R := \{(i, j) \in C(m, n, 2) \,:\, f(i, j) \neq (\approx, 1)\}$.

The different components of the circuit board are placed at specific positions. For a more compact definition of these positions we set

$$a_s := (6T + 2)s + 1, \qquad s \in [S]_0.$$

Turning to the definition of the initializer, we remark that the initializer is contained inside the box $(a_S, 1) + [2(T - 1)]_0^2$. We will specify the row and column sums when the corresponding connectors are introduced. In terms of the block constraints we define for every $(u, v) \in [T - 1]_0^2$:

$$f(a_S + 2u, 1 + 2v) := \begin{cases} (\approx, 1) & : \quad u + v = T - 1, \\ (=, 0) & : \quad \text{otherwise.} \end{cases}$$

The block $B(a_S + 2(T - t), 2t - 1)$, $t \in [T]$, will be called *τ_t-chip (of the initializer)*. For an illustration of an initializer, see the bottom right $[8] \times [8]$-box in Figure 7(a).

We define $S+1$ connectors. Each connector is contained inside a box $(a_s, a_s)+[2(T-1)]_0^2$ with $s \in [S]_0$. In terms of block constrains we define for every $s \in [S]_0$ and $(u, v) \in [T - 1]_0^2$:

$$f(a_s + 2u, a_s + 2v) := \begin{cases} (=, 2) & : \quad u + v = T - 1, \\ (=, 0) & : \quad \text{otherwise.} \end{cases}$$

For the row and column sums we define for every $s \in [S]_0$ and $l \in [2(T - 1)]_0$:

$$r_{a_s+l} := c_{a_s+l} := \begin{cases} 3 & : \quad l \in 2\mathbb{N}_0, \\ 1 & : \quad \text{otherwise.} \end{cases}$$

The block $B(a_s+2(T-t), a_s+2(t-1))$, $a_s \in [S]_0$, $t \in [T]$, will be called *$\neg\tau_t$-chip (of the s-th connector)*. For an illustration of a connector, see the bottom left $[8] \times [8]$-box in Figure 7(a).

Next we define the S clause chips. Each clause chip is contained in a box $(a_s, a_s + 2T) + [6T + 1]_0^2$ with $s \in [S - 1]_0$. The box $(a_s, a_s + 2T) + [2T - 1]_0 \times [4T - 1]_0$ is a *(vertical) collector*, the box $(a_s + 2T, a_s + 2T) + [1]_0 \times [4T - 1]_0$ is a *(vertical) verifier*, the box $(a_s + 2T + 2, a_s + 2T) + [4T - 1]_0^2$ is a *transmitter*, the box $(a_s + 2T + 2, a_s + 6T) + [4T - 1]_0 \times [1]_0$ is a *(horizontal) verifier*, and the box $(a_s + 2T + 2, a_s + 6T + 2) + [4T - 1]_0 \times [2T - 1]_0$ is a *(horizontal) collector*. For an illustration of a clause chip, see the top left $[26] \times [26]$-box in Figure 7(a).

For $s \in [S - 1]_0$ we define the row and column sums for the verifiers (we will call these row and column sums *verifier sums*) by

$$\begin{aligned} r_{a_s+6T} \quad &:= c_{a_s+2T+1} := 1, \\ r_{a_s+6T+1} &:= c_{a_s+2T} \quad := 0. \end{aligned}$$

The remaining row and columns sums for the transmitters, i.e., for $s \in [S - 1]_0$ and $l \in [T - 1]_0$ are defined by

$$\begin{aligned} r_{a_s+2T+4l} \quad &:= c_{a_s+2T+4l+2} := 0, \\ r_{a_s+2T+4l+1} &:= c_{a_s+2T+4l+3} := 2, \\ r_{a_s+2T+4l+2} &:= c_{a_s+2T+4l+4} := 1, \\ r_{a_s+2T+4l+3} &:= c_{a_s+2T+4l+5} := 0. \end{aligned}$$

Note that sums for the rows and columns meeting the collectors are already defined as there is a corresponding connector in the same row or, respectively, column (see also Figure 10). We have thus completed our definition of the row and column sums of our instance and continue now with the remaining block constraints.

