Utrecht University

Utilising duality in discrete tomography problems

Steven Fleuren

March 2020

MASTER THESIS

Supervisor: Dr. T. van Leeuwen
Second reader: Dr. S. Dirksen

Abstract

We consider the problem of reconstructing binary images given the line sums (projections) of the cell values along a few prescribed directions. This problem is computationally hard, and the projection data can be noisy, so in many cases finding an exact reconstruction is not feasible. For this reason we work with a least-squares formulation of the problem, where the aim is to find a binary image of which the vector of its line sums is close to the given projection data. In this paper, we will mostly focus on preprocessing algorithms that can partially reconstruct the image, assigning values to some of the cells but leaving others undetermined.

The two main approaches we investigated are both based on some form of duality. The first one is based on Langrangian duality: in a recent paper, A. Kadu and T. van Leeuwen introduced a dual formulation of the discrete tomography problem and used the optimal solution of the dual problem to partially reconstruct the binary image. We will further analyse this approach. In particular, we will show that a certain relaxation gives raise to an equivalent dual problem, which opens up ways to compute the dual optimal through different methods. Additionally, we give sufficient conditions under which a non-optimal dual solution enforces the same cell value assignment as the optimal solution.

The approach is based on roof duality, a concept introduced in 1984 as a way to find lower bounds on real, quadratic functions with a binary domain. Additionally, methods that compute the roof duality lower bound can also identify some variable assignments that must hold for any minimiser of the function. While algorithms based on roof duality have become more sophisticated over the years, to our knowledge it has not been applied in the field of discrete tomography so far. We will show that in many cases roof duality based methods will not be able find good partial reconstructions.

Contents

1. Introduction

Discrete tomography concerns the reconstruction of discrete valued images, given the outcome of a finite number of weighted sums of the values of the image's cells. See Figure 1 for an example. There are multiple variations of discrete tomography, many of which can be stated as a problem of the form

$$\text{find an } x \in \mathcal{U}^N \text{ satisfying } Ax = y, \tag{1}$$

where x represents the image we want to reconstruct, $\mathcal{U} = \{u_1, \ldots, u_s\} \subset \mathbb{R}$ gives the allowed cell values, $A \in \mathbb{R}^{M \times N}$ and $y \in \mathbb{R}^N$. In most cases, $M < N$ and A is a sparse, rank deficient matrix. In general, the decision problem of (1) is NP-hard, as are many variants of discrete tomography, see e.g. [8]. This means that for large images finding a solution to (1) is often not feasible. This is one of the reasons to consider to consider a least squares formulation of the problem given by

$$\min_{x \in \mathcal{U}^N} \frac{1}{2} \|Ax - y\|_2^2 . \tag{2}$$

Solving (2) exactly is at least as hard as solving (1), but by using (2) we can get an idea of how close an $x \in \{0,1\}^N$ is to satisfying $Ax = y$. A second important reason to use (2) instead of (1) is that in practise we often have to deal with noisy measurements, and as a result the true solution χ does not exactly satisfy $A\chi = y$, but the value of $\|A\chi - y\|_2^2$ will still be low.

In this paper we will mainly focus on preprocessing techniques. In particular, we are interested in finding $I \subseteq \{1, \ldots, N\}$ and $\alpha_i \in \mathcal{U}$, $i \in I$ such that adding the constraints $x_i = \alpha_i$ does not increase the minimal objective value of (2). This then given a new constrained least squares problem with a matrix of size $M \times (N - |I|)$, which should be easier to solve than the original form.

1.1. Our contributions.

We will investigate three preprocessing techniques. The first one is based on all the size one autarkies of the instance. In short, we can identify all $i \in [N]$ and

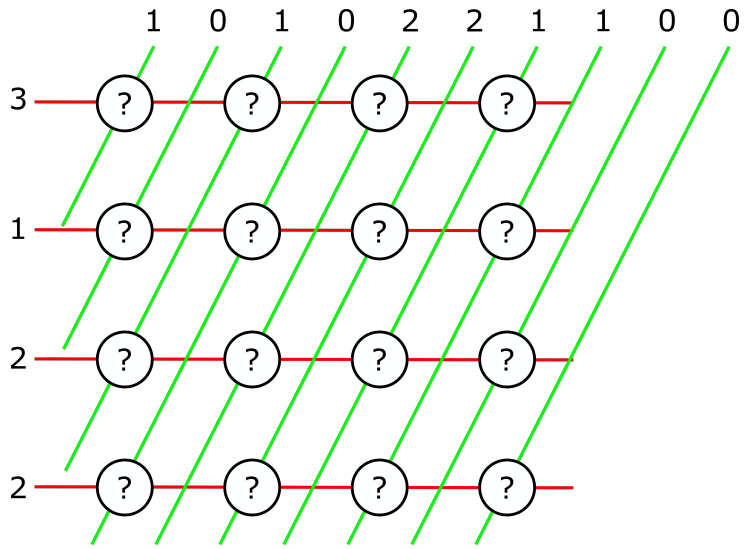


Figure 1. An example of a discrete tomography problem. For each line, the sum of values of the cells is known. Cells values are restricted to 1 (black) or 0 (white). This instance has two exact solutions, shown in Figure 2.

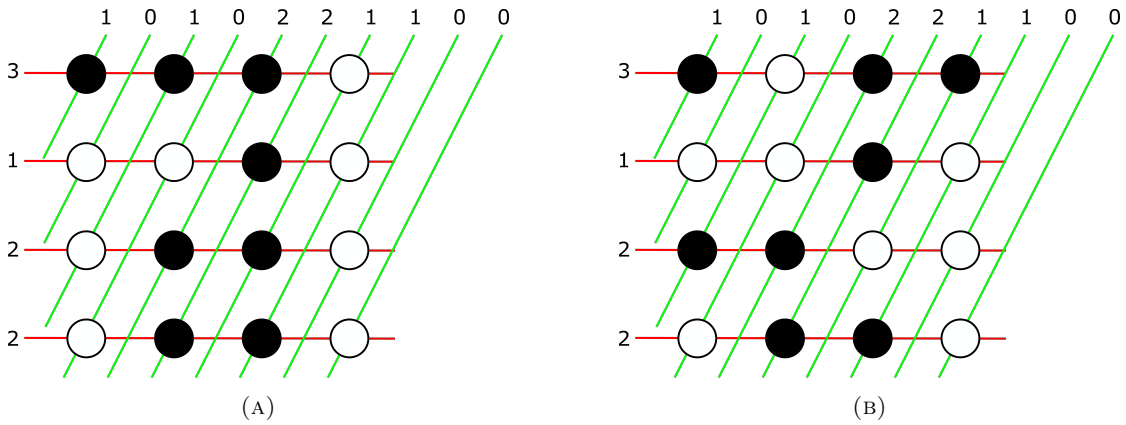


FIGURE 2. The solutions of the discrete tomography problem described by Figure 1.

$\alpha_i \in \{0,1\}$ such that replacing x_i by α_i will lower the objective value of x for all $\{0,1\}^N$. We have not been able to find sources where the method has been used before, but since the result on which it is based is fairly easy we think it is likely that we are not the first to observe it. For the second technique, based on Langrangian duality, we mostly build on the results from [12]. We give some extra insights on how the dual given in [12] relates to (relaxations of) the original primal problem. We show that the the optimal solution of the dual problem is unique, and can be used to derive a lower bound on $\frac{1}{2}\|Ax-y\|_2^2$ when we assume that x satisfies certain constraints. Additionally we extend some of the results to dual variables that are almost optimal. We recreated parts of an experiment performed in [12] to confirm some of our presumptions. The last technique we will discuss is called roof duality. All the algorithms and concept relating to this concept were introduced in other sources ([10], [3], [4], [13]). Our contributions are the results regarding applying these concepts to the discrete tomography problem. We will apply all three techniques to small scale test images to see how well they perform in practise.

