ON DOUBLE-RESOLUTION IMAGING AND DISCRETE TOMOGRAPHY*

ANDREAS ALPERS† AND PETER GRITZMANN†

Abstract. Superresolution 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, in each dimension, the resolution of grayscale images of binary objects by fusion with double-resolution tomographic data that have been acquired from two viewing angles. We show that this task is polynomial-time solvable if the gray levels have been reliably determined. The problem 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.

Key words. discrete mathematics, combinatorics, discrete tomography, superresolution, polynomial-time algorithms, computational complexity

AMS subject classifications. 68R05, 68Q25, 49N45, 94A08, 68Q25

DOI. 10.1137/17M1115629

1. Introduction. Different imaging techniques in tomography have different characteristics that strongly depend on the specific data acquisition setup and the imaged tissue/material. Hence it is a major issue (and at the heart of current research; see, e.g., [10, 14, 27, 34, 36, 39, 43]) to improve the resolution of an image by combining different imaging techniques. In general it is, however, not clear, how this can actually be performed efficiently. The present paper addresses this multimodal approach from a basic algorithmic point of view in two different ways using a discrete model of binary image reconstruction. First, we show that, indeed, the resolution of binary objects can be efficiently improved if the data are reliable. We derive a polynomial-time algorithm for double resolution that works correctly for exact data (and can be extended to a heuristic for the more general case). On the other hand, we show that noisy data in the low-resolution image do not simply reduce the quality of the reconstruction but add additional complexity. Hence, intuitively speaking, and in a sense that will be made mathematically precise later, it might not be possible to compensate for the faultiness of low-resolution imaging efficiently by incorporating other additional higher-resolution information (even if the latter is noise free).

More specifically, the results of the present 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, 3, 7, 46, 49]. 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 high-resolution binary image we utilize the gray levels and two additional high-resolution projections, which may have been

---

†Zentrum Mathematik, Technische Universität München, D-85747 Garching bei München, Germany (alpers@ma.tum.de, gritzmann@tum.de).
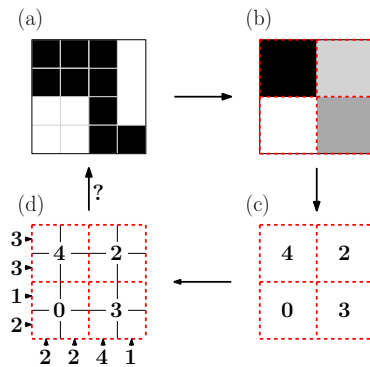
FIG. 1. *The superresolution imaging task* DR. (a) *Original (unknown) high-resolution image;* (b) *the corresponding low-resolution grayscale image;* (c) *gray levels converted into block constraints;* (d) *taken in combination with high-resolution row and column sum data. The task is to reconstruct from* (d) *the original binary image shown in* (a).

acquired by different imaging techniques or modalities (e.g., via scanning transmission electron microscopy [46]). 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 are taken from orthogonal directions. In our context it suffices that the X-ray data have been taken with high resolution according to the discretization of the low-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 superresolution imaging [29, 33, 35, 37, 42, 47], and its particular applications [8, 20, 40, 48] in discrete tomography [1, 17, 22, 25, 26, 28, 30, 32, 45], the problems 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] and, to some extent, at the end of this section. Background information on such problems involving different kinds of additional constraints can be found in [11, sect. 4].

Other applications belong to the realm of *dynamic discrete tomography* [6, 5, 7, 49]. 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 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 [6].

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 theorems can also be seen as results on finding b-matchings that are subject to specific additional constraints. Related (but intrinsically quite different) results on matchings subject to so-called budget constraints can be found, e.g., in [21].
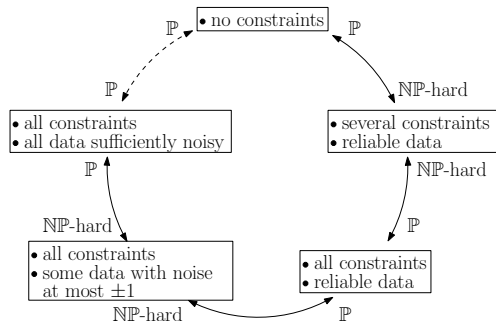
FIG. 2. *Overview of complexity jumps for the problem of reconstructing a binary image from row and column sums and additional* 2×2*-block constraints.*

For superresolution imaging the two main contributions of this paper are as follows. On the one hand, we show that reliable bimodal tomographic data can be utilized efficiently, i.e., the resolution, in each dimension, can be doubled in polynomial time if the gray levels of the low-resolution image have been determined precisely (Theorem 2.1). 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 2.3 and Corollary 2.5). 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. While this is not the focus of the present paper let us point out that, even in the presence of (a reasonably restricted degree of) noise, the approach leading to Theorem 2.1 can, in principle, be extended to a fast heuristic for doubling the resolution of such images in each dimension. At present, however, we do not know how such an approach would compare with other heuristics on real-world data.

From the perspective of discrete tomography, the contributions of this paper can be interpreted as follows. It is well known that the problem of reconstructing binary images from X-ray data taken from two directions can be solved in polynomial time [16, 41]. Typically, this information does not determine the image uniquely (see, e.g., [23] and the papers quoted there). Hence, one would like to take and utilize additional measurements. If, however, we add additional constraints that enforce that the solutions satisfy the X-ray data taken from a third direction, then the problem becomes $\mathbb{NP}$-hard, and it remains $\mathbb{NP}$-hard if X-ray data from even more directions are given [18] (see also [15] for results on a polyatomic version).

As it turns out, the case of block constraints behaves somewhat differently. Theorem 2.3 and Corollary 2.5 show that the problem of reconstructing a binary image from X-ray data taken from two directions is again $\mathbb{NP}$-hard if we add *several* (but not all) block constraints (which need to be satisfied with equality). However, and possibly less expectedly, if we include *all* block constraints, then the problem becomes polynomial-time solvable (Theorem 2.1). If, on the other hand, from all block constraints *some* of the data come with *noise* at most ± 1, then the problem becomes again $\mathbb{NP}$-hard (Theorem 2.3 and Corollary 2.5). And yet again, if from all block constraints all of the data are *sufficiently noisy*, then the problem is in $\mathbb{P}$ (as this is again the problem of reconstructing binary images from X-ray data taken from two directions). An overview of these complexity jumps is given in Figure 2.
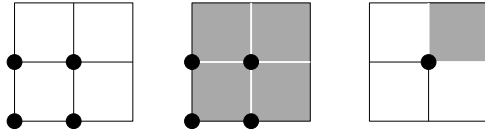
FIG. 3. *Lattice points (left) and pixels (middle); right: pixel associated with its lattice point.*

2. Notation and main results. Let $\mathbb{Z}$, $\mathbb{N}$, and $\mathbb{N}_0$ denote the set of integers, natural numbers, and nonnegative integers, respectively. For $k \in \mathbb{N}$ set $k\mathbb{N}_0 := \{ki : i \in \mathbb{N}_0\}$, $[k] := \{1, \ldots, k\}$, and $[k]_0 := \{0, \ldots, k\}$. 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-coordinates 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 $\{(i-1)k+1, \ldots, ik\} \times [n]$ is a *vertical strip (of width k)*. A *horizontal strip (of width k)* is a set of the form $[m] \times \{(j-1)k+1, \ldots, jk\}$.

Sets of the form $([a, b] \times [c, d]) \cap \mathbb{Z}^2$ with $a, b, c, d \in \mathbb{Z}$ and $a \le b$, $c \le d$, 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., $\bigcup_{(i,j) \in C(m,n,k)} B_k(i, j) = [m] \times [n]$.

Let us remark that, as defined, all our images consist of lattice points. However, the figures will be given in terms of pixels. Accordingly, the block $B_k(i, j)$ will be depicted as $(i, j) + [0, k-1]^2$; see Figure 3. The individual pixels are $(p, q) + [0, 1]^2$, $p, q \in [2]$, i.e., are identified with the lattice point at their lower-left corner. Of course, the fact that a point is or is not present in a solution is indicated by a black or a white pixel, respectively.

For $\varepsilon, k \in \mathbb{N}_0$ with $k \ge 2$ we define the task of (noisy) superresolution:
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 measurements)
$R \subseteq C(m, n, k)$,	(corner points of reliable gray value measurements)
$v(i, j) \in [k^2]_0$, $\quad (i, j) \in C(m, n, k)$,	(gray value measurements)

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

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

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

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

$$\sum_{(p,q) \in B_k(i,j)} \xi_{p,q} \in v(i, j) + [-\varepsilon, \varepsilon], \quad (i, j) \in C(m, n, k) \setminus R, \text{ (noisy block constraints)}$$
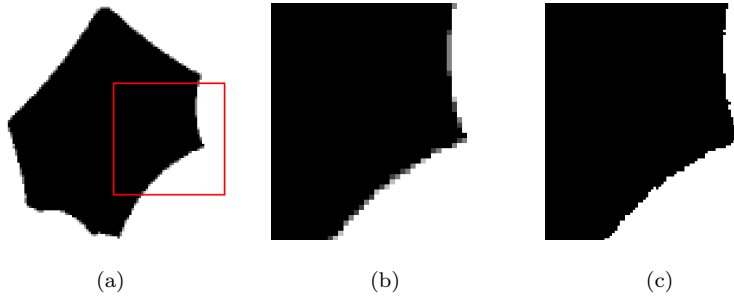
or decide that no such solution exists.