For the vertical collector, we define for every $s \in [S-1]_0$ and $(u,v) \in [T-1]_0^2$:

$$f(a_s + 2u, a_s + 2T + 2v) := \begin{cases} (\approx, 1) & : \quad 2u + v \in \{2T-2, 2T-1\}, \\ (=, 0) & : \quad \text{otherwise.} \end{cases}$$

The box $(a_s + 2T - 2 - 2t, a_s + 2T + 1 + 4t) + [1]_0 \times [1]_0$, $t \in [T-1]_0$, will be called *τ_t-chip (of the $(s+1)$-th vertical collector).*

For the horizontal collector we define similarly for every $s \in [S-1]_0$ and $(u,v) \in [T-1]_0^2$:

$$f(a_s + 2T + 2 + 2u, a_s + 6T + 2 + 2v) := \begin{cases} (\approx, 1) & : \quad u - 2v \in \{0, 1\}, \\ (=, 0) & : \quad \text{otherwise.} \end{cases}$$

The box $(a_s + 2T + 3 + 4t, a_s + 6T + 2 + 2t) + [1]_0 \times [1]_0$, $t \in [T-1]_0$, will be called *τ_t-chip (of the $(s+1)$-th horizontal collector).*

Note that the τ_t-chips, $t \in [T]$, of the collectors are not contained in blocks. In fact, each such τ_t-chip intersects two blocks (which, in turn, are allowed to contain at most two ones). For any $t \in [T]$ we will also refer to the τ_t- and $\neg\tau_t$-chips on the circuit board as *Boolean chips.*

For the empty space above the vertical connector, i.e., for $s \in [S-1]_0$ and $(u,v) \in [T+1]_0^2$, we define

$$f(a_s + 2u, a_s + 6T + 2v) := (=, 0).$$

Now we turn to the block constraints for the clause-dependent part of chip, i.e., the verifiers and the transmitters.

With U_s and N_s, $s \in [S]$, we denote the indices of the variables that appear unnegated and, respectively, negated in the s-th clause. For instance, for $\tau_1 \vee \neg\tau_2 \vee \tau_3$ (regarded as the first clause) we have $U_1 = \{1, 3\}$ and $N_1 = \{2\}$.

For the vertical verifiers we define for every $s \in [S]$, $t \in [T]$, and $v \in [2]$:

$$f(a_{s-1} + 2T, a_{s-1} + 2T + 4t - 2v) := \begin{cases} (\approx, 1) & : \quad ((t \in U_s) \wedge (v = 1)) \vee ((t \in N_s) \wedge (v = 2)), \\ (=, 0) & : \quad \text{otherwise.} \end{cases}$$

Similarly, for the horizontal verifiers we define for every $s \in [S]$, $t \in [T]$, and $u \in [2]$:

$$f(a_{s-1} + 2T + 2 + 4t - 2u, a_{s-1} + 6T) := \begin{cases} (\approx, 1) & : \quad ((t \in U_s) \wedge (u = 2)) \vee ((t \in N_s) \wedge (u = 1)), \\ (=, 0) & : \quad \text{otherwise.} \end{cases}$$

For the transmitters we define for every $s \in [S]$ and $u \in [2T-1]$:

$$f(a_{s-1} + 2T + 2 + 2u, a_{s-1} + 2T + 2(u-1)) := \begin{cases} (=, 0) & : \quad (u \in 2\mathbb{N}_0) \vee ((u+1)/2 \in N_s), \\ (\approx, 1) & : \quad \text{otherwise,} \end{cases}$$

$$f(a_{s-1} + 2T + 2 + 2(u-1), a_{s-1} + 2T + 2u) := \begin{cases} (=, 0) & : \quad (u \in 2\mathbb{N} + 1) \vee ((u+1)/2 \in U_s), \\ (\approx, 1) & : \quad \text{otherwise.} \end{cases}$$

The remaining block constraints in the transmitter are set to zero, i.e., for every $s \in [S]$, $(u,v) \in [2T-1]_0^2$ with $u - v \notin \{-1, 1\}$ we set

$$f(a_{s-1} + 2T + 2 + 2u, a_{s-1} + 2T + 2v) := (=, 0).$$

For $s \in [S]$ and $t \in [T]$ we refer to the sets of points

$$(a_{s-1} + 2T, a_{s-1} + 2T) + \begin{cases} \{(1, 4t-2), (4t, 4t-3), (4t-1, 4T)\} & : \quad t \in U_s, \\ \{(1, 4t-3), (4t-1, 4t-2), (4t, 4T)\} & : \quad t \in N_s, \\ \{(4t-1, 4t-2), (4t, 4t-3)\} & : \quad \text{otherwise,} \end{cases}$$

as (s,t)-*configuration.* Figure 11 gives an illustration.