1.2. **Outline.** After formalising the discrete tomography problem and other relevant concepts in Section 2, we will introduce a simple preprocessing technique, based on identifying size one autarkies, in Section 3. This technique will generally not reduce the problem size by much, or not at all. However, the computation cost is cheap and the new least squares problem is guaranteed to satisfy some properties that make it easier analyse the techniques described in Section 5.

In Section 4 we will focus on methods based on Langrangian duality [5], which is a fairly general concept that can be applied to many constrained optimisation problems. In [12], the authors derived the Langrangian dual program of an constrained minimisation problem equivalent to (2). Furthermore, they found that they could use the optimal dual variables of this dual program to partially reconstruct a solution of (2). In this paper we will show that the dual program of a relaxation of (2) is equivalent to the dual problem found by the authors of [12]. We then use this equivalence to derive some properties of the dual program: we show that it has a unique optimal solution that can be expressed in terms of A, y and an optimal solution of the relaxation of (2). Furthermore, we give a lower bound on $\|Ax-y\|_2$ for $x \in \mathcal{U}^N$ that do not agree with the partial solution based on the optimal dual variables. We also show how one can use almost optimal dual solutions in this lower bound, which is important in practise since they can be computed numerically. Lastly, we revisit an experiment reported in [12] and explain how our findings are related to their numerical results.

In Section 5 we consider a different technique, based on the concept of roof duality [10]. This technique be applied to problems of the form

$$\min_{x \in \{0,1\}^N} f(x), \tag{3}$$

where f is a quadratic function. Roof duality gives a lower bound on (3) and can sometimes identify assignments of the form $x_i = \alpha_i$ that must hold in optimal solutions of (3). The roof duality lower bound can be formulated in a variety of ways. One way is by constructing a directed graph with edge capacities based on f, and then computing the maximum flow in this graph. This approach was extended in [4], where it was shown that by analysing the strongly connected components of the residual graph one can obtain even more information about the minimum of (3). With their method they were able to obtain optimal solutions for maximum cut problems on graphs derived from Very-large-scale integration (VLSI) circuit chip design, maximum clique problems on instances derived from fault diagnosis and minimum vertex cover problems in random planar graphs of up to 500,000 vertices. However, as we will show, roof duality based techniques are usually not effective when applied to discrete tomography problems.

In Section 6 we give some experimental results we obtained by applying the methods discussed in the previous sections to artificially created discrete tomography problems.

2. Preliminaries

Given a natural number n, we denote the set $\{1, \dots n\}$ by $[n]$. We denote the n-dimensional all ones vector by $\mathbf{1}$ and the all zeros vector by $\mathbf{0}$. The standard basis vectors of $\mathbb{R}^n$ are denoted by $e_1, \dots, e_n$. For matrices, we use I for the identity matrix, E for the all ones matrix and Z for the zero matrix. We will sometimes use subscripts to emphasise the dimension of a matrix, e.g. $Z_{m,n}$ denotes the $m \times n$ zero matrix. For a given matrix A, we write a^i for its i-th column. Given two equally sized real vectors v, w, we say that $v \leq w$ if and only if $v_i \leq w_i$ for all $i \in [n]$; we use $<, \geq$ and $>$ in a similar way. We denote the set of non-negative real numbers by $\mathbb{R}_{\geq 0}$

2.1. **Discrete tomography.** The description of discrete tomography that we will give in this section is mainly based on [11]. We do however make a few adjustments to the notation used to better suit the sequel. The goal of discrete tomography is to reconstruct a d-dimensional image χ of size $N = n_1 \times \cdots \times n_d$. Since the domain of χ is finite, we can enumerate it and consider χ to be an element of $\mathcal{U}^N$, where $\mathcal{U} = \{u_1, u_2\} \subset \mathbb{R}$ gives the allowed cell values. In order to reconstruct χ, we are allowed to use M different *projections* $\mathcal{P} = \{P_1, \dots, P_M\}$. Each projection P_i corresponds to a *ray* R_i, which is a line in $\mathbb{R}^d$ given by

$$R_i = \{\lambda v^i + w^i \mid \lambda \in \mathbb{R}\},$$

where $v^i, w^i \in \mathbb{R}^d$ are constant vectors. Denote the set of all used rays by $\mathcal{R} = \{R_1, \dots, R_M\}$. The projection P_i is given by

$$P_i = \sum_{j \in [N]} c^i_j \chi_j.$$

Here, each c^i_j is defined by a given weight function[1] $w : \mathcal{R} \times [N] \to \mathbb{R}$ as

$$c^i_j := w(R_i, j).$$

How weight function w is chosen depends on what kind of image χ represents. In some settings (see for instance [16]) it is appropriate to assume that each value $j \in [N]$ corresponds to a box in $\mathbb{R}^d$, and that the value $w(R_i, j)$ corresponds with the length of the line piece given by the intersection of the ray R_i and the box corresponding to j. In other settings j corresponds to a point p_j in $\mathbb{R}^d$, and the value of $w(R_i, j)$ is given by

$$w(R_i, j) = \begin{cases} 1 & \text{if } p_j \in R_i, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

We illustrate the distinction in Example 2.1. In both cases, the problem of finding an $x \in \mathcal{U}^N$ that satisfies all the projections can be expressed as

$$\text{find an } x \in \mathcal{U}^N \text{ satisfying } Ax = y, \tag{5}$$

where the rows of $A \in \mathbb{R}^{M \times N}$ are given by $(c^1)^T, \dots, (c^M)^T$ and $y_i = P_i$ for all $i \in [M]$. We should note that while $x = \chi$ is a solution to (2.1), many more solutions could exist, especially when the number of projections is low. In this case it is necessary to make additional assumptions about χ, such as connectedness or convexity, to obtain a meaningful reconstruction.

Example 2.1. In Figure 3 we give two examples of a projection with direction $(1, -1)$. In the image on the left the x_i correspond to points in $\mathbb{R}^2$. The matrix A corresponding to this problem

[1] Note that this weight function is similar to but not the same as the one in [11].

is given by

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

On the right each x_i represents a square, and we have

$$A = \begin{pmatrix} 0 & 0 & c_3^1 & 0 \\ c_1^2 & 0 & c_3^2 & c_4^2 \\ c_1^3 & c_2^3 & 0 & c_4^3 \end{pmatrix},$$

where each c_k^j is proportional to the length of the section of the ray R_k that intersects the square corresponding to x_j.

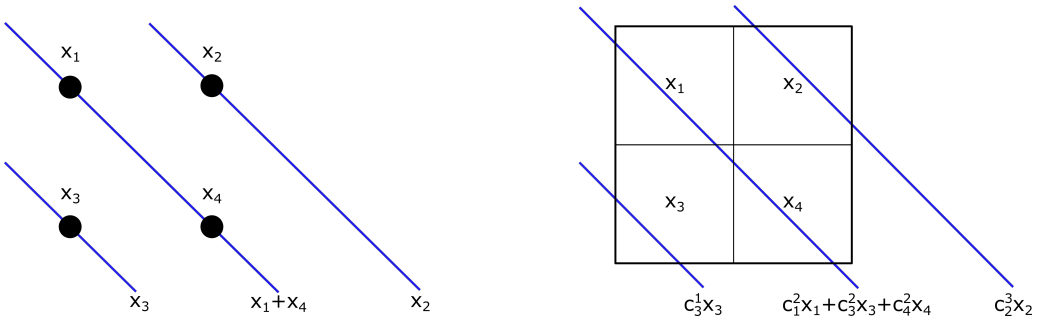


FIGURE 3. Two different types of discrete tomography, see Example 2.1.