FIG. 4. *Algorithm* 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 part of the* 200×200 *high-resolution binary image obtained by the algorithm.*

The numbers $r_1, \ldots, r_n$ and $c_1, \ldots, c_m$ are the row and column sum measurements of the high-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. (While it may seem unusual to denote a nonnegative integer by ε we chose this notation to indicate the specific role of ε of quantifying an error.) 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 is the gray value for the corresponding $k \times k$-pixel low-resolution image. 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 $\mathrm{NDR}(\varepsilon) := \mathrm{NSR}(2, \varepsilon)$. In the reliable situation, i.e., for $\varepsilon = 0$, we simply speak of *double resolution* and set $\mathrm{DR} := \mathrm{NSR}(2, 0)$. (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 are provided from two viewing angles at double resolution. In other words, we show the following.

THEOREM 2.1. $\mathrm{DR} \in \mathbb{P}$.

Figures 4 and 5 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 4, the original phantom is a binary 200×200 image of a crystalline sample taken from [2]. It is assumed that the low-resolution grayscale image shown in Figure 4(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 (see Algorithm 1) yields the binary image shown in Figure 4(c).

In Figure 5, the original phantom is a binary 100×100 image, for which the low-resolution grayscale image shown in Figure 5(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 5(b).
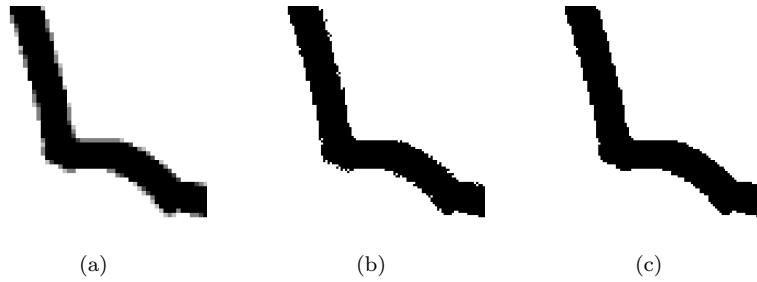
FIG. 5. *Algorithm 1 applied to a phantom.* (a) *Low-resolution* 50×50 *grayscale image;* (b) *high-resolution* 100×100 *binary image obtained by the algorithm;* (c) *high-resolution* 100×100 *binary image minimizing the total variation.*

The results shown in Figures 4 and 5 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 2.2. *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 nonuniqueness in practice is regularization [8, 9, 24, 38]. To illustrate the possibility of adapting a regularization scheme to our context, we show in Figure 5(c) a binary solution minimizing the *total variation*. In fact, the solution is obtained from Figure 5(a) by applying 15 so-called local switches, each of which is strictly decreasing the value of the total variation functional

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

where

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

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

and $|| \cdot ||_2$ denotes the Euclidean norm. See, e.g., [12] 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) nonconstant number of uncertainties the problem becomes hard.

THEOREM 2.3. $\mathrm{NDR}(\varepsilon)$ *is* $\mathbb{NP}$-*hard for any* $\varepsilon > 0$.

(For background material on complexity theory, see, e.g., [19].)
Further, the $\mathbb{NP}$-hardness extends to the task of checking uniqueness.

COROLLARY 2.4. *The problem of deciding whether a given solution of an instance of* $\mathrm{NDR}(\varepsilon)$ *with* $\varepsilon > 0$ *has a nonunique solution is* $\mathbb{NP}$-*complete.*

As it turns out, the problem $\mathrm{NDR}(\varepsilon)$ with $\varepsilon > 0$ remains $\mathbb{NP}$-hard for larger block sizes.

COROLLARY 2.5. $\mathrm{NSR}(k, \varepsilon)$ *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.

The present paper is organized as follows. We deal with the case of reliable data in section 3. Results on $\mathrm{NDR}(\varepsilon)$ and $\mathrm{NSR}(k, \varepsilon)$ 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 4 and 5.

The main result of this section is Theorem 2.1. In its proof we show that DR decomposes into five problems that can be solved independently. The five problems are restricted single-graylevel versions of DR where each block is required to contain the same number $\nu \in [4]_0$ of ones. In fact, we allow for restricted problem instances (and in particular row and column sums) to be defined only on subsets

$$(1) \qquad\qquad G(I) := \bigcup_{(i,j) \in I} B(i,j) \subseteq [m] \times [n]$$

of the grid $[m] \times [n]$ given 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

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

denote the projection of I onto the first and second coordinates, respectively.

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

$\mathrm{DR}(\nu)$

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 measurements)

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

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 $\mathrm{DR}(\nu)$, $\nu \in \{0, 4\}$, are trivial. In fact, the only potential solution x^* satisfying the block constraints is given by $\xi^*_{p,q} = \nu/4$, $(p, q) \in G(I)$, and
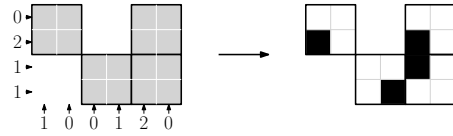
FIG. 6. *Illustration of* DR(1). *(Left) Row and column sums and blocks* $B(i,j)$ *with* $(i,j) \in I$ *in gray color. (Right) Solution defined by* (2).

it is checked easily and in polynomial time whether this satisfies the row and column sums.

The next lemma and corollary deal with $DR(\nu)$, $\nu \in \{1,3\}$. Their statements and proofs use 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

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

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

LEMMA 3.1.
(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*

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

(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

$$a_{i,j} := i + \min\left\{ l \in \{0,1\} : \sigma_i(j) \le \sum_{h=0}^{l} c_{i+h} \right\},$$

(2)
$$b_{i,j} := j + \min\left\{ l \in \{0,1\} : \rho_j(i) \le \sum_{h=0}^{l} r_{j+h} \right\},$$

$$\xi_{p,q}^* := \begin{cases} 1 & : & (p,q) = (a_{i,j}, b_{i,j}), \\ 0 & : & \text{otherwise.} \end{cases}$$

Figure 6 gives an illustration.

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

$$\sum_{p:(p,j)\in G(I)} \xi^*_{p,j} = |\{p : ((p,j) \in I) \wedge (\rho_j(p) \leq r_j)\}| = r_j,$$

$$\sum_{p:(p,j)\in G(I)} \xi^*_{p,j+1} = |\{p : ((p,j) \in I) \wedge (\rho_j(p) > r_j)\}| = \rho_j(m) - r_j = r_{j+1},$$

$$\sum_{q:(i,q)\in G(I)} \xi^*_{i,q} = |\{q : ((i,q) \in I) \wedge (\sigma_i(q) \leq c_i)\}| = c_i,$$

$$\sum_{q:(i,q)\in G(I)} \xi^*_{i+1,q} = |\{q : ((i,q) \in I) \wedge (\sigma_i(q) > c_i)\}| = \sigma_i(n) - c_i = c_{i+1}.$$

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 $\xi'_{i,j} := \xi'_{l,j+1} := 0$, $\xi'_{i,j+1} := \xi'_{l,j} := 1$, and $\xi'_{p,q} := \xi^*_{p,q}$ 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 nonzero 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 (2). Clearly, all steps can be performed in polynomial time. □

COROLLARY 3.2.
(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*

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

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

Proof. The results follow directly from the results for DR(1) in Lemma 3.1 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 [41]).

LEMMA 3.3. *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 nonnegative integers $\zeta_{i,j}$, $\eta_{i,j}$, $(i, j) \in I$, such*

that

$$\sum_{p:(p,j)\in I} \zeta_{p,j} = r'_j, \qquad j \in \Pi_y(I),$$

$$\sum_{q:(i,q)\in I} \eta_{i,q} = c'_i, \qquad i \in \Pi_x(I),$$

$$\zeta_{i,j} + \eta_{i,j} \leq 1, \quad (i,j) \in I,$$

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

Proof. We show that the problem can be formulated as a linear program involving a totally unimodular coefficient matrix (for background material, see [44]).

We assemble the variables $\zeta_{i,j}$, $\eta_{i,j}$ for $(i,j) \in I$ into a $2 \cdot |I|$-dimensional vector and rephrase the constraints in matrix form $Ax = r'$, $A'x = c'$, and $(E\ E)x \leq \mathbf{1}$, 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

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

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. □

Next we study the structure of solutions in terms of the patterns of the blocks. This will allow us to focus on special classes of solutions, which we will later call *reduced* solutions. The problem of finding reduced solutions to a given instance of DR will decompose into the five independent subtasks of solving instances of DR(ν), $\nu \in \{0, \dots, 4\}$. The concept of reduced solutions will also play a role in our proof of Lemma 3.5 showing that DR(2) $\in \mathbb{P}$.