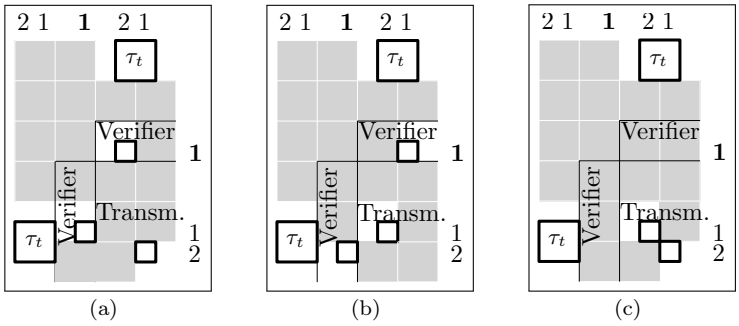


FIGURE 11. The three possible (s,t)-configurations (depicted as bold-framed 1×1 boxes) for (a) $t \in U_s$, (b) $t \in N_s$, and (c) $t \in [T] \setminus (U_s \cup N_s)$.

Outside the initializer, connectors, and clause chips we set everything to zero, i.e., for every $(i,j) \in ([m] \times [n]) \cap C(m,n,2)$ with

$$(i,j) \notin ((a_S,1) + [2(T-1)]_0^2) \cup \bigcup_{s \in [S-1]_0} \big(((a_s,a_s) + [2(T-1)]_0^2) \cup ((a_s,a_s+2T) + [6T+1]_0^2))\big)$$

we set $f(i,j) := (=,0)$.

This concludes the formal definition of the instance $\mathcal{I}'$. We shall now show that $\mathcal{I}'$ admits a solution if, and only if, the 1-IN-3-SAT instance $\mathcal{I}$ admits a solution.

Let $\mathcal{G} \subseteq [m] \times [n]$ denote the set of points of the Boolean chips and (s,t)-configurations, $(s,t) \in [S] \times [T]$.

Claim 1: In any solution x^* of $\mathcal{I}'$ we have $\xi^*_{p,q} = 0$ for every $(p,q) \in ([m] \times [n]) \setminus \mathcal{G}$.

To see this, we first remark that the claim holds for all points outside each clause chip since the only blocks there that are not set to zero are the Boolean chips of the initializer and connectors. For the points inside a clause chip, we need to distinguish three cases since there are three differently structured (s,t)-configurations, in which different blocks are set to zero. They are shown in Figure 11. The figure shows, however, that in all three cases it holds that the points outside the corresponding Boolean chips and the (s,t)-configuration are either set to zero by block constraints or by zeros of suitable row or column sums. This shows the claim.

Claim 2: In any solution x^* of $\mathcal{I}'$ every τ_t- and $\neg\tau_t$-chip, $t \in [T]$, is of one of the types $\ell \in \{1,2,3,4\}$ shown in Figure 8.

For this, consider a fixed index $t \in [T]$. Each τ_t-chip is contained in a horizontal or vertical strip that contains, except for a $\neg\tau_t$-chip, no other points of $\mathcal{G}$. Since we required via block constraints that each $\neg\tau_t$-chip contains two ones and since the respective row or column sums in the strip sum up to four there need to be exactly two ones also in each τ_t-chip. There are four possibilities to place two ones into a 2×2 box if it is required (as is the case via our row and column sums) that no two ones are in the rightmost column and upper row of the box. The four possibilities are shown in Figure 8.

Claim 3: In any solution x^* of $\mathcal{I}'$ it holds that for every $t \in [T]$ the τ_t-chips are either all of type 1 or type 2.

To show this, consider again a fixed index $t \in [T]$. Let $V_0, \dots, V_{4S+2} \subseteq \mathcal{G}$ denote the sequence with (i) $V_0 = V_{4S+2}$ is the τ_t-chip of the initializer, (ii) $V_{4(s-1)+1}$ is the $\neg\tau_t$-chip of the $(s-1)$-th connector, $s \in [S+1]$, (iii) $V_{4(s-1)+2}$ is the τ_t-chip of the s-th vertical collector, $s \in [S]$, (iv) $V_{4(s-1)+3}$ is the (s,t)-configuration of the s-th clause, $s \in [S]$, and (v) $V_{4(s-1)+4}$ is the τ_t-chip of the s-th horizontal collector, $s \in [S]$.