For the derivation of most of the techniques in this paper, we do not have to make strong assumptions on the properties of A. However, in a few cases we will work with a specific form of discrete tomography, which we will refer to as the *lattice set reconstruction* problem[2]. This type of discrete tomography has been used in papers as [12], [2], [8]. It is formulated as follows: Let each $j \in [N]$ correspond to a unique point $p_j \in [n_1] \times \cdots \times [n_d]$, and let the weight function be given by (4). The rays are given by m sets of parallel lines $\mathcal{R}_i, i \in [m]$ with direction v^i. None of the directions are parallel to each other. For each direction, all the lines of the form $\{\lambda v^i + w \mid \lambda \in \mathbb{R}\}$ that contain at least one of the points is used. The leftmost image of Figure 3 gives an example of this form of discrete tomography with $m = 1, n_1 = n_2 = 2$ and direction $v^1 = \begin{pmatrix} 1 & -1 \end{pmatrix}^T$. The example we gave in the introduction, Figure 1, is a lattice set reconstruction problem as well. Here the directions are given by $(1,0)^T$ (red) and $(1,2)^T$ (green). In general, the problem of finding an exact reconstruction in this setting is NP-hard if $m \geq 3$ [8].

As we discussed earlier, our main focus will be on a least-squares formulation of (2.1). In particular, we will consider the *binary constrained least-squares* (BCLS) problem given by

$$\begin{array}{ll} \text{minimise} & f_{LS} := \frac{1}{2}\|Ax - y\|_2^2 \\ \text{subject to} & x \in \{0,1\}^N. \end{array} \tag{BCLS}$$

[2]Most papers that use this formulation simply call it the discrete tomography problem, we use a different term to avoid confusion with the more general definition from [11]

Note that we assumed that $\mathcal{U} = \{0, 1\}$. This is without loss of generality, since if $\mathcal{U} = \{u_1, u_2\}$ we can substitute $x = u_1\tilde{x} + u_2(e - \tilde{x})$ in f_{LS} to obtain

$$(u_1 - u_2)^2 \min_{\tilde{x}\in\{0,1\}^N} \frac{1}{2} \left\| A\tilde{x} + \frac{1}{u_1 - u_2}(u_2 Ae - y) \right\|_2^2, \tag{6}$$

which is again a discrete tomography problem with the same matrix A and $\mathcal{U} = \{0, 1\}$. We will call an ordered pair (A, y) with $A \in \mathbb{R}^{M\times N}$ and $y \in \mathbb{R}^M$ an *instance* of minimisation problem BCLS and denote the set of all instances of BCLS by $\mathcal{I}$.

It is worth noting that discrete tomography problems with more than two grey values can also be phrased as a problem of form BCLS, if $\mathcal{U}$ is given by

$$\mathcal{U} = \{u_0, u_0 + \varepsilon, \ldots, u_0 + k\varepsilon\}$$

for some $\varepsilon > 0$. The corresponding least squares problem is then equivalent to

$$\begin{array}{ll} \text{minimise} & \frac{1}{2} \left\| u_0 A\mathbf{1} - y + \varepsilon \sum_{i=1}^k Ax^i \right\|_2^2, \\ \text{subject to} & x^1, \ldots, x^k \in \{0, 1\}^N, \end{array} \tag{7}$$

which can be written as a problem of form BCLS by concatenating $x^1, \ldots x^k$ into a single vector and setting

$$\tilde{A} = \underbrace{\begin{pmatrix} A & A & \cdots & A \end{pmatrix}}_{k \text{ times}}.$$

If the original A has dimensions $M \times N$, then $\tilde{A}$ will be an $M \times kN$ matrix. Hence if the number of grey values is small the methods we will discuss are applicable, but if k is large other approaches might be more suitable.

2.2. **Persistency and autarky.** The methods we discuss in this paper construct *partial solutions*, also called *partial labelings* [14], to BCLS. Here a partial solution is an element $z \in \{0, 1, \emptyset\}^N$, where we use the symbol $\emptyset$ to indicate that the value of $z_i = \emptyset$ is undetermined. We denote $I_z := |\{i \in [N] \mid z_i \neq \emptyset\}|$ and say that the *size* of z is the number of elements in I_z. Given a vector $x \in \{0, 1\}^N$, we define the *fusion* [14] $x[z]$ of x and z as

$$(x[z])_i := \begin{cases} x_i & \text{if } z_i = \emptyset, \\ z_i & \text{otherwise.} \end{cases}$$

We say that a partial solution z is a *strong autarky* [4] for function $f : \{0, 1\}^N \to \mathbb{R}$ if $f(x[z]) < f(x)$ for all $x \in \{0, 1\}^N$ such that $x[z] \neq x$. Similarly, z is a *weak autarky* for f if $f(x[z]) \leq f(x)$ for all $x \in \{0, 1\}^N$. Related is the concept of *persistency*, introduced in [10]. Given some $i \in [N]$ and $\alpha \in \{0, 1\}$ we say that the assignment $x_i := \alpha$ is weakly (strongly) persistent if $x_i = \alpha$ for some minimiser of f. Furthermore, we say that a partial solution z is persistent if for each $i \in I_z$, the assignment $x_i := z_i$ is strongly persistent. Note that if z is a strong autarky, then it is also persistent.

The goal if the methods we will discuss is to find persistent partial solutions of BCLS, which allows us to reduce the original problem to a smaller one. More precisely, given a persistent or autark partial solution z, let $r_z : \mathcal{I} \to \mathcal{I}$ (recall that $\mathcal{I}$ denotes the set of all instances) be its *problem reduction*, that is

$$r_z(A, y) = (r_z^1(A), r_z^2(y))$$

where r_z^1 deletes the columns of A corresponding to the determined entries of z and

$$r_z^2(y) = y - \sum_{i\in I_z} z_i a^i.$$

Solving BCLS for the smaller instance $r_z(A, y)$ then also gives a solution for the original instance (A, y).

Example 2.2. Consider the instance (A, y) given by

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}, \; y = \frac{1}{5}\begin{pmatrix} 6 \\ 2 \\ 6 \\ 2 \end{pmatrix}.$$

This matrix corresponds to two horizontal and two vertical projections of a 2×2 image. The partial solution $z = (\emptyset, \emptyset, \emptyset, 0)$ is strongly autark. This can be shown by computing $f(x) - f(x[z])$ for all $x \in \{0, 1\}^3 \times \{1\}$. There is also an easier way to show that $z = (\emptyset, \emptyset, \emptyset, 0)$ is strongly autark, as we will show in Section 3. Since z is strongly autark, we can reduce the instance to

$$r_z(A, y) = \left(\begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \frac{1}{5}\begin{pmatrix} 6 \\ 2 \\ 6 \\ 2 \end{pmatrix} \right).$$

2.3. **The pseudo-inverse.** Given a real[3] matrix A, we denote its *(Moore-Penrose) pseudo-inverse* by $A^\dagger$. For readers unfamiliar with this concept we state the properties of $A^\dagger$ that we will use later. We use the following definition of $A^\dagger$:

Definition 2.3 ((10.11) - (10.14) in [1])**.** Given an $M \times N$ matrix A, we say that $A^\dagger \in \mathbb{R}^{N \times M}$ is the pseudo-inverse of B if it satisfies the following conditions:

(i) $AA^\dagger A = A$;
(ii) $A^\dagger AA^\dagger = A^\dagger$;
(iii) $(AA^\dagger)^T = AA^\dagger$;
(iv) $(A^\dagger A)^T = A^\dagger A$.

One can show that there is a unique matrix that satisfies these conditions. Note that there exist many equivalent definitions of the pseudo-inverse, the one we mentioned here has the advantage that it is easy to verify if a matrix B is the pseudo-inverse of A. One important property of $A^\dagger$ is the following:

Proposition 2.4 ([15])**.** *Let $A \in \mathbb{R}^{M \times N}$, $z \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$. Then $\|Az - y\|_2 \leq \|Ax - y\|_2$ for all $x \in \mathbb{R}^N$ if and only if*

$$z = A^\dagger y + (I - A^\dagger A)w. \tag{8}$$

for some $w \in \mathbb{R}^N$.