Blocks in solutions are filled with zeros and ones. There are 16 such combinations. By grouping the combinations that have the same row sums, we distinguish the nine block types $\mathfrak{A}_1$, $\mathfrak{A}_2$, $\mathfrak{B}_1$, $\mathfrak{B}_2$, $\mathfrak{B}_3$, $\mathfrak{C}_1$, $\mathfrak{C}_2$, $\mathfrak{D}_1$, $\mathfrak{D}_2$ shown in Figure 7. Some of the types consist of several subtypes/combinations: type $\mathfrak{A}_i$ and $\mathfrak{C}_i$ consist of the two subtypes $\mathfrak{A}_{i,i'}$ and $\mathfrak{C}_{i,i'}$ $i, i' \in [2]$, respectively, while $\mathfrak{B}_3$ consists of the subtypes $\mathfrak{B}_{3,1}, \dots, \mathfrak{B}_{3,4}$. For any given solution, we will later count the number of occurrences of the different block types in each horizontal and vertical strip. The corresponding variables for the horizontal strip $[m] \times \{j, j+1\}$ are indicated in Figure 7. For instance, α_j counts the number of occurrences of block type $\mathfrak{A}_1$, while α_{j+1} counts the number of occurrences of block type $\mathfrak{A}_2$.

We introduce now the concept of local switches. It is based on the following observation. If, for instance, a horizontal strip contains a block of type $\mathfrak{A}_{1,i}$, $i \in [2]$,
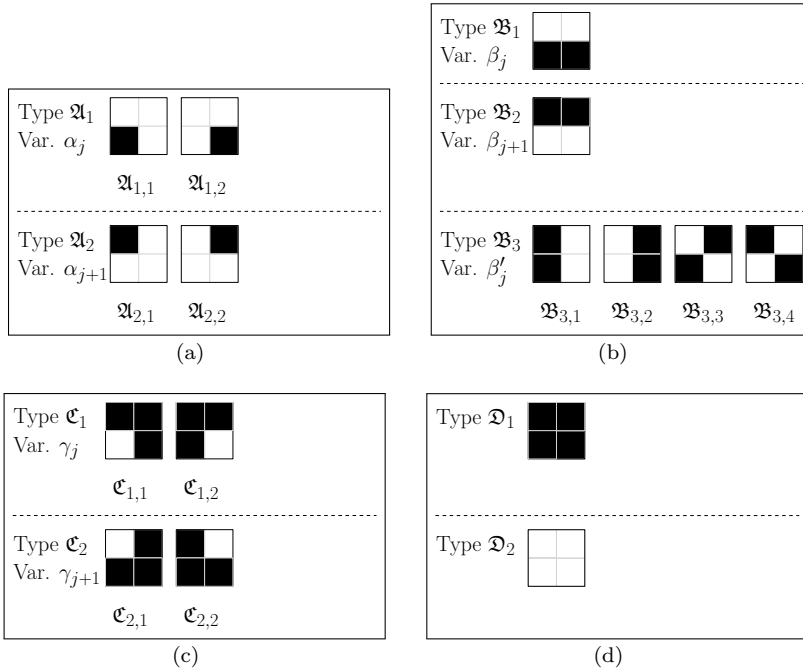
FIG. 7. *Block types* $\mathfrak{A}_1$, $\mathfrak{A}_2$, $\mathfrak{B}_1$, $\mathfrak{B}_2$, $\mathfrak{B}_3$, $\mathfrak{C}_1$, $\mathfrak{C}_2$, $\mathfrak{D}_1$, $\mathfrak{D}_2$. *The indicated variables* α_j, α_{j+1}, β_j, β_{j+1}, β'_j, γ_j, γ_{j+1} *count the number of occurrences of the respective 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. For referencing purposes, the figure also introduces the subtypes* $\mathfrak{A}_{i_1,i_2}$ *of* $\mathfrak{A}_{i_1}$, $\mathfrak{B}_{3,i_3}$ *of* $\mathfrak{B}_3$, *and* $\mathfrak{C}_{i_1,i_2}$ *of* $\mathfrak{C}_{i_1}$, $i_1, i_2 \in [2]$, $i_3 \in [3]$.

and another one of type $\mathfrak{B}_2$ we may replace them by a block of type $\mathfrak{A}_{2,i}$ and a block of type $\mathfrak{B}_{3,3}$, respectively, without changing the row and column constraints. Such a change will be called a *horizontal local switch of class* $\langle 1 \rangle$. In a similar way we define *horizontal local switches of class* $\langle t \rangle$, $t = 2, \ldots, 7$, according to Table 1. Horizontal local switches of class $\langle 7 \rangle$ affect only single blocks (blocks of type $\mathfrak{B}_{3,4}$ are turned into blocks of type $\mathfrak{B}_{3,3}$).

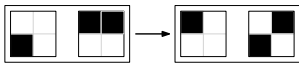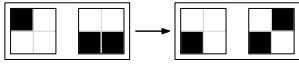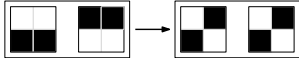The (similar) switches in the vertical strips, called vertical local switches, are listed in Table 2.
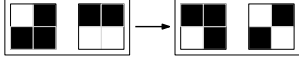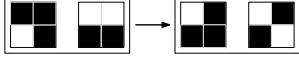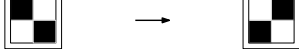
A switch is called *local* if it is a horizontal or vertical local switch of some class $\langle t \rangle$, $t \in [7]$. A solution is called *reduced* if no local switch of any class $\langle t \rangle$, $t \in [7]$, can be applied. Note that local switches are directed in the sense that they turn the types of blocks listed in the second column of Tables 1 and 2 into the types given in the respective third column of the tables. Oppositely directed switches, called *reversed local switches*, will be used later when we discuss questions of uniqueness.

LEMMA 3.4.
(i) *Application of a local switch to a solution of a given instance of* DR *yields again a solution of the same instance.*
(ii) *An instance of* DR *has a solution if and only if there is a reduced solution.*

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

TABLE 1
Horizontal local switches. The parameters i and i' are elements of $[2]$.

Class	Turns	Into	Illustration
$\langle 1 \rangle$	$(\mathfrak{A}_{1,i}, \mathfrak{B}_2)$	$\rightarrow$ $(\mathfrak{A}_{2,i}, \mathfrak{B}_{3,3})$	
$\langle 2 \rangle$	$(\mathfrak{A}_{2,i}, \mathfrak{B}_1)$	$\rightarrow$ $(\mathfrak{A}_{1,i}, \mathfrak{B}_{3,3})$	
$\langle 3 \rangle$	$(\mathfrak{B}_1, \mathfrak{B}_2)$	$\rightarrow$ $(\mathfrak{B}_{3,3}, \mathfrak{B}_{3,3})$	
$\langle 4 \rangle$	$(\mathfrak{C}_{2,i}, \mathfrak{B}_2)$	$\rightarrow$ $(\mathfrak{C}_{1,i}, \mathfrak{B}_{3,3})$	
$\langle 5 \rangle$	$(\mathfrak{C}_{1,i}, \mathfrak{B}_1)$	$\rightarrow$ $(\mathfrak{C}_{2,i}, \mathfrak{B}_{3,3})$	
$\langle 6 \rangle$	$(\mathfrak{A}_{1,i}, \mathfrak{C}_{1,i'})\rightarrow$	$(\mathfrak{A}_{2,i}, \mathfrak{C}_{2,i'})$	
$\langle 7 \rangle$	$(\mathfrak{B}_{3,4})$	$\rightarrow$ $(\mathfrak{B}_{3,3})$	

Let us now turn to (ii). Suppose the given instance has a solution x^*. The local switches of class $\langle t \rangle$, $t \in [7] \setminus \{6\}$, increase the number of blocks $B(i,j)$ of type $\mathfrak{B}_{3,3}$, i.e.,

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

The process of applying local switches of class $\langle t \rangle$, $t \in [7] \setminus \{6\}$, thus needs to terminate since the number of blocks in each problem instance is finite. Further, note that local switches of class $\langle 6 \rangle$ applied in horizontal strips increase the number of blocks of type $\mathfrak{A}_2$. The vertical switches of that class do not decrease the number of such blocks, but increase the number of blocks which are of type $\mathfrak{A}_{1,2}$ or $\mathfrak{A}_{2,2}$. Hence, after finitely many switches, a given solution is converted into a reduced one. □

LEMMA 3.5. $\mathrm{DR}(2) \in \mathbb{P}$.

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.

By rearranging the rows and columns, if necessary, we may assume that

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

Now, first suppose that the given instance has a solution. Then, by Lemma 3.4 there exists a reduced one. In the following, we consider such a reduced solution.

The potential block types, and corresponding variables β_j, β_{j+1}, and β'_j counting the number of occurrences of the respective block types in the horizontal strip $[m] \times \{j, j+1\}$ of a reduced solution, are shown in Figure 7(b). Block types counted by the same variable have equal row sums.