By vertical transmission (see Figure 12) and Claim 2, we have the following implication: If a τ_t-chip of V_{2l-1}, $l \in [2S+1]$, is of type 1, 3 or 4, then the τ_t-chip of V_{2l+1} is of type 1, and the points of V_{2l} are also uniquely determined (here, $V_{4S+4} := V_1$).

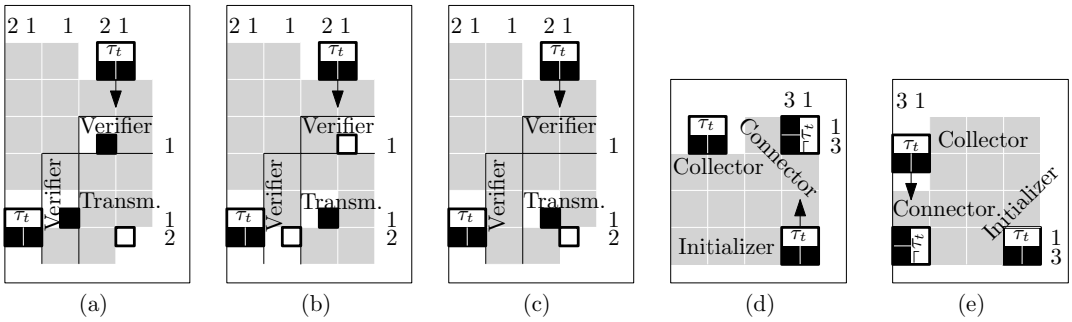


FIGURE 12. Vertical transmission of a type $\ell \in \{1, 3, 4\}$ block. (a,b,c) Transmission through the three differently structured (s, t)-configurations. (d,e) Transmission through a connector.

By horizontal transmission (see Figure 13) and Claim 2, we have the following implication: If a τ_t-chip of V_{2l-1}, $l \in [2S+1]$, is of type 2, 3 or 4, then the τ_t-chip of V_{2l-3} is of type 2, and the points of V_{2l-2} are also uniquely determined (here, $V_{-1} := V_{4S+1}$).

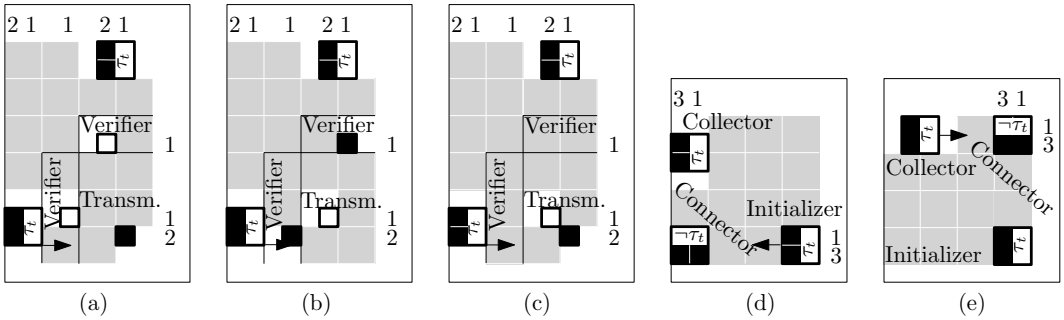


FIGURE 13. Horizontal transmission of a type $\ell \in \{2, 3, 4\}$ block. (a,b,c) Transmission through the three differently structured (s, t)-configurations. (d,e) Transmission through a connector.

Since $V_0 = V_{4S+2}$, we have proved Claim 3, and, in particular, we have ruled out that any τ_t-chip is of type 3 and 4.

The types 1 and 2 can be associated with Boolean values. In τ_t-chips, and in particular those of the initializer, we associate with type 1 the value TRUE and with type 2 the value FALSE.

Claim 4: There is a 1-to-1 correspondence between the (not-necessarily satisfying) truth-assignments for $\mathcal{I}$ and the solutions x^* to the $\mathcal{I}''$-variant of $\mathcal{I}'$ that allows violation of the verifier sums.