Using properties (i) - (iv) of Definition 2.3 one can derive the following additional properties of the pseudo-inverse ((10.17) - (10.20) in [1]):

$$(A^\dagger)^\dagger = A;$$
$$(A^T)^\dagger = (A^\dagger)^T;$$
$$(kA)^\dagger = k^\dagger A^\dagger \text{ for any } k \in \mathbb{R};$$
$$(A^T A)^\dagger = A^\dagger (A^T)^\dagger.$$

One can use these properties to derive that

$$(A^\dagger)^T (I - A^\dagger A) = (A^\dagger)^T - \left((A^\dagger A)^T A^\dagger\right)^T = (A^\dagger)^T - \left(A^\dagger AA^\dagger\right)^T = (A^\dagger)^T - (A^\dagger)^T = Z_{M,N}.$$

[3]The properties we discuss can be generalised to complex matrices, but we only need to work with real matrices in this paper.

As a result, any z given by (2.4) satisfies $\|z\|_2 \geq \left\|A^\dagger y\right\|_2$, and equality holds if and only if $w = 0$.

The pseudo-inverse can also be used to compute solutions to linear systems, as is seen in the following proposition:

Proposition 2.5 ([15])**.** *Let $A \in \mathbb{R}^{M\times N}$, $z \in \mathbb{R}^N$ and $y \in \mathbb{R}^M$. Then $Az = y$ if and only if $y \in \text{Range}(A)$ and*

$$z = A^\dagger y + (I - A^\dagger A)w$$

for some $w \in \mathbb{R}^N$.

Proof. Suppose $y \in \text{Range}(A)$ and z satisfies (2.4). Then $y = Ax$ for some $x \in \mathbb{R}^N$, and we have

$$Az = A\left(A^\dagger y + (I - A^\dagger A)w\right) = AA^\dagger Ax + (A - AA^\dagger A)w = Ax = y.$$

Now suppose $Az = y$. By Proposition 2.4, z must satisfy (2.4). Furthermore, since $Az \in \text{Range}(A)$, so is y. □

The properties described above will be used in some proofs in Section 4.

3. Size one autarkies

In this section we will introduce a simple prepossessing algorithm, based on identifying all size one autarkies. This method will not be able to reduce the problem size by much in most instances, for reasons that will become clear later. However, by applying the algorithm we can give some guarantees on the resulting reduced problem that will simplify the analysis of the roof duality based method we will describe later. In this section we will assume that A is entry-wise non-negative. We can find size one autarkies by using the following lemma:

Lemma 3.1. *Let z be a partial solution of size 1, and let $k \in [N]$ be such that $z_k \neq \emptyset$. Assume that $A_{ij} \geq 0$ for all $i \in [M]$, $j \in [N]$. Then z is a weak autarky for f_{LS} if and only if*

$$(1 - 2z_k) \left\langle y, a^k \right\rangle + z_k \left\langle A\mathbf{1}, a^k \right\rangle \leq \frac{1}{2} \left\| a^k \right\|_2^2 .$$

Additionally, z is a strong autarky if and only if the above inequality is strict.

Proof. We only prove the weak autarky case since the proof for strong autarky is almost identical. We have to show that

$$\min_{x \in \{0,1\}^N} f_{LS}(x) - f_{LS}(x[z]) \geq 0.$$

Let $\tilde{A} \in \mathbb{R}^{M \times N-1}$ be the matrix given by deleting the k-th column from A. Note that if $x_k = z_k$, then $f_{LS}(x) - f_{LS}(x[z]) = 0$, so we can assume without loss of generality that

$$\min_{x \in \{0,1\}^N} f_{LS}(x) - f_{LS}(x[z]) = \min_{\tilde{x} \in \{0,1\}^{N-1}} \frac{1}{2} \left\| \tilde{A}\tilde{x} + (1 - z_k)a^k - y \right\|_2^2 - \frac{1}{2} \left\| \tilde{A}\tilde{x} + z_k a^k - y \right\|_2^2 .$$

We derive

$$\begin{aligned}
&\frac{1}{2} \left\| \tilde{A}\tilde{x} + (1 - z_k)a^k - y \right\|_2^2 - \frac{1}{2} \left\| \tilde{A}\tilde{x} + z_k a^k - y \right\|_2^2 = \\
&(1 - z_k) \left(\tilde{A}\tilde{x} - y \right)^T a^k + (1 - z_k)^2 \frac{1}{2} \left\| a^k \right\|_2^2 - z_k \left(\tilde{A}\tilde{x} - y \right)^T a^k - z_k^2 \frac{1}{2} \left\| a^k \right\|_2^2 = \\
&(1 - 2z_k) \left(\tilde{A}\tilde{x} - y \right)^T a^k + (1 - 2z_k) \frac{1}{2} \left\| a^k \right\|_2^2 .
\end{aligned}$$

Using the fact that A is non-negative, we have

$$\min_{\tilde{x} \in \{0,1\}^{N-1}} \left(\tilde{A}\tilde{x} - y \right)^T a^k - \frac{1}{2} \left\| a^k \right\|_2^2 = -y^T a^k + \frac{1}{2} \left\| a^k \right\|_2^2$$

and

$$\min_{\tilde{x} \in \{0,1\}^{N-1}} - \left(\tilde{A}\tilde{x} - y \right)^T a^k + \frac{1}{2} \left\| a^k \right\|_2^2 = \left(y - \tilde{A}\mathbf{1} \right)^T a^k - \frac{1}{2} \left\| a^k \right\|_2^2 = (y - A\mathbf{1})^T a^k + \frac{1}{2} \left\| a^k \right\|_2^2$$

Combining the two expressions we find

$$\min_{x \in \{0,1\}^N} f_{LS}(x) - f_{LS}(x[z]) = (2z_k - 1) \left\langle y, a^k \right\rangle - z_k \left\langle A\mathbf{1}, a^k \right\rangle + \frac{1}{2} \left\| a^k \right\|_2^2$$

and the result follows. □

Example 3.2. We apply Lemma 3.1 to the instance from our earlier example: Let instance (A, y) be given by

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}, \; y = \frac{1}{5} \begin{pmatrix} 6 \\ 2 \\ 6 \\ 2 \end{pmatrix}.$$

The partial solution $z = (\emptyset, \emptyset, \emptyset, 0)$ is strongly autark, since

$$\langle y, a^4 \rangle = \frac{4}{5} < 1 = \frac{1}{2} \left\| a^k \right\|_2^2 .$$

We can reduce the instance to

$$r_z(A, y) = \left(\begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \frac{1}{5} \begin{pmatrix} 6 \\ 2 \\ 6 \\ 2 \end{pmatrix} \right) .$$

Note that we can compute $A\mathbf{1}$ in $\mathcal{O}(MN)$ time. If A is non-negative, and after computing $A\mathbf{1}$, we can use the Lemma above to check if a partial solution of size 1 is an autarky for f_{LS} in $\mathcal{O}(M)$ time. Since there are $2n$ of such partial solutions, we can find all autarkies of size 1 in $\mathcal{O}(MN)$ time. We can then combine the autarkies we found into a single autark partial solution z and reduce instance (A, y) to $r_z(A, y)$. This new instance might have new size 1 autarkies! We can repeat the procedure until we end up with an instance (A', y') that does not have a size 1 autarky. This instance must then satisfy

$$\frac{1}{2} \left\| a' \right\|_2^2 < \langle y', a' \rangle < \langle A'\mathbf{1}, a' \rangle - \frac{1}{2} \left\| a' \right\|_2^2$$

for all columns a' of A'. Being able to make this assumption above will prove useful in Section 5. In the context of discrete tomography, a column of a^i of A will only violate the inequalities above if the rays that p_i contributes either all have a high ($y_j \approx (A\mathbf{1})_j$) projection value or all have a low ($y_j \approx 0$) one. In practise most points p_i contribute to some rays of which the projection value is not near either of the two extremes, in which case x_i cannot be determined by this method.

4. Langrangian duality

Langrangian duality is a technique for finding bounds and/or solutions for constrained optimisation problems. The main idea is to add linear penalty terms to the objective function as a substitute for the original rigid constraints. In the minimisation case, the optimal value of the relaxed problem will be a lower bound for the original problem. By multiplying each linear penalty term a corresponding weight we can alter this lower bound. It is generally of interest to find the weights that maximise this lower bound; this is called the Langrangian dual problem. In [12], Van Leeuwen and Kadu gave an equivalent formulation of BCLS and derived an explicit form of its dual problem. They found that for small problems with a unique and exact solution they could extract optimal solutions of BCLS from the optimal dual solutions, and conjectured that this approach also works for bigger problems. In this section we will further analyse this approach.