Class	Turns	Into	Illustration
$\langle 1 \rangle$	$(\mathfrak{B}_{3,2}, \mathfrak{A}_{i,1})$	$\rightarrow \quad (\mathfrak{B}_{3,3}, \mathfrak{A}_{i,2})$	
$\langle 2 \rangle$	$(\mathfrak{B}_{3,1}, \mathfrak{A}_{i,2})$	$\rightarrow \quad (\mathfrak{B}_{3,3}, \mathfrak{A}_{i,1})$	
$\langle 3 \rangle$	$(\mathfrak{B}_{3,2}, \mathfrak{B}_{3,1})$	$\rightarrow \quad (\mathfrak{B}_{3,3}, \mathfrak{B}_{3,3})$	
$\langle 4 \rangle$	$(\mathfrak{B}_{3,2}, \mathfrak{C}_{i,2})$	$\rightarrow \quad (\mathfrak{B}_{3,3}, \mathfrak{C}_{i,1})$	
$\langle 5 \rangle$	$(\mathfrak{B}_{3,1}, \mathfrak{C}_{i,1})$	$\rightarrow \quad (\mathfrak{B}_{3,3}, \mathfrak{C}_{i,2})$	
$\langle 6 \rangle$	$(\mathfrak{C}_{i',1}, \mathfrak{A}_{i,1})$	$\rightarrow \quad (\mathfrak{C}_{i',2}, \mathfrak{A}_{i,2})$	
$\langle 7 \rangle$	$(\mathfrak{B}_{3,4})$	$\rightarrow \quad (\mathfrak{B}_{3,3})$	

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

$$
\begin{aligned}
2\beta_j \quad + \quad \beta'_j &= r_j, \\
2\beta_{j+1} + \quad \beta'_j &= r_{j+1},
\end{aligned}
$$

which implies

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

As the solution is reduced no horizontal local switch of class $\langle 3 \rangle$ can be applied, hence, it is impossible that both $\beta_j > 0$ and $\beta_{j+1} > 0$. This, together with $r_j \geq r_{j+1}$ and (3), implies

$$\beta_{j+1} = 0, \qquad \beta_j = (r_j - r_{j+1})/2, \quad \text{and} \quad \beta'_j = r_{j+1}.$$

(Note that by $\beta_{j+1} = 0$ there is no block of type $\mathfrak{B}_2$. As no horizontal local switch of class $\langle 7 \rangle$ can be applied, there is neither a block of type $\mathfrak{B}_{3,4}$.)
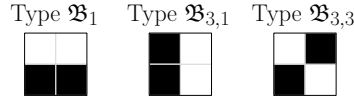
Type $\mathfrak{B}_1$ Type $\mathfrak{B}_{3,1}$ Type $\mathfrak{B}_{3,3}$

Fig. 8. *The three potential block types in solutions of* DR(2) *instances considered in the proof of Lemma* 3.5. *Values 1 and 0 are indicated in black and white, respectively.*

The same argument applies to the other horizontal strips, and a similar argument holds for the vertical strips (which, in particular, rules out blocks of type $\mathfrak{B}_{3,2}$). Hence, we conclude that, due to reduction, DR(2) amounts to the task of assigning one of the three block types $\mathfrak{B}_1, \mathfrak{B}_{3,1}, \mathfrak{B}_{3,3}$ shown in Figure 8 to each of the blocks $B(i,j)$, $(i,j) \in I$, such that, in particular, the number of type $\mathfrak{B}_1$ and type $\mathfrak{B}_{3,1}$ blocks in each horizontal and vertical strip, respectively, equals some prescribed cardinality.

Now, we introduce 0/1-variables $\zeta_{i,j}$, $\eta_{i,j}$, $\lambda_{i,j}$, $(i,j) \in I$, that are one if and only if $B(i,j)$ is of types $\mathfrak{B}_1$, $\mathfrak{B}_{3,1}$, $\mathfrak{B}_{3,3}$, respectively. Then we ask for 0/1-solutions that satisfy

$$\sum_{p:(p,j)\in I} \zeta_{p,j} = (r_j - r_{j+1})/2, \qquad j \in \Pi_y(I),$$

$$\sum_{p:(p,j)\in I} (\eta_{p,j} + \lambda_{p,j}) = \rho_j(m) - (r_j - r_{j+1})/2, \qquad j \in \Pi_y(I),$$

$$\sum_{q:(i,q)\in I} \eta_{i,q} = (c_i - c_{i+1})/2, \qquad i \in \Pi_x(I),$$

$$\sum_{q:(i,q)\in I} (\zeta_{i,q} + \lambda_{i,q}) = \sigma_i(n) - (c_i - c_{i+1})/2, \qquad i \in \Pi_x(I),$$

$$\zeta_{i,j} + \eta_{i,j} + \lambda_{i,j} = 1, \qquad (i,j) \in I.$$

Eliminating $\lambda_{i,j} = 1 - \zeta_{i,j} - \eta_{i,j}$, $(i,j) \in I$, we see that the 0/1-solutions of this system are in one-to-one correspondence with the 0/1 solution of

$$(4) \qquad \begin{aligned} \sum_{p:(p,j)\in I} \zeta_{p,j} &= (r_j - r_{j+1})/2, \quad j \in \Pi_y(I), \\ \sum_{q:(i,q)\in I} \eta_{i,q} &= (c_i - c_{i+1})/2, \quad i \in \Pi_x(I), \\ \zeta_{i,j} + \eta_{i,j} &\leq 1, \qquad\qquad (i,j) \in I. \end{aligned}$$

As we assume that our DR(2) instance is feasible (hence containing two ones in each block), we have $r_j + r_{j+1} = 2\rho_j(m)$ and $c_i + c_{i+1} = 2\sigma_i(n)$ for every $j \in \Pi_y(I)$ and $i \in \Pi_x(I)$. Hence, if a solution to (4) is expanded, replacing the $\zeta_{i,j}^*$, $\eta_{i,j}^*$, $1 - \zeta_{i,j}^* - \eta_{i,j}^*$, $(i,j) \in I$, by the respective block types, then, indeed

$$\sum_{p:(p,j)\in G(I)} \xi_{p,j}^* = 2 \cdot (r_j - r_{j+1})/2 + \rho_j(m) - (r_j - r_{j+1})/2 = r_j,$$

$$\sum_{p:(p,j)\in G(I)} \xi_{p,j+1}^* = \rho_j(m) - (r_j - r_{j+1})/2 = r_{j+1},$$

$$\sum_{q:(i,q)\in G(I)} \xi_{i,q}^* = 2 \cdot (c_i - c_{i+1})/2 + \sigma_i(n) - (c_i - c_{i+1})/2 = c_i,$$

$$\sum_{q:(i,q)\in G(I)} \xi_{i+1,q}^* = \sigma_i(n) - (c_i - c_{i+1})/2 = c_{i+1}$$

for every $j \in \Pi_y(I)$ and $i \in \Pi_x(I)$. Any 0/1-solution to (4) thus provides a solution to the given DR(2) instance. Since 0/1-solutions to (4) can be found in polynomial time (see Lemma 3.3), we have, hence, determined a solution to our instance in polynomial time. Note that this solution is reduced as it does not contain blocks of type $\mathfrak{B}_2$, $\mathfrak{B}_{3,2}$, and $\mathfrak{B}_{3,4}$.

It remains to consider the case that the given DR(2) instance is infeasible. Then one of the previous steps must fail: either (4) admits no 0/1-solution or the expanded "solution" does not satisfy all row and column sums. Both can be detected in polynomial time (see also Lemma 3.3). $\qquad \square$

Let us now turn to the task of solving DR. Suppose, we are given a feasible instance of DR that does not contain any block constraints of the form $v(i,j) = \nu$, $\nu \in \{0,4\}$, and whose row and column sums satisfy

$$(5) \qquad r_j \geq r_{j+1} \qquad \text{and} \qquad c_i \geq c_{i+1}, \quad (i,j) \in C(m,n,2).$$

We call such an instance a *proper* instance.

In the following, we consider for fixed $j \in [n] \cap (2\mathbb{N}_0 + 1)$ the horizontal strip $[m] \times \{j, j+1\}$. We set

$$v_j^{(1)} := |\{i \in [m] \cap (2\mathbb{N}_0 + 1) : v(i,j) = 1\}|,$$
$$v_j^{(2)} := |\{i \in [m] \cap (2\mathbb{N}_0 + 1) : v(i,j) = 2\}|,$$
$$v_j^{(3)} := |\{i \in [m] \cap (2\mathbb{N}_0 + 1) : v(i,j) = 3\}|.$$

Let us now consider a solution of the proper instance. A count of the block types of the solution in that strip (the variables for the block types are indicated in Figure 7), yields

$$\alpha_j \qquad\qquad + 2\beta_j \qquad\qquad + \beta_j' + \gamma_j + 2\gamma_{j+1} = r_j,$$
$$\alpha_{j+1} + \qquad 2\beta_{j+1} + \beta_j' + 2\gamma_j + \gamma_{j+1} = r_{j+1},$$
$$(6) \qquad \alpha_j + \alpha_{j+1} \qquad\qquad\qquad\qquad\qquad = v_j^{(1)},$$
$$(7) \qquad\qquad \beta_j + \beta_{j+1} + \beta_j' \qquad\qquad\qquad = v_j^{(2)},$$
$$(8) \qquad\qquad\qquad\qquad\qquad \gamma_j + \gamma_{j+1} = v_j^{(3)}.$$

From the system we obtain

$$(9) \qquad \alpha_j + \beta_j - \beta_{j+1} \qquad + \gamma_{j+1} = r_j - v_j^{(2)} - v_j^{(3)},$$
$$(10) \qquad -\alpha_j - \beta_j + \beta_{j+1} + \gamma_j \qquad = r_{j+1} - v_j^{(1)} - v_j^{(2)} - v_j^{(3)}.$$

We will show that for reduced solutions, the variable values of α_j, α_{j+1}, β_j, β_j', β_{j+1}, γ_j, γ_{j+1} are uniquely determined and can actually be computed from the data (i.e., from r_j, r_{j+1}, $v_j^{(1)}$, $v_j^{(2)}$, and $v_j^{(3)}$). This will then allow us to split the task into instances of DR(ν), $\nu \in [3]$.