By construction, there is neither a horizontal nor a vertical strip that intersects the Boolean chips for two variables τ_{i_1}, τ_{i_2}, with $i_1 \neq i_2$. The arguments from Claim 3 are thus independently valid for each $t \in [T]$. We can hence consider an arbitrary (not-necessarily satisfying) truth-assignment for $\mathcal{I}$, fill in the corresponding types in the initializer and extend them to yield a solution to the $\mathcal{I}''$-variant of $\mathcal{I}'$ with unspecified verifier sums. Vice versa, by reading off the chip types in the initializer of a solution to $\mathcal{I}''$ yields a truth-assignment for $\mathcal{I}$. We have already seen in the proof of Claim 3 that the type of each τ_t-chip, $t \in [T]$, in the initializer determines all ones in the corresponding $\neg\tau_t$-chips and (s, t)-configurations, $s \in [S]$. Hence, the correspondence is indeed 1-to-1.

Claim 5: The satisfying truth-assignments of $\mathcal{I}$ are in 1-to-1 correspondence with the solutions of $\mathcal{I}'$.

This can now be seen by taking into account that the τ_t-chip, $t \in [T]$, of the s-th horizontal and vertical verifier, $s \in [S]$, contributes a one to the horizontal and, respectively, vertical verifier in the s-th clause chip if, and only if, the τ_t-chip is TRUE (is of type 1) with $t \in U_s$ or the τ_t-chip is FALSE (is of type 2) with $t \in N_s$; see Figures 11, 12(a,b,c), and 13(a,b,c). The verifier sums thus ensure that exactly one literal in the s-th clause is TRUE. This, together with Claim 4, proves the claim.

By Claim 5 it holds that $\mathcal{I}'$ admits a solution if, and only if, $\mathcal{I}$ admits a solution.

Finally, we note that the transformation runs in polynomial time. □

In Section 2 we remarked that $n\mathrm{DR}(\varepsilon)$ can be viewed as version of DR where small "occasional" uncertainties in the gray levels are allowed. Reviewing our proof of Theorem 3, the term "occasional" can be quantified as meaning "of the order of the square root of the number of blocks."

Let us further remark that the arguments in the above proof do not rely on the particular value of $\varepsilon > 0$ in the noisy block constraints. These constraints can be replaced by any other types of block constraints as long as they allow for 0, 1, and 2 ones to be contained in each of the corresponding blocks. By this observation it is possible to adapt our proof to several other contexts where the reconstruction task involves other types or combinations of block constraints; see [5].

Finally, note that the transformation in the proof of Theorem 3 is parsimonious (see Claim 5 in the above proof). Hence the problem of deciding whether a given solution of an instance of $\textsc{n}\mathrm{DR}(\varepsilon)$ with $\varepsilon > 0$ has a non-unique solution is $\mathbb{NP}$-complete; for an $\mathbb{NP}$-hardness proof of Unique-1-in-3-SAT see [13, Lemma 4.2].

Proof of Corollary 1. We use a transformation from $\textsc{n}\mathrm{DR}(\varepsilon)$. Let

$$\mathcal{I} := (m, n, r_1, \ldots, r_n, c_1, \ldots, c_m, R, v(1,1), \ldots, v(m-1, n-1))$$

denote an instance of $\textsc{n}\mathrm{DR}(\varepsilon)$.

We set $m' := mk/2$ and $n' := nk/2$. For $(i,j) \in ([m] \times [n]) \cap C(m,n,2)$ and $l \in [k]$ we define

$$r'_{\frac{k}{2}(j-1)+l} := \begin{cases} r_{j+l} & : \quad l \in [2], \\ 0 & : \quad \text{otherwise}, \end{cases}$$

$$c'_{\frac{k}{2}(i-1)+l} := \begin{cases} c_{i+l} & : \quad l \in [2], \\ 0 & : \quad \text{otherwise}, \end{cases}$$