In the next subsection we introduce some general concepts related to convex optimisation and Langrangian duality that will be useful in the succeeding part. After that, in Section 4.2, we show that the Langrangian dual problem of a certain relaxation of BCLS is equivalent to the dual problem derived in [12]. In Section 4.4 we show that this equivalence can be used to prove some properties of the dual problem and to compute its optimal values.

4.1. **Convex optimisation concepts.**

Here we will explain some of the theory regarding convex optimisation that we will apply to the binary tomography problem in the remainder of this section. We use [5] as our main source for this part. Consider an optimisation problem of the form

$$\begin{array}{ll} \text{minimise} & p(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i \in [s], \\ & h_i(x) = 0, \quad i \in [t]. \end{array} \tag{9}$$

We say that $x \in \mathbb{R}^N$ is a *feasible* solution of (9) if x satisfies its constraints. A feasible solution x^* is called *optimal* if $p(x^*) \leq p(x)$ for all feasible solutions x. The *Langrangian* of (9) is defined as

$$\mathcal{L}(x, \lambda, \nu) := p(x) + \sum_{i \in [s]} \lambda_i f_i(x) + \sum_{i \in [t]} \nu_i h_i(x).$$

We say that λ_i (c.q. ν_i) is the *dual variable* corresponding to the constraint $f_i(x) \leq 0$ (c.q. $(h_i(x) = 0)$. Note that if x is a feasible solution of (9) and $\lambda \geq 0$, then $p(x) \geq \mathcal{L}(x, \lambda, \nu)$. In order to find a lower bound for every feasible x we define the *(Langrange) dual function* as

$$g(\lambda, \nu) := \inf_{x \in \mathbb{R}^N} \mathcal{L}(x, \lambda, \nu).$$

Since g is a pointwise infimum of set of affine functions, g itself is concave. The corresponding *dual problem* is finding the dual variables that give the best lower bound, i.e.

$$\begin{array}{ll} \text{maximise} & g(\lambda, \nu) \\ \text{subject to} & \lambda_i \geq 0, \quad i \in [s] \end{array} \tag{10}$$

In this context we refer to (9) as the *primal problem* and to $x_1, \ldots, x_n$ as the *primal variables*.

Let p^* be the optimal value of (9) and let d^* be the optimal value of (10). In general, we have $d^* \leq p^*$. This relation is often referred to as *weak duality*. In some cases $d^* = p^*$, in which case we speak of *strong duality*. For certain classes of optimisation problems we can guarantee that strong duality holds. Slater's condition says that if (9) is a *convex optimisation problem* (i.e. $p, f_1, \ldots, f_s$ are convex, $h_1, \ldots, h_t$ are affine) and there exists a *strictly feasible point* x (i.e. $f_i(x) < 0$ for all $i \in [s]$ and $h_i(x) = 0$ for all $i \in [t]$), then strong duality holds and, even stronger, there exist x^* and (λ^*, ν^*) such that $p(x^*) = g(\lambda^*, \nu^*)$. See [5] for a proof of this fact.

Now suppose there exists a primal optimum solution x^* and a dual optimum solution (λ^*, ν^*) such that $p(x^*) = g(\lambda^*, \nu^*)$. If p and the constraint functions are all differentiable, then the following conditions hold:

$$f_i(x^*) \leq 0, \qquad i \in [s], \tag{11}$$

$$h_i(x^*) = 0, \qquad i \in [t], \tag{12}$$

$$\lambda_i^* \geq 0, \qquad i \in [s], \tag{13}$$

$$\lambda_i^* f_i(x^*) = 0, \qquad i \in [s], \tag{14}$$

$$\nabla p(x^*) + \sum_{i\in[s]} \lambda_i^* \nabla f_i(x^*) + \sum_{i\in[t]} \nu^* \nabla h_i(x^*) = 0. \tag{15}$$

Conditions (11), (12) and (13) follow directly from the feasibility of x^* and λ^*. The equality (14) is known as *complementary slackness*. It follows from the fact that $g(\lambda^*, \nu^*) \leq \mathcal{L}(x^*, \lambda^*, \nu^*) \leq p(x^*)$. Since $p(x^*) = g(\lambda^*, \nu^*)$ both inequalities hold as equalities, and

$$0 = \mathcal{L}(x^*, \lambda^*, \nu^*) - p(x) = \sum_{i\in[s]} \lambda_i^* f_i(x^*) + \sum_{i\in[t]} \nu_i^* h_i(x^*).$$

The latter sum is zero as a result of condition (12), each term of the former sum is non-negative by (11) and (13), and (14) follows. Condition (15) can be derived from the equation $g(\lambda^*, \nu^*) = \mathcal{L}(x^*, \lambda^*, \mu^*)$, which implies that x^* minimises $\mathcal{L}(\cdot, \lambda^*, \mu^*)$. Hence the gradient of $\mathcal{L}(\cdot, \lambda^*, \mu^*)$ evaluated at x^* is zero.

Together, (11)-(15) are known as the *Karush-Kuhn-Tucker conditions*, or KKT conditions for short. The above shows that the KKT conditions are necessary for x^* and (λ^*, ν^*) to be optimal, assuming differentiability of the objective and constraint function. If additionally (9) is convex, then $\mathcal{L}(\cdot, \lambda, \nu)$ is convex for all dual feasible (λ, ν), and therefore any x and (λ, ν) that satisfy the KKT conditions must be optimal. For this reason the KKT conditions play an important role in algorithms for finding an optimum for this kind of primal problem.

4.2. **Formulation of the dual problem.** By writing BCLS in an appropriate form we can apply the theory described above to the discrete tomography problem. The authors of [12] used a variant[4] of the following formulation:

$$\begin{array}{ll} \text{minimise} & \frac{1}{2}\left\|Ax - y\right\|_2^2 \\ \text{subject to} & x = H(\phi), \\ & \phi \in \mathbb{R}^N, \end{array} \tag{16}$$

where H is the Heaviside function. In [12] it was shown that the dual of (16) is given by

$$\begin{array}{ll} \text{maximise} & \frac{1}{2}y^T y - \frac{1}{2}\left\|\nu - A^T y\right\|^2_{(A^TA)^\dagger} - \sum_{i\in[N]} \max(\nu_i, 0) \\ \text{subject to} & \nu \in \text{Range}(A^T). \end{array} \tag{BCLSD}$$

To show this fact the exact value of $H(0)$ does not have to be the conventional value of $1/2$, the derivation of BCLSD in [12] also works if $H(0) = 1$ (or $H(0) = 0$). For convenience we will use $H(0) = 1$ in the remainder of this paper, in which case (16) is equivalent to BCLS. So in a sense the above can be considered as the Langrangian dual problem of BCLS; as such, we will refer to it as the *Binary Constrained Least Squares Dual* (BCLSD) problem.

[4]The authors of [12] used $\mathcal{U} = \{-1, 1\}$, we transformed their results to our choice of $\mathcal{U} = \{0, 1\}$.

In the sequel, we will consider the following convex optimisation problem, which we will call the *Convex Constrained Least Squares* (CCLS) problem:

$$\begin{array}{ll} \text{minimise} & \frac{1}{2}\|Ax-y\|_2^2 \\ \text{subject to} & \mathbf{0} \leq x \leq \mathbf{1}, \quad i \in [N], \end{array} \tag{CCLS}$$

This problem is a relaxation of BCLS; as such, its objective value gives a lower bound on the objective value of BCLS. We proceed as follows: first we will derive the Langrangian dual problem of CCLS. Next, we show that this dual problem, which we refer to as the *Convex Constrained Least Squares Dual* (CCLSD) problem, is equivalent to BCLSD. Then we will use the concepts introduced in Section 4.1 to show that strong duality holds between CCLS and its dual. This will in turn allow us to apply the KKT conditions to relate optimal solutions of CCLS to optimal solutions of BCLSD and CCLSD.