We begin with a simple observation.

LEMMA 3.6. *Any reduced solution to a proper instance of* DR *satisfies* $\beta_{j+1} = 0$.

Proof. Suppose there is a reduced solution with $\beta_{j+1} > 0$. As we cannot apply any further horizontal local switch of class $\langle t \rangle$, $t \in \{1, 3, 4\}$, we need to have $\alpha_j =$

$\beta_j = \gamma_{j+1} = 0$, hence, by (6) and (8), $\alpha_{j+1} = v_j^{(1)}$, and $\gamma_j = v_j^{(3)}$. Using (9) and (10) we obtain

$$r_j = v_j^{(2)} + v_j^{(3)} - \beta_{j+1},$$
$$r_{j+1} = v_j^{(1)} + v_j^{(2)} + 2v_j^{(3)} + \beta_{j+1},$$

which imply $r_j < v_j^{(2)} + v_j^{(3)} \leq r_{j+1}$, a contradiction to (5). □

Given a proper instance, we distinguish three cases.
- Case 1: Every reduced solution satisfies $\beta_j > 0$.
- Case 2: There is a reduced solution satisfying $\beta_j = 0$, and in any such solution we have $\alpha_j > 0$.
- Case 3: There is a reduced solution satisfying $\beta_j = \alpha_j = 0$.

LEMMA 3.7. *Given a proper instance of* DR. *Then,*

$$
\begin{array}{llll}
\text{Case 1} & \Leftrightarrow & v_j^{(3)} & \leq r_{j+1} < v_j^{(2)} + v_j^{(3)}, \\
\text{Case 2} & \Leftrightarrow & v_j^{(2)} + v_j^{(3)} & \leq r_{j+1} < v_j^{(1)} + v_j^{(2)} + v_j^{(3)}, \\
\text{Case 3} & \Leftrightarrow & v_j^{(1)} + v_j^{(2)} + v_j^{(3)} & \leq r_{j+1} \leq v_j^{(1)} + v_j^{(2)} + 2v_j^{(3)},
\end{array}
$$

and in all cases the variable values of α_j, α_{j+1}, β_j, β_j', β_{j+1}, γ_j, γ_{j+1} *are uniquely determined by the data.*

Proof. By Lemma 3.4, a proper instance has a reduced solution. Hence, one of the disjoint Cases 1–3 occurs. Once we have established the implications "⇒" the reverse implications follow, because the stated intervals for the r_j are disjoint (thereby uniquely specifying the case).

Hence it suffices to prove the implications ⇒. Let us begin with Case 1. As we cannot apply any further horizontal local switch of class $\langle t \rangle$, $t \in \{2, 5\}$, we must have $\alpha_{j+1} = \gamma_j = 0$, hence, $\alpha_j = v_j^{(1)}$ and $\gamma_{j+1} = v_j^{(3)}$. Equation (10) implies $\beta_j = v_j^{(2)} + v_j^{(3)} - r_{j+1}$. As $0 < \beta_j \leq v_j^{(2)}$, we thus have $v_j^{(3)} \leq r_{j+1} < v_j^{(2)} + v_j^{(3)}$. Further, (7) yields $\beta_j' = r_{j+1} - v_j^{(3)}$. Hence, all values of the variables are uniquely determined.

In Case 2, as $\beta_{j+1} = \beta_j = 0$, we have $\beta_j' = v_j^{(2)}$ and, as we cannot apply any further horizontal local switch of class $\langle 6 \rangle$, we need to have $\gamma_j = 0$, hence, $\gamma_{j+1} = v_j^{(3)}$. Equation (10) implies $\alpha_j = v_j^{(1)} + v_j^{(2)} + v_j^{(3)} - r_{j+1}$. As $0 < \alpha_j \leq v_j^{(1)}$, we thus have $v_j^{(2)} + v_j^{(3)} \leq r_{j+1} < v_j^{(1)} + v_j^{(2)} + v_j^{(3)}$. Further, (6) yields $\alpha_{j+1} = r_{j+1} - v_j^{(2)} - v_j^{(3)}$. Hence, again, all variables are determined.

In Case 3, as $\beta_{j+1} = \beta_j = \alpha_j = 0$, we have $\beta_j' = v_j^{(2)}$ and $\alpha_{j+1} = v_j^{(1)}$. Equation (10) implies $\gamma_j = r_{j+1} - v_j^{(1)} - v_j^{(2)} - v_j^{(3)}$. As $0 \leq \gamma_j \leq v_j^{(3)}$ we thus have $v_j^{(1)} + v_j^{(2)} + v_j^{(3)} \leq r_{j+1} \leq v_j^{(1)} + v_j^{(2)} + 2v_j^{(3)}$. Further, (8) yields $\gamma_{j+1} = v_j^{(1)} + v_j^{(2)} + 2v_j^{(3)} - r_{j+1}$, and all variable values are determined. □

We are now prepared to give a proof of Theorem 2.1.

Proof of Theorem 2.1. 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 $\mathrm{DR}(\nu)$, $\nu \in \{0, \ldots, 4\}$. The subtasks are then polynomial-time solvable by Lemma 3.1, Corollary 3.2 and Lemma 3.5 (and trivially for $\mathrm{DR}(0)$ and $\mathrm{DR}(4)$). 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.

We start with some preprocessing if necessary. 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 (5) holds.

Suppose, the problem instance is feasible, hence proper. (We return to the infeasible case later.) The instance has a solution and, hence, by Lemma 3.4, a reduced one. Thus, considering a horizontal strip $[m] \times \{j, j+1\}$ with $j \in [n] \cap (2\mathbb{N}_0 + 1)$, one of the Cases 1, 2 or 3 needs to occur. Which of the cases occurs is, by Lemma 3.7, determined by the data, simply by testing whether we have

$$
\begin{aligned}
v_j^{(3)} &\leq r_{j+1} < v_j^{(2)} + v_j^{(3)}, & \text{or} \\
v_j^{(2)} + v_j^{(3)} &\leq r_{j+1} < v_j^{(1)} + v_j^{(2)} + v_j^{(3)}, & \text{or} \\
v_j^{(1)} + v_j^{(2)} + v_j^{(3)} &\leq r_{j+1} \leq v_j^{(1)} + v_j^{(2)} + 2v_j^{(3)}.
\end{aligned}
$$
(11)

In all three cases we know the unique values of the α_j, α_{j+1}, β_j, β_{j+1}, β'_j, γ_j, and γ_{j+1}. Therefore we know for each $\nu \in [3]$ the row sums

$$
\sum_{p:(p,j)\in G(I(\nu))} \xi_{p,j} \qquad \text{and} \qquad \sum_{p:(p,j)\in G(I(\nu))} \xi_{p,j+1}
$$

with

$$
I(\nu) := \{(i,j) \in C(m,n,2) : v(i,j) = \nu\}
$$

for $G(I(\nu))$ according to (1). In particular, we have

$$
\begin{aligned}
\sum_{p:(p,j)\in G(I(1))} \xi_{p,j} &= \alpha_j, & \sum_{p:(p,j)\in G(I(1))} \xi_{p,j+1} &= \alpha_{j+1}, \\
\sum_{p:(p,j)\in G(I(2))} \xi_{p,j} &= 2\beta_j + \beta'_j, & \sum_{p:(p,j)\in G(I(2))} \xi_{p,j+1} &= \beta'_j, \\
\sum_{p:(p,j)\in G(I(3))} \xi_{p,j} &= \gamma_j + 2\gamma_{j+1}, & \sum_{p:(p,j)\in G(I(3))} \xi_{p,j+1} &= 2\gamma_j + \gamma_{j+1}.
\end{aligned}
$$
(12)

Similarly, we obtain the individual vertical constraints. We can therefore decompose the reconstruction problem into the three independent subproblems $\mathrm{DR}(\nu)$, $\nu \in [3]$, which, by Lemma 3.1, Corollary 3.2, and Lemma 3.5, are solvable in polynomial time.

It remains to consider the case that the given instance of DR is infeasible. In this case, the decomposition into the subproblems needs to yield either an infeasible subproblem or the returned "solution" needs to violate one of the constraints. As both are detected in polynomial time, the proof of this theorem is concluded.   □

Note that the above proof is constructive. A corresponding polynomial-time algorithm for solving DR is summarized in Algorithm 1.

We also remark that the solutions returned by the proposed algorithm are always reduced. This is easily seen by (i) verifying that the values α_j, α_{j+1}, β_j, β_{j+1}, β'_j,

---

Algorithm 1 Compute a solution to DR.