$$v'(k(i-1)/2+1, k(j-1)/2+1)) := v(i,j),$$

and

$$R' := \{(k(i-1)/2+1, k(j-1)/2+1) : (i,j) \in R\}.$$

This defines an instance

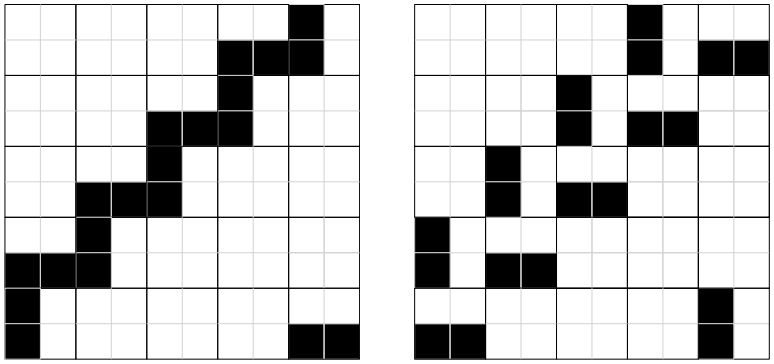
$$\mathcal{I}' := (m', n', r'_1, \ldots, r'_{n'}, c'_1, \ldots, c'_{m'}, R', v'(1,1), \ldots, v'(m'-k+1, n'-k+1))$$

of $\textsc{n}\mathrm{SR}(k,\varepsilon)$. Clearly, the instance $\mathcal{I}$ of $\textsc{n}\mathrm{DR}(\varepsilon)$ admits a solution if, and only if, $\mathcal{I}'$ of $\textsc{n}\mathrm{SR}(k,\varepsilon)$ admits a solution (by filling/extracting the 2×2-blocks of $\mathcal{I}$ into/from the $k \times k$-blocks of $\mathcal{I}'$). This is a polynomial-time transformation, proving the theorem. □

5. Final Remarks

Recall that in the proof of Theorem 1 we have made extensive use of the concept of local switches. Each such switch involves four points. Figure 14 shows, on the other hand, two solutions of a DR instance that differ by 20 pixels, and for which it is easily checked that they are the only two solutions for that instance (such a structure has been utilized in the proof of Theorem 3). In this case there is thus no sequence of small switches transforming one solution into the other. The existence of such large irreducible switches is well-known for the $\mathbb{NP}$-hard reconstruction problem from more than two directions [24] (and would follow independently from $\mathbb{P} \neq \mathbb{NP}$). The problem DR, however, admits large switches albeit is polynomial-time solvable.

In addition to the block constraints our computational tasks rely only on row and column sums, i.e., projection data from two viewing angles. For more than two viewing directions we obtain the $\mathbb{NP}$-hardness of $\textsc{n}\mathrm{SR}(k,\varepsilon)$ for any $k \geq 2$ and $\varepsilon > 0$ directly from the construction in [13]. The $\mathbb{NP}$-hardness of the corresponding variants of $\textsc{n}\mathrm{SR}(k,0)$ for $k \geq 2$, can also be derived from the construction in [13]. First the construction is thinned out so as to make sure that each block contains at most one element



(a) (b)

FIGURE 14. An example of a large switching component (in the sense of [24]) for an instance of DR. The solutions shown in (a) and (b) are the only two solutions for that instance. Both solutions are reduced.

of the grid of candidate points. Then, a second copy with inverted colors is interwoven by applying a translation by $(1,1)^T$ or $(1,-1)^T$. This ensures that each non-empty block contains exactly one one.

The complexity status of $\text{NSR}(k,0)$ as introduced in Section 2 remains open for $k \notin \{1,2\}$. We conjecture that the problem is $\mathbb{NP}$-hard for larger k.

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] A. Alpers, A. Brieden, P. Gritzmann, A. Lyckegaard, and H. F. Poulsen. Generalized balanced power diagrams for 3D representations of polycrystals. *Phil. Mag.*, 95(9):1016–1028, 2015.

[3] A. Alpers, R.J. Gardner, S. König, R.S. Pennington, C.B. Boothroyd, L. Houben, R.E. Dunin-Borkowski, and K.J. Batenburg. Geometric reconstruction methods for electron tomography. *Ultramicroscopy*, 128(C):42–54, 2013.

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

[5] A. Alpers and P. Gritzmann. Reconstructing binary matrices under window constraints from their row and column sums. manuscript, 2017.

[6] 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.

[7] A. Alpers, H. F. Poulsen, E. Knudsen, and G. T.Herman. A discrete tomography algorithm for improving the quality of 3DXRD grain maps. *J. Appl. Crystallogr.*, 39(4):582–588, 2006.