Proposition 4.1. *The Langrangian dual problem of CCLS is given by*

$$\begin{array}{ll} \textit{maximise} & \frac{1}{2}y^Ty - \frac{1}{2}\left\|A^Ty-(\alpha-\beta)\right\|^2_{(A^TA)^\dagger} - \beta^T\mathbf{1}, \\ \textit{subject to} & \alpha-\beta \in \mathrm{Range}(A^T), \\ & \alpha,\beta \in \mathbb{R}^N_{\geq 0}. \end{array} \tag{CCLSD}$$

Proof. Associate the variables $\alpha, \beta \in \mathbb{R}^N$ with the constraints $\mathbf{0} \leq x$ and $x \leq \mathbf{1}$ respectively. Then the Lagrangian of CCLS is given by

$$\mathcal{L}(x,\alpha,\beta) = \frac{1}{2}\|Ax-y\|_2^2 - \alpha^Tx + \beta^T(x-\mathbf{1}).$$

Since $\mathcal{L}$ is quadratic in x and the quadratic coefficient A^TA is positive semidefinite, x^* is a minimiser of $\mathcal{L}(\cdot,\alpha,\beta)$ if and only if $\nabla\mathcal{L}(x^*,\alpha,\beta) = 0$, i.e, x^* solves

$$A^TAx^* = A^Ty + \alpha - \beta.$$

A solution exists if and only if $\alpha-\beta$ is in the range of A^T, in which case all minimisers are of the form

$$\begin{aligned} x^* &= (A^TA)^\dagger(A^Ty+\alpha-\beta) + (I-A^\dagger A)w \\ &= A^\dagger y + (A^TA)^\dagger(\alpha-\beta) + (I-A^\dagger A)w \end{aligned}$$

for some $w \in \mathbb{R}^N$. So if $(\alpha-\beta) \in \mathrm{Range}(A^T)$ we can find an expression for the dual function $g(\alpha,\beta)$ by substituting a minimiser x^* in $\mathcal{L}(x,\alpha,\beta)$. Substituting $x^* = A^\dagger y + (A^TA)^\dagger(\alpha-\beta)$ in $\|Ax^*-y\|_2^2$ gives

$$\begin{aligned} \|Ax^*-y\|_2^2 &= \left\|(AA^\dagger - I)y + A^{\dagger T}(\alpha-\beta)\right\|_2^2, \\ &= \left\|(AA^\dagger-I)y\right\|_2^2 + \left\|A^{\dagger T}(\alpha-\beta)\right\|_2^2, \\ &= \|y\|_2^2 - y^TAA^\dagger y + \left\|A^{\dagger T}(\alpha-\beta)\right\|_2^2. \end{aligned}$$

Hence, provided $(\alpha - \beta) \in \text{Range}(A^T)$, the dual function g can be written as

$$\begin{aligned} g(\alpha,\beta) &= \inf_{x\in\mathbb{R}^N} \mathcal{L}(x,\alpha,\beta), \\ &= \frac{1}{2}\left\|Ax^* - y\right\|_2^2 - (\alpha-\beta)^T x^* - \beta^T \mathbf{1}, \\ &= \frac{1}{2}\left(\|y\|_2^2 - y^T AA^\dagger y + \left\|A^{\dagger T}(\alpha-\beta)\right\|_2^2\right) - (\alpha-\beta)^T(A^\dagger y + (A^TA)^\dagger(\alpha-\beta)) - \beta^T\mathbf{1}, \\ &= -\frac{1}{2}\left(y^T AA^\dagger y + 2(\alpha-\beta)A^\dagger y + y^T A\left(A^TA\right)^\dagger A^T y\right) - \beta^T\mathbf{1} + \frac{1}{2}y^Ty, \\ &= -\frac{1}{2}\left\|A^Ty - (\alpha-\beta)\right\|_{(A^TA)^\dagger}^2 - \beta^T\mathbf{1} + \frac{1}{2}y^Ty. \end{aligned}$$

If $(\alpha-\beta) \notin \text{Range}(A^T)$ then $\mathcal{L}(\cdot,\alpha,\beta)$ is not bounded from below, so in that case $g(\alpha,\beta) = -\infty$. Thus the Langrangian dual problem of CCLS is given by CCLSD. □

To see that BCLSD and CCLSD are equivalent, note that any optimal solution (α^*,β^*) of CCLSD satisfies $\alpha_i^*\beta_i^* = 0$ for all $i \in [N]$. So given a optimal solution of CCLSD we can construct a solution ν of BCLSD with the same objective value by setting $\nu = \beta^* - \alpha^*$. Conversely, setting $\alpha_i = \max(0,-\nu_i^*)$ and $\beta_i = \max(0,\nu_i^*)$ gives a solution of CCLSD given an optimal solution ν^*.

4.3. **Insights about the dual program.** Before we show that strong duality holds between CCLS and its dual, we will first give some consideration to the structure of BCLSD. Consider BCLSD and eliminate the constraint $\nu \in \text{Range}(A^T)$ by substituting $\nu = A^T\mu, \mu \in \mathbb{R}^M$. This results in the equivalent problem

$$\begin{array}{ll} \text{maximise} & h_{\min}(\mu) := -\frac{1}{2}\left\|A^T\mu - A^Ty\right\|_{(A^TA)^\dagger}^2 - \sum_{i\in[N]}\max\left((A^T\mu)_i, 0\right) + \frac{1}{2}y^Ty \\ \text{subject to} & \mu \in \mathbb{R}^M. \end{array} \tag{17}$$

Given a solution μ of (17) we can retrieve a solution ν of BCLSD by setting $\nu = A^T\mu$. We can express $h_{\min}$ as $h_{\min}(\mu) = \min_{x\in\{0,1\}^N} h(x,\mu)$ where we define

$$h(x,\mu) := -\frac{1}{2}\left\|A^T\mu - A^Ty\right\|_{(A^TA)^\dagger}^2 - x^TA^T\mu + \frac{1}{2}y^Ty$$

and $h_x := h(x,\cdot)$. Using this notation we see that (17) is equivalent to

$$\max_{\mu\in\mathbb{R}^M}\min_{x\in\{0,1\}^N} h(x,\mu). \tag{18}$$

By applying the max-min inequality we find that

$$\max_{\mu\in\mathbb{R}^M}\min_{x\in\{0,1\}^N} h(x,\mu) \le \min_{x\in\{0,1\}^N}\max_{\mu\in\mathbb{R}^M} h(x,\mu).$$

We will show that the right hand side of this inequality is equivalent to the primal problem BCLS.

Since h_x is a concave quadratic function for each $x \in \mathbb{R}^N$, we can determine its maximisers by setting its gradient to zero.

Lemma 4.2. *Given x, the function h_x is maximised by μ^* if and only if*

$$\mu^* = AA^\dagger y - Ax + (I - AA^\dagger)w$$

for some $w \in \mathbb{R}^M$.