1: $I(\nu) := \{(i,j) \in C(m,n,2) : v(i,j) = \nu\}$, $\nu \in [4]_0$.
2: Solve DR(0) with $I := I(0)$ by filling the $B_2(i,j)$, $(i,j) \in I$, with zeros.
3: Solve DR(4) with $I := I(4)$ by filling the $B_2(i,j)$, $(i,j) \in I$, with ones.
4: Reduce the row and column sums accordingly.
5: Determine the row and column sums for the DR(ν) instances, $\nu \in [3]$
   (row sums according to (11), (12); column sums analogously).
6: Solve DR(1) with $I := I(1)$ according to (2).
7: Solve DR(2) with $I := I(2)$ according to Lemma 3.5.
8: Solve DR(3) with $I := I(3)$ according to Corollary 3.2.
9: **if** the returned $\xi^*_{p,q}$, $(p,q) \in [m] \times [n]$, satisfy all constraints **then**
10:    **return** $\xi^*_{p,q}$, $(p,q) \in [m] \times [n]$.
11: **else**
12:    **return** "Instance is infeasible."
13: **end if**

γ_j, and γ_{j+1}, $j \in [n]$, in each of the three cases in (11) (and, correspondingly, for the variables in the vertical strips) do not allow an application of any of the local switches of class $\langle t \rangle$, $t \in [6]$, 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 $\langle 7 \rangle$ can be applied.

Now we turn to uniqueness.

Proof of Theorem 2.2. By Theorem 2.1, either infeasibility is detected or a reduced 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 (Condition 1) there is only one reduced solution, and (Condition 2) every solution is a reduced solution.

The solution returned by the algorithm is reduced. 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 (11), i.e., the definition depends only on the particular values of the $r_1, \ldots, r_n$, and $c_1, \ldots, c_m$ (note that $v_j^{(1)}$, $v_j^{(2)}$, $v_j^{(3)}$ can be directly determined from the input). 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 3.1(ii) and Corollary 3.2(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 the linear program considered in Lemma 3.3, a basic feasible solution v^* of the linear program, i.e., a vertex of the feasible region, 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 $f(x^*) = v^{*T} v^*$. (Note that multiple solutions contain the same number of ones as this is given by the sum of the row and column sums, respectively.) Hence uniqueness for DR(2), and therefore Condition 1, 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., Condition 1 is satisfied. We have shown in Lemma 3.4 that every solution can be reduced by applying a sequence of local switches. Reversing the sequence and the local switches, we thus see that Condition 2 holds if and only if there is no reversed local switch that can be applied to x^*. There are $O(m^2 n^2)$

possible pairs of blocks that need to be checked to form a reversed local switch, hence, Condition 2 can also be checked in polynomial time.   □

4. Data uncertainty. In the proof of Theorem 2.3 we use a transformation from the following $\mathbb{NP}$-complete problem (see [13]):

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: Decide whether there exists 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 [18] 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 [18] 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 [18] 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 2.3. 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 $\mathrm{NDR}(\varepsilon)$ 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 $\xi_{p,q}$. Again, the figures will show the associated pixels $(p, q) + [0, 1]^2$; see Figure 3.

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 9(a) shows the circuit board for our instance.

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 nonzero components $\xi_{p,q}^*$ are only possible in the white bold-framed boxes. The nonzero 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 9(a)).
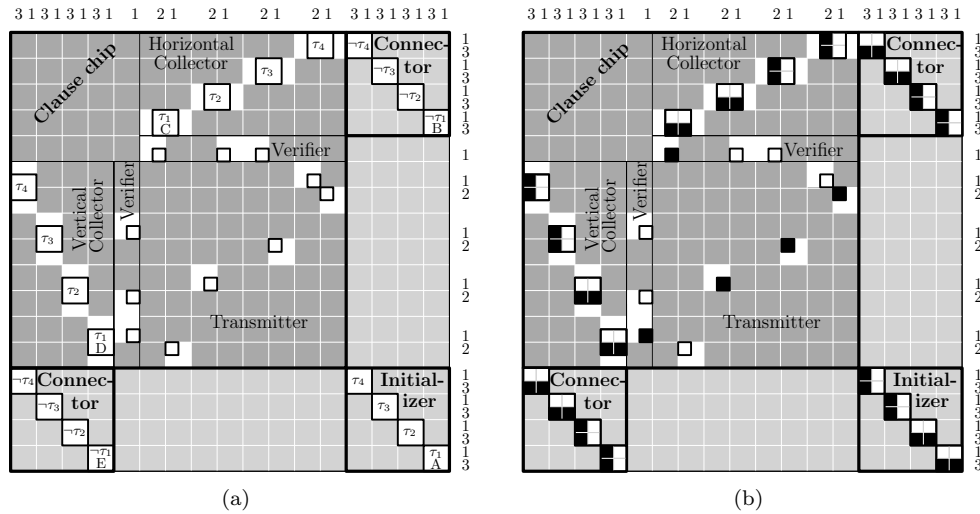
FIG. 9. *Transformation from* 1-IN-3-SAT *for the instance* $\mathcal{I} := (S, T, \mathcal{C}) := (1, 4, \{\tau_1 \vee \neg\tau_2 \vee \tau_3\})$. (a) *The circuit board. By setting to zero suitable blocks and row and column sums we make sure that the nonzero components* $\xi_{p,q}^*$ *of a solution are only possible in the bold-framed boxes within the white blocks in the clause chip, connectors, and the initializer.* (b) *A solution* x^* *(nonzero 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* $\mathcal{I}$.
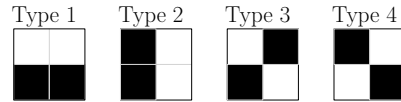

FIG. 10. *The four possible types of blocks for* τ_t- *and* $\neg\tau_t$-*chips,* $t \in [T]$. *In* τ_t-*chips, the types* 1 *and* 2 *represent the Boolean values* TRUE *and* FALSE, *respectively. (Note that the block types* 1– 4 *correspond to the previously introduced types* $\mathfrak{B}_1$, $\mathfrak{B}_{3,1}$, $\mathfrak{B}_{3,3}$, *and* $\mathfrak{B}_{3,4}$, *respectively. The new labeling is more concise in the present context as no grouping of different blocks into a single type is required anymore.)*

A key property of our construction is that the truth assignments will be in one-to-one correspondence with the solutions of the constructed $\text{NDR}(\varepsilon)$ instance if the row and column sum constraints related to the verifiers are left unspecified. The row and column sum constraints related to the verifiers will then 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 10, we set each τ_t-chip, $t \in [T]$, of the initializer to type 1 if $\tau_t^* = \text{TRUE}$, otherwise we set it to type 2.

Let us, as an example, consider $t = 1$ and the sequence of Boolean chips marked A, B, C, D, and E in Figure 9(a); see also Figure 11. 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 in order to satisfy the column sums in the vertical strip. Chip C must 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. In particular, the point of the transmitter in a row intersecting D must be 0. Now, chip D needs to contain two ones since the vertical strip containing chip D contains four ones, two of
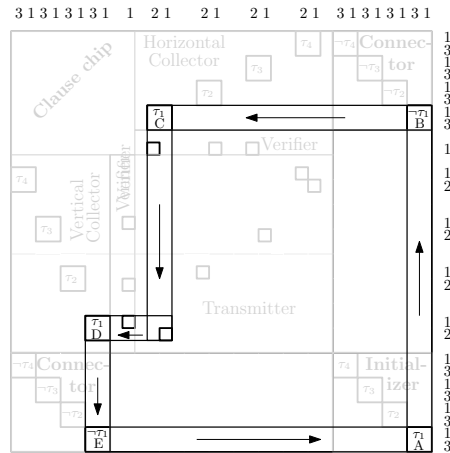
FIG. 11. *The chain of reasoning (vertical transmission) for deducing the types of the Boolean chips marked* A, B, C, D, *and* E *from Figure* 9.

which need to be contained in chip E via block constraints. Hence, we conclude from the row sums that D is of type 1. Considering now the vertical strip intersecting D we see 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, and 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.

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 11).

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 9(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 sth clause is TRUE.

Now, consider the reverse direction for the proof, i.e., suppose a solution x^* to our instance $\mathcal{I}'$ of Figure 9(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
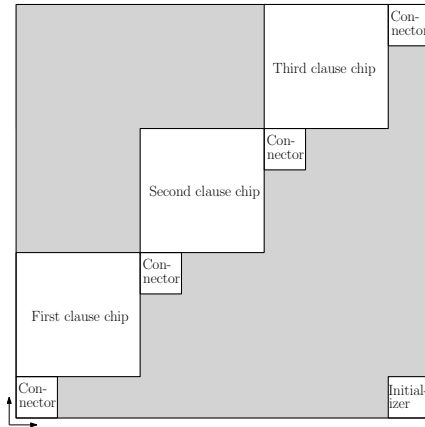
FIG. 12. *The general layout of the circuit board (here for $S = 3$). Nonzero components $\xi^*_{p,q}$ of a solution will be possible at most for (p,q) contained in the white area.*

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 10. 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 neither prescribe the exact number of ones in the τ_t-chips of the horizontal or the vertical collectors, nor in the (s, t)-configurations that are introduced later and which involve the transmitters (see also Figures 13, 14, and 15).

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

Proof of Theorem 2.3. 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 12 illustrates the general layout of the construction.

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

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

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

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

The block $B(a_{S+1} + 2(T - t), 2t - 1)$, $t \in [T]$, 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 9(a).