[8] M. Bertero and P. Boccacci. *Introduction to Inverse Problems in Imaging*. IOP Publishing, Philadelphia, 1998.

[9] R. A. Brualdi. *Combinatorial Matrix Classes*. Cambridge University Press, Cambridge, 2006.

[10] A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vision*, 20(1):89–97, 2004.

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

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

[13] R. J. Gardner, P. Gritzmann, and D. Prangenberg. On the computational complexity of reconstructing lattice sets from their X-rays. *Discrete Math.*, 202(1-3):45–71, 1999.

[14] M. R. Garey and D. S. Johnson. *Computers and Intractability*. W. H. Freeman and Co., San Francisco, USA, 1979.

[15] G. Van Gompel, K. J. Batenburg, E. Van de Casteele, W. van Aarle, and J. Sijbers. A discrete tomography approach for superresolution micro-CT images: application to bone. In *2010 IEEE Intern. Symp. Biomedical Imaging: From Nano to Macro*, pages 816–819, 2010.

[16] F. Grandoni, R. Ravi, M. Singh, and R. Zenklusen. New approaches to multi-objective optimization. *Math. Program.*, 146(1):525–554, 2014.

[17] 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.

[18] P. C. Hansen. *Discrete Inverse Problems: Insights and Algorithms*. SIAM, Philadelphia, 2010.

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

[20] G. T. Herman and A. Kuba (eds.). *Advances in Discrete Tomography and its Applications*. Birkhäuser, Boston, 2007.
[21] R. W. Irving and M. R. Jerrum. Three-dimensional statistical data security problems. *SIAM J. Comput.*, 23(1):170–184, 1994.
[22] J. A. Kennedy, O. Israel, A. Frenkel, R. Bar-Shalom, and H. Azhari. Super-resolution in PET imaging. *IEEE Trans. Med. Imaging*, 25(2):137–147, 2006.
[23] 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.
[24] T. Y. Kong and G. T. Herman. On which grids can tomographic equivalence of binary pictures be characterized in terms of elementary switching operations? *Int. J. Imaging Syst. Technol.*, 9(2-3):118–125, 1998.
[25] A. K. Kulshreshth, A. Alpers, G. T. Herman, E. Knudsen, L. Rodek, and H. F. Poulsen. A greedy method for reconstructing polycrystals from three-dimensional x-ray diffraction data. *Inverse Probl. Imaging*, 3(1):69–85, 2009.
[26] G. E. Marai, D. H. Laidlaw, and J. J. Crisco. Super-resolution registration using tissue-classified distance fields. *IEEE Trans. Med. Imaging*, 25(2):177–187, 2006.
[27] P. Milanfar (ed.). *Super-Resolution Imaging*. CRC Press, Boca Raton, 2010.
[28] J. L. Mueller and S. Siltanen. *Linear and Nonlinear Inverse Problems with Practical Applications*. SIAM, Philadelphia, 2012.
[29] L. Rodek, H. F. Poulsen, E. Knudsen, and G. T. Herman. A stochastic algorithm for reconstruction of grain maps of moderately deformed specimens based on X-ray diffraction. *J. Appl. Crystallogr.*, 40(2):313–321, 2007.
[30] H. J. Ryser. Combinatorial properties of matrices of zeros and ones. *Canad. J. Math.*, 9(1):371–377, 1957.
[31] A. Schatzberg and A. J. Devaney. Super-resolution in diffraction tomography. *Inverse Probl.*, 8(1):149–164, 1992.
[32] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Chichester, UK, 1986.
[33] 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.
[34] M. C. Scott, C.-C. Chen, M. Mecklenburg, C. Zhu, R. Xu, P. Ercius, U. Dahmen, B. C. Regan, and J. Miao. Electron tomography at 2.4-ångström resolution. *Nature*, 483(7390):444–447, 2012.
[35] E. Y. Sidky, Y. Duchin, X. Pan, and C. Ullberg. A constrained, total-variation minimization algorithm for low-intensity x-ray CT. *Med. Phys.*, 38(7):S117, 2011.
[36] W. van Aarle, K. J. Batenburg, G. Van Gompel, E. Van de Casteele, and J. Sijbers. Super-resolution for computed tomography based on discrete tomography. *IEEE Trans. Image Process.*, 23(3):1181–1193, 2014.
[37] 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.