Proof. The gradient of h_x is given by

$$\begin{aligned} \nabla h_x &= A(A^TA)^\dagger A^Ty - A(A^TA)^\dagger A^T\mu - Ax, \\ &= AA^\dagger y - AA^\dagger\mu - Ax. \end{aligned}$$

Setting the gradient to zero to find the maximisers of h_x gives

$$AA^\dagger \mu^* = AA^\dagger y - Ax.$$

All solutions of the above are of the form

$$\mu^* = AA^\dagger y - Ax + (I - AA^\dagger)w,\ w \in \mathbb{R}^M.$$

□

Proposition 4.3. *The primal problem BCLS is equivalent to*

$$\min_{x\in\{0,1\}^N} \max_{\mu\in\mathbb{R}^M} h(x,\mu)$$

Proof. If μ^* is a maximiser of h_x then we can express $h(x,\mu^*)$ as

$$\begin{aligned} h(x,\mu^*) &= -\frac{1}{2}\left\|A^T(AA^\dagger - I)(y-w) - A^TAx\right\|^2_{(A^TA)^\dagger} - x^TA^Ty - x^TA^TAx + \frac{1}{2}y^Ty, \\ &= -\frac{1}{2}\left\|A^TAx\right\|^2_{(A^TA)^\dagger} - x^TA^Ty - x^TA^TAx + \frac{1}{2}y^Ty, \\ &= \frac{1}{2}\left\|Ax-y\right\|_2^2. \end{aligned}$$

Hence

$$\min_{x\in\{0,1\}^N} \max_{\mu\in\mathbb{R}^M} h(x,\mu) = \min_{x\in\{0,1\}^N} \frac{1}{2}\left\|Ax-y\right\|_2^2.$$

□

Example 4.4. To illustrate the behaviour of the dual problem we consider the case $A = \begin{pmatrix}1, 0.2\end{pmatrix}$. In Figure 4 we plotted each h_x for y-values 0.5, 1 and 1.5. In the first case there does not exist a binary solution to $Ax = y$, but there do exist solutions in $[0,1]^2$. This means that $g(\alpha,\beta) \leq g(\mathbf{0},\mathbf{0}) = 0$ for all $\alpha,\beta \in \mathbb{R}^N$, so $(\alpha^*,\beta^*) = (\mathbf{0},\mathbf{0})$, $\nu^* = \mathbf{0}$ and $\mu^* = 0$ are optimal solutions of CCLSD, BCLSD and (17) respectively. The best binary solution is $x = (0,1)^T$ with corresponding $\widetilde{\mu}$ given by

$$\widetilde{\mu} = AA^\dagger y - Ax = 0.5 - 0.2 = 0.3.$$

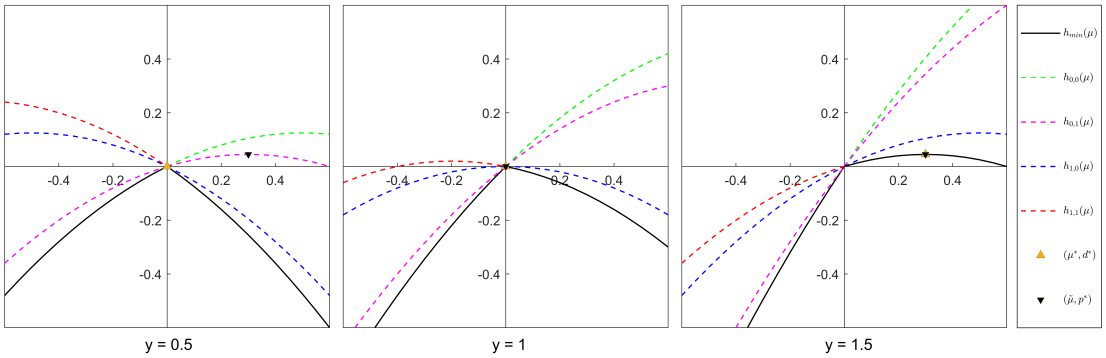


FIGURE 4. Plot of the functions h_x for each $x \in \{0,1\}^2$ and three different values of y. We denote by p^* and d^* the optimal values of respectively BCLS and CCLS.

In the second case $x = (1, 0)^T$ is an exact solution, so $\mu^* = \widetilde{\mu} = 0$. In the third case $x = (1, 1)^T$ is an optimal solution of the relaxed primal CCLS, so we have

$$\mu^* = \widetilde{\mu} = 1.5 - 1.2 = 0.3.$$

Note that only in the first case there is a duality gap between BCLS and BCLSD, while $d^* = p^*$ in the second and third case. We will see in the next section that there is no duality gap if and only if CCLS has a binary optimal solution.

4.4. **Strong duality and its consequences.** We can write CCLS as a optimisation problem of form (9) by setting

$$p = f_{LS},\ f_i(x) = -x_i,\ f_{i+N} = x_i - 1 \text{ for } i \in [N].$$

We see that $p, f_1, \ldots, f_{2N}$ are convex and that $x = \frac{1}{2}\mathbf{1}$ is a strictly feasible point, so Slater's condition holds and there exist x^* and (α^*, β^*) such that $p(x^*) = g(\alpha^*, \beta^*)$. Moreover, p and the constraint function are differentiable and as such the KKT conditions apply. Combining this with the relation $\nu^* = \beta^* - \alpha^*$ immediately leads to two useful results.

Lemma 4.5. *Let x^* be an optimal solution of CCLS. Then the optimal solution ν^* of BCLSD is unique and given by*

$$\nu^* = A^T y - A^T A x^*.$$

Proof. Because of the equivalence of BCLSD and CCLSD, it suffices to show that $\beta^* - \alpha^* = A^T y - A^T A x^*$ if (α^*, β^*) optimises CCLSD. Let (α^*, β^*) be an optimal solution of CCLSD, and recall that both α^* and β^* correspond to inequality constraints. By KKT condition (15) we have

$$\begin{aligned} 0 &= \nabla p(x^*) + \sum_{i \in [N]} \left(\alpha_i^* \nabla f_i(x^*) + \beta_i^* \nabla f_{N+i}(x^*)\right), \\ &= A^T A x^* - A^T y + \sum_{i \in [N]} \left(-\alpha_i^* e_i + \beta_i^* e_i\right), \\ &= A^T A x^* - A^T y + \beta^* - \alpha^*, \end{aligned}$$

hence $\beta^* - \alpha^* = A^T y - A^T A x^*$. □

Lemma 4.6. *If ν^* is an optimal solution of BCLSD and $\nu_i^* \neq 0$ for some $i \in [N]$, then any optimal solution x^* of CCLS must satisfy $x_i^* = H(\nu_i^*)$.*

Proof. Let x^* be an optimal solution of CCLS and let (α^*, β^*) be the optimal solution of CCLSD. By the complementary slackness condition (14) we have

$$\alpha_i^* > 0 \implies x_i^* = 0,\ \beta_i^* > 0 \implies x_i^* = 1.$$

Since the optimal ν^* of BCLSD satisfies $\nu^* = \beta^* - \alpha^*$ the above is equivalent to

$$\nu_i^* \neq 0 \implies H(\nu_i^*) = x_i^*$$

□

In [12], the authors posed the idea to compute the optimal solution ν^* of BCLSD, and then construct a partial solution z of BCLS by setting $z_i = H(\nu_i^*)$ for all $i \in [N]$ such that ν^* is non-zero. The above shows that this approach will not be successful when any optimal solution x^* of BCLS (or CCLS) is a global minimiser, since in that case

$$\nu^* = -\nabla f_{LS}(x^*) = 0.$$

Despite this, the authors of [12] were able to reconstruct noiseless small scale images with their method. At first sight, this seems contradictory. We will address this phenomenon in Section 4.5.

In light of Lemma 4.6, consider the following method of constructing a partial solution of BCLS: first compute an optimal solution x^* of CCLS; then, for each $i \in [N]$ such that $x_i^* \in \{0,1\}$, set $z_i = x_i^*$. All the cell values that are determined by the method in [12] will also be determined by this method, since if $\nu_i^* \neq 0$ then x_i^* is binary. In some sense one can be more confident that setting $z_i = \alpha$ is persistent if the corresponding ν^* is non-zero. However, this is not guaranteed to be the case, at least not for general matrices A as we will show in the following counterexample:

Example 4.7. Set

$$A = \begin{pmatrix} -1 & 3 \\ 12 & -4 \end{pmatrix} \text{ and } y = \frac{1}{5}\begin{pmatrix} 18 \\ 8 \end{pmatrix}.$$

The matrix A is invertible and $x_c := (3/5, 7/5)^T$ satisfies $Ax_c = y$. Since x_c is the unique minimiser of $f_{LS}(x)$, $(x_c)_1 \in (0,1)$ and $(x_c)_2 > 1$, we have $\nu_2^* > 0$. We compute

$$\|A\mathbf{0} - y\|_2^2 = \frac{388}{25}, \quad \|Ae_1 - y\|_2^2 = \frac{3233}{25}, \quad \|Ae_2 - y\|_2^2 = \frac{793}{25}, \quad \|A\mathbf{1} - y\|_2^2 = \frac{1088}{25},$$

so BCLS is minimised by $\bar{x} = \mathbf{0}$, and $\bar{x}_2 \neq H(\nu_2^*)$. See Figure 5 for an illustration.

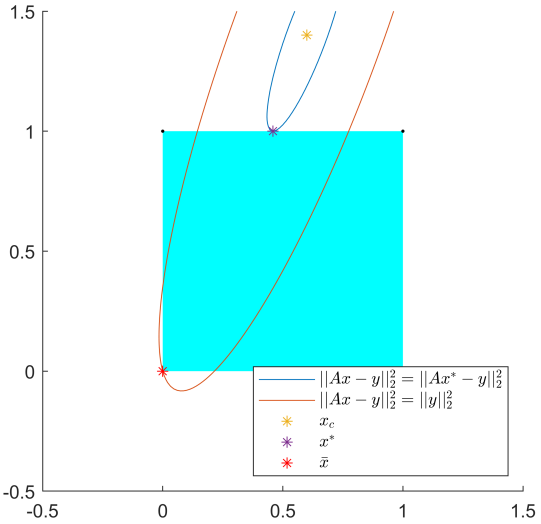


FIGURE 5. Illustration of the constrained least square problem with A and y as in Example 4.7. Here x^* is the optimal solution of CCLS, and $\bar{x}$ is the optimal solution of BCLS

To get an idea of when the situation $\bar{x}_i \neq H(\nu_i^*)$ occurs, it will be useful to give some consideration to the geometry of the example. Given some parameter $q > 0$, the set of solutions $x \in \mathbb{R}^n$ that satisfy $\frac{1}{2}\|Ax - y\|_2^2 = q$ is given by an ellipse centered at x_c. The principal axes of the ellipse are given by the eigenvectors of A^TA and their lengths are inversely proportional to the eigenvalues of A^TA. Since the ellipse is a level set, $\nabla p(x)$ is perpendicular to it at each point x on the ellipse. In particular, $\nu^* = -\nabla p(x^*)$ is perpendicular to the curve given by $\|Ax - y\|_2^2 = \|Ax^* - y\|_2^2$ at x^*. In order to create an example where $0 = \bar{x}_i \neq H(\nu_i^*)$, we need to be able to draw an ellipse that goes through $\bar{x}$, has its centre x_c satisfy $(x_c)_i > 1$ and does not enclose any other point in $\{0,1\}^2$. To be able to achieve this the matrix A must be poorly

conditioned (but not underdetermined because then ν^* would become 0) , and y must be chosen such that the major principal axis of the ellipse intersects the boundary of $[0,1]^2$ near $\bar{x}$.

In order to use ν_i^* to derive a guaranteed persistent partial assignment it needs to be large enough in absolute value. In particular, we can give a lower bound on the value of $f_{LS}(x)$ for x that satisfy $H(-\nu_i^*) = x_i$ for all $i \in I$ for some subset $I \subseteq [N]$. The value of this lower bound depends on the absolute values of the corresponding ν_i^*. Then if we can find a upper bound on the optimal value of BCLS, we can conclude that the optimal solution $\bar{x}$ of BCLS must satisfy $H(\nu_i^*) = \bar{x}_i$ for at least one $i \in I$. We will state the mentioned lower bound in Lemma 4.8. In practise, one often does not have access to the exact value of ν^*, only to an approximation thereof. To account for this we extend the lower bound of Lemma 4.8 to dual variables that are almost optimal. This result is given by Theorem 4.12. After that we give some ideas about how to derive an upper bound on the optimal value of BCLS.

Lemma 4.8. *Let x^* be an optimal solution of CCLS and let ν^* be the optimal solution of BCLSD. Suppose $I \subseteq [N]$ is such that $\nu_i^* \neq 0$ for all $i \in I$. Let $x \in \{0,1\}^N$ satisfy*

$$H(-\nu_i^*) = x_i$$

for all $i \in I$. Then

$$f_{LS}(x) \geq f_{LS}(x^*) + \left(\frac{\sum_{i \in I} |\nu_i^*|}{4 f_{LS}(x^*)} + 1 \right) \sum_{i \in I} |\nu_i^*| \,.$$

Proof. Since x^* is an optimal solution of CCLS, $H(\nu_i^*) = x_i^*$ for all $i \in I$. Let $\Delta x = x - x^*$. Then $\Delta x_i = -\operatorname{sign}(\nu_i^*)$ for all $i \in I$. Note that since $x \in [0,1]^N$ the inequality $\nu_i^* \Delta x_i \leq 0$ holds for all $i \in [N]$. We have

$$\begin{aligned}
\frac{1}{2} \|Ax - y\|_2^2 - \frac{1}{2} \|Ax^* - y\|_2^2 &= \Delta x^T A^T (Ax^* - y) + \frac{1}{2} \|A\Delta x\|_2^2 \,, \\
&= -\Delta x^T \nu^* + \frac{1}{2} \|A\Delta x\|_2^2 \,, \\
&\geq \frac{1}{2} \|A\Delta x\|_2^2 + \sum_{i \in I} |\nu_i^*| \,.
\end{aligned}$$

Furthermore, by the Cauchy–Schwarz inequality we have

$$\Delta x^T A^T (Ax^* - y) \leq \|Ax^* - y\|_2 \, \|A\Delta x\|_2 \,,$$

hence

$$\begin{aligned}
\frac{1}{2} \|Ax - y\|_2^2 &\geq \frac{1}{2} \|Ax^* - y\|_2^2 + \frac{1}{2} \left(\frac{\Delta x^T A^T (Ax^* - y)}{\|Ax^* - y\|} \right)^2 + \sum_{i \in I} |\nu_i^*| \,, \\
&\geq \frac{1}{2} \|Ax^* - y\|_2^2 + \left(\frac{\sum_{i \in I} |\nu_i^*|}{2 \|Ax^* - y\|_2^2} + 1 \right) \sum_{i \in I} |\nu_i^*| \,.
\end{aligned}$$

□

In practise, one often does not have access to the exact values of x^* and ν^*, only to approximations of them. In order to be able to apply the above results effectively, we need to be able to express the results above in terms of such approximations. More concretely, we will introduce the concept of *pseudo-optimality*, and give a generalisation of Lemma 4.8 that utilises this concept. Here, we define pseudo-optimality as follows:

Definition 4.9. Given some $\varepsilon \geq 0$, we say that the primal-dual pair $(\hat{x}, (\hat{\alpha}, \hat{\beta}))$, with $\hat{x}, \hat{\alpha}, \hat{\beta} \in \mathbb{R}^N$, is *$\varepsilon$-pseudo-optimal* if

(i) $\mathbf{0} \leq \hat{x} \leq \mathbf{1}$;
(ii) $\hat{\alpha}, \hat{\beta} \in \mathbb{R}^N_{\geq 0}$;
(iii) $\hat{\alpha}_i \hat{x}_i \leq \varepsilon$ for all $i \in [N]$;
(iv) $\hat{\beta}_i(1 - \hat{x}_i) \leq \varepsilon$ for all $i \in [N]$;
(v) $\hat{\alpha}_i \hat{\beta}_i = 0$ for all $i \in [N]$;
(vi) $\hat{\beta} - \hat{\alpha} = A^T y - A^T A\hat{x}$.

If $\varepsilon = 0$, condition (v) is redundant and the others are simply the KKT optimality conditions for CCLS. Similar to what we did before, we can set

$$\hat{\nu} = \hat{\beta} - \hat{\alpha} = A^T y - A^T A\hat{x}$$

to obtain a solution of BCLSD, which we will relate to ν^* later. For small ε, the value of $f_{LS}(\hat{x})$ can be shown to be close to optimal:

Lemma 4.10. *Let $\varepsilon \geq 0$ and let $(\hat{x}, (\hat{\alpha}, \hat{\beta}))$ be ε-pseudo-optimal. Let x^* be an optimal solution of CCLS. Then*

$$f_{LS}