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

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

The block $B(a_s + 2(T - t), a_s + 2(t - 1))$, $s \in [S + 1]$, $t \in [T]$, will be called $\neg\tau_t$-*chip (of the sth connector)*. For an illustration of a connector, see the bottom left $[8] \times [8]$-box in Figure 9(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 $s \in [S]$. 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 9(a) depicted in dark gray.

For $s \in [S]$ we define the row sums of the horizontal and the column sums of the vertical verifiers (we will call these row and column sums *verifier sums*) by

$$r_{a_s + 6T} := c_{a_s + 2T + 1} := 1,$$
$$r_{a_s + 6T + 1} := c_{a_s + 2T} := 0.$$

The remaining row and columns sums for the transmitters (meeting also certain verifiers and collectors) are for $s \in [S]$ and $l \in [T-1]_0$ defined by

$$r_{a_s+2T+4l} := c_{a_s+2T+4l+2} := 0,$$
$$r_{a_s+2T+4l+1} := c_{a_s+2T+4l+3} := 2,$$
$$r_{a_s+2T+4l+2} := c_{a_s+2T+4l+4} := 1,$$
$$r_{a_s+2T+4l+3} := c_{a_s+2T+4l+5} := 0.$$

Note that sums for the rows and columns meeting the collectors are already defined as there is a transmitter or connector for each row and column meeting a collector (see also Figure 12). 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 $s \in [S]$ and $t \in [T]$, $v_t \in (2\mathbb{Z}+1) \cap \{3-4t, 3-4t+1, \ldots, 4(T-t)+1\}$,

$$f(a_s + 2(T-t), a_s + 2T + 4t - 3 + v_t) := \begin{cases} (\approx, 1) & : & v_t \in \{-1, 1\}, \\ (=, 0) & : & \text{otherwise.} \end{cases}$$

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

Similarly, for the horizontal collector we define for every $s \in [S]$, $t \in [T]$, $u_t \in (2\mathbb{Z}+1) \cap \{3-4t, 3-4t+1, \ldots, 4(T-t)+1\}$,

$$f(a_s + 2T + 4t - 1 + u_t, a_s + 6T + 2t) := \begin{cases} (\approx, 1) & : & u_t \in \{-1, 1\}, \\ (=, 0) & : & \text{otherwise.} \end{cases}$$

The box $(a_s + 2T + 4t - 1, a_s + 6T + 2t) + [1]_0 \times [1]_0$, $t \in [T]$, will be called a τ_t-*chip (of the sth 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 collector, i.e., for $s \in [S]$ 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 sth 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 $N_1 = \{2\}$.

Depending on whether a given variable τ_t appears negated, unnegated, or not at all in the sth clause, we will distinguish three different types of verifier and transmission parts forming a so-called (s, t)-configuration. Formally, for $s \in [S]$ and $t \in [T]$ we refer to the sets of points

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

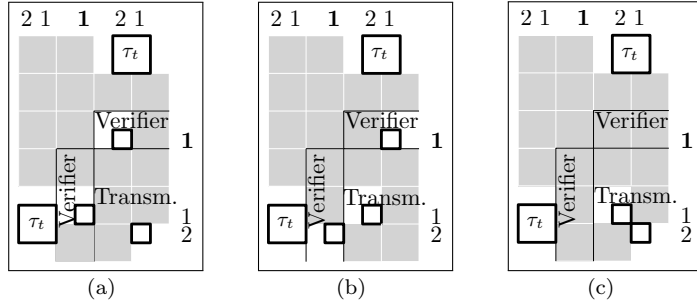as an (s, t)-*configuration*. Figure 13 gives an illustration.

FIG. 13. *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) $t \in [T] \setminus (U_s \cup N_s)$.

For the three different types of (s,t)-configurations, we need to set different blocks to zero. The formal definition is as follows.

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

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

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

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

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

$$f(a_s + 2T + 2 + 2u, a_s + 2T + 2(u-1)) := \begin{cases} (\approx, 1) : (u \in 2\mathbb{N}_0 + 1) \wedge ((u+1)/2 \in U_s), \\ (=, 0) : \text{otherwise,} \end{cases}$$

$$f(a_s + 2T + 2 + 2(u-1), a_s + 2T + 2u) := \begin{cases} (\approx, 1) : (u \in 2\mathbb{N}_0 + 1) \wedge ((u+1)/2 \in N_s), \\ (=, 0) : \text{otherwise.} \end{cases}$$

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

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

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

$$(13) \quad (i,j) \notin ((a_{S+1}, 1) + [2(T-1)]_0^2) \cup ((a_{S+1}, a_{S+1}) + [2(T-1)]_0^2)$$
$$\cup \bigcup_{s \in [S]} (((a_s, a_s) + [2(T-1)]_0^2) \cup ((a_s, a_s + 2T) + [6T+1]_0^2)))$$

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, $(s,t) \in [S] \times [T]$.
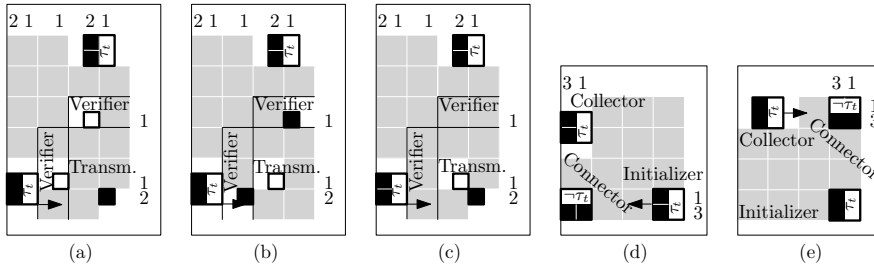
FIG. 14. *Horizontal transmission of a type 2 block. (a,b,c) Transmission through the three differently structured (s,t)-configurations. (d,e) Transmission through a connector. Type 3 and 4 blocks are transmitted in the same way (only the initial block, indicated by the arrow tail, is replaced by the respective type 3 or 4 block).*

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 13. In all three cases the points outside the corresponding Boolean chips in the two collectors 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 from $\{1, 2, 3, 4\}$ shown in Figure 10.

For this, consider a fixed index $t \in [T]$. Each τ_t-chip is contained in a horizontal or vertical strip that, except for a $\neg\tau_t$-chip, contains no other points of $\mathcal{G}$. Since by the block constraints 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. The row and column sums enforce that no two ones are in the rightmost column and upper row of the box of each τ_t-chip. Hence, there are at most the four possibilities shown in Figure 10.

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

To show this, consider again a fixed index $t \in [T]$. Let $V_0, \ldots, V_{4S+2} \subseteq \mathcal{G}$ denote the sequence with (i) $V_0 = V_{4S+2}$ is the τ_t-chip of the initializer, (ii) V_1 is the $\neg\tau_t$-chip of the $(S+1)$th connector, (iii) V_{4s+1} is the $\neg\tau_t$-chip of the $(S-s+1)$th connector, (iv) V_{4s-2} is the τ_t-chip of the $(S-s+1)$th horizontal collector, (v) V_{4s-1} is the set of points in the transmitter of the (s,t)-configuration of the $(S-s+1)$th clause, and (vi) V_{4s} is the τ_t-chip of the $(S-s+1)$th vertical collector, $s \in [S]$; see Figure 11 for the case $S = 1$ and Figure 12 again for the general layout.

By horizontal transmission (see Figure 14) and Claim 2, we have the following implication: if a τ_t-chip of V_{2l}, $l \in [2S]_0$, is of type 2, 3, or 4, then the τ_t-chip of V_{2l+2} is of type 2, and the points of V_{2l+1} are also uniquely determined.

By vertical transmission (see Figure 15) and Claim 2, we have the following implication: if a τ_t-chip of V_{2l}, $l \in [2S+1]$, is of type 1, 3, or 4, then the τ_t-chip of V_{2l-2} is of type 1, and the points of V_{2l-1} are also uniquely determined.

Hence, if we start at an arbitrary τ_t-chip and perform vertical and horizontal transmissions that meet at some Boolean chip we obtain a contradiction unless the τ_t-chip was of type 1 or 2.
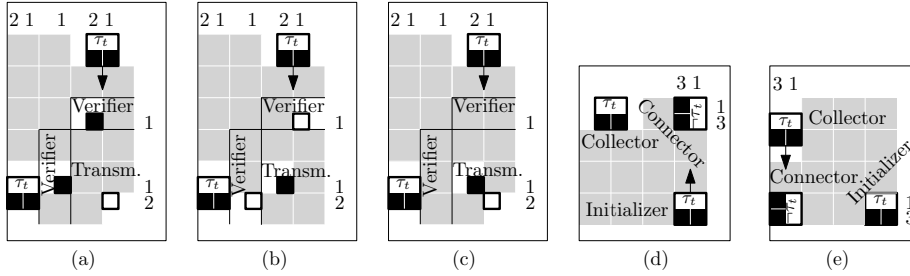
Fig. 15. *Vertical transmission of a type 1 block.* (a,b,c) *Transmission through the three differently structured (s,t)-configurations.* (d,e) *Transmission through a connector. Type 3 and 4 blocks are transmitted in the same way (only the initial block, indicated by the arrow tail, is replaced by the respective type 3 or 4 block).*

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 one-to-one 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, i.e., of the column sum constraints $c_i = 1$ for the vertical verifiers or the row sum constraints $r_i = 1$ for the horizontal verifiers.

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, the chip types in the initializer of a solution to $\mathcal{I}''$ yield 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 one-to-one.

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

This can now be seen by observing that the τ_t-chip, $t \in [T]$, of the sth horizontal and vertical collector, $s \in [S]$, contributes a one to the horizontal and also to the vertical verifier in the sth 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 13, 15(a), (b), (c), and 14(a), (b), (c). The verifier sums thus ensure that exactly one literal in the sth clause is TRUE. This, together with Claim 4, proves the claim.

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

Finally, we note that the transformation runs in polynomial time. $\square$

In section 2 we remarked that $n\mathrm{DR}(\varepsilon)$ can be viewed as a version of DR where small "occasional" uncertainties in the gray levels are allowed. Reviewing the proof of Theorem 2.3, the term "occasional" can be quantified as meaning "of the order of the square root of the number of blocks." More precisely, of $(S(3T+1)+T)^2$ blocks on the circuit board, there are $S(6T+3)+T$ with uncertainty.

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, certainly 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 2.3 is parsimonious (see Claim 5 in the above proof). Hence the problem of deciding whether a given solution of an instance of $\mathrm{NDR}(\varepsilon)$ with $\varepsilon > 0$ has a nonunique solution is $\mathbb{NP}$-complete (which proves Corollary 2.4); for an $\mathbb{NP}$-hardness proof of UNIQUE-1-IN-3-SAT see [18, Lem. 4.2].

Proof of Corollary 2.5. We use a transformation from $\mathrm{NDR}(\varepsilon)$. 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 $\mathrm{NDR}(\varepsilon)$. Note that n and m are even.

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 gives the 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 $\mathrm{NSR}(k, \varepsilon)$. Clearly, the instance $\mathcal{I}$ of $\mathrm{NDR}(\varepsilon)$ admits a solution if and only if the instance $\mathcal{I}'$ of $\mathrm{NSR}(k, \varepsilon)$ 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 corollary. ⬜

5. Final remarks. Recall that in the proof of Theorem 2.1 we have made extensive use of the concept of local switches. Each such switch involves four points. Figure 16 shows, on the other hand, two solutions of a DR instance that differ by 20 1×1-pixels. It is easily checked that these are the only solutions for the given instance.
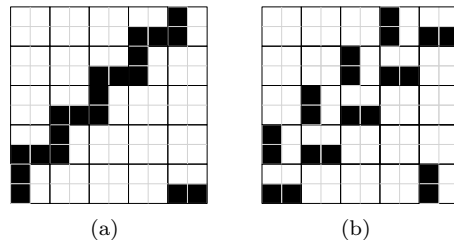

(a)   (b)

FIG. 16. *An example of a large switching component (in the sense of* [31]*) for an instance of* DR. *The solutions shown in* (a) *and* (b) *are the only two solutions for that instance. Both solutions are reduced.*

(Note that we used such a structure in the proof of Theorem 2.3.) Hence, there is 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 [31] (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 $\mathrm{NSR}(k, \varepsilon)$ for any $k \geq 2$ and $\varepsilon > 0$ directly from the construction in [18]. The $\mathbb{NP}$-hardness of the corresponding variants of $\mathrm{NSR}(k, 0)$ for $k \geq 2$, can also be derived from [18]. First the construction is thinned out so as to make sure that each block contains at most one element 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 nonempty block contains exactly one one.

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

Also, it would be interesting to analyze the effect of the results of the present paper on the "degree of ill-posedness" of the underlying problem (see [4]).

Acknowledgment. We are grateful to the anonymous referees for their valuable comments on a previous version of the paper.

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 (2011), pp. 374–376.

[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 (2015), pp. 1016–1028.

[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 (2013), pp. 42–54.

[4] A. Alpers and P. Gritzmann, *On stability, error correction, and noise compensation in discrete tomography*, SIAM J. Discrete Math., 20 (2006), pp. 227–239.

[5] A. Alpers and P. Gritzmann, *Reconstructing binary matrices under window constraints from their row and column sums*, Fund. Inform., 155 (2017), pp. 321–340.

[6] A. Alpers and P. Gritzmann, *Dynamic discrete tomography*, Inverse Problems, 34 (2018), 034003.

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

[8] 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 (2006), pp. 582–588.

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

[10] O. Bill, M. Faouzi, R. Meuli, P. Maeder, M. Wintermark, and P. Michel, *Added value of multimodal computed tomography imaging: Analysis of 1994 acute ischaemic strokes*, Eur. J. Neurol., 24 (2017), pp. 167–174.

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

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

[13] S.A. Cook, *The complexity of theorem-proving procedures*, in Procedings of the 3rd Annual ACM Symposium on the Theory of Computing, ACM, New York, 1971, pp. 151–158.

[14] Z. (W.) Di, S. Leyffer, and S.M. Wild, *Optimization-based approach for joint x-ray fluorescence and transmission tomographic inversion*, SIAM J. Imaging Sci., 9 (2016), pp. 1–23.

[15] C. Dürr, F. Guiñez, and M. Matamala, *Reconstructing 3-colored grids from horizontal and vertical projections is NP-hard: A solution to the 2-atom problem in discrete tomography*, SIAM J. Discrete Math., 26 (2012), pp. 330–352.

[16] D. Gale, *A theorem on flows in networks*, Pacific J. Math., 7 (1957), pp. 1073–1082.

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

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

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

[20] 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 International Symposium on Biomedical Imaging: From Nano to Macro, IEEE, Piscataway, NJ, 2010, pp. 816–819.

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

[22] P. Gritzmann, *On the reconstruction of finite lattice sets from their x-rays*, in Discrete Geometry for Computer Imagery, DCGI'97, Lecture Notes in Comput. Sci. 1347, E. Ahronovitz and C. Fiorio, eds., Springer, Berlin, 1997, pp. 19–32.

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

[24] P.C. Hansen, *Discrete Inverse Problems: Insights and Algorithms*, Fundam. Algorithms 7, SIAM, Philadelphia, 2010.

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

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

[27] R. Hovden, P. Ercius, Y. Jiang, D. Wang, Y. Yu, H. D. Abruña, V. Elser, and D.A. Muller, *Breaking the Crowther limit: Combining depth-sectioning and tilt tomography for high-resolution, wide-field 3D reconstructions*, Ultramicroscopy, 140 (2014), pp. 26–31.

[28] R.W. Irving and M.R. Jerrum, *Three-dimensional statistical data security problems*, SIAM J. Comput., 23 (1994), pp. 170–184.

[29] J.A. Kennedy, O. Israel, A. Frenkel, R. Bar-Shalom, and H. Azhari, *Super-resolution in PET imaging*, IEEE Trans. Med. Imaging, 25 (2006), pp. 137–147.

[30] 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 (1995), pp. 131–155.

[31] 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 (1998), pp. 118–125.

[32] 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 (2009), pp. 69–85.

[33] E. López-Rubio, *Superresolution from a single noisy image by the median filter transform*, SIAM J. Imaging Sci., 9 (2016), pp. 82–115.

[34] J.D. Malone, M.T. El-Haddad, I. Bozic, L.A. Tye, L. Majeau, N. Godbout, A.M. Rollins, C. Boudoux, K.M. Joos, S.N. Patel, and Y.K. Tao, *Simultaneous multimodal ophthalmic imaging using swept-source spectrally encoded scanning laser ophthalmoscopy and optical coherence tomography*, Biomed. Opt. Express, 8 (2017), pp. 193–206.

[35] G.E. Marai, D.H. Laidlaw, and J.J. Crisco, *Super-resolution registration using tissue-classified distance fields*, IEEE Trans. Med. Imaging, 25 (2006), pp. 177–187.

[36] L. Martí-Bonmatí, R. Sopena, P. Bartumeus, and P. Sopena, *Multimodality imaging techniques*, Contrast Media Mol. Imaging, 5 (2010), pp. 180–189.

[37] P. Milanfar, ed., *Super-Resolution Imaging*, CRC Press, Boca Raton, FL, 2010.

[38] J.L. Mueller and S. Siltanen, *Linear and Nonlinear Inverse Problems with Practical Applications*, Comput. Sci. Eng. 10, SIAM, Philadelphia, 2012.

[39] A.D. Parsons, S.W.T. Price, N. Wadeson, M. Basham, A.M. Beale, A.W. Ashton, J.F.W. Mosselmans, and P.D. Quinn, *Automatic processing of multimodal tomography datasets*, J. Synchrotron Rad., 24 (2017), pp. 248–256.

[40] 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 (2007), pp. 313–321.

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

[42] A. Schatzberg and A.J. Devaney, *Super-resolution in diffraction tomography*, Inverse Problems, 8 (1992), pp. 149–164.

[43] M.J. Schrapp and G.T. Herman, *Data fusion in x-ray computed tomography using a superiorization approach*, Rev. Sci. Instrum., 85 (2014), 053701.

[44] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, Chichester, UK, 1986.

[45] 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 (1993), pp. 4150–4153.

[46] 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 (2012), pp. 444–447.

[47] 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 (2011), pp. S117–S125.

[48] 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 (2014), pp. 1181–1193.

[49] 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* 3*D slip velocities and plasma column lengths of a gliding arc discharge*, Appl. Phys. Lett., 106 (2015), 044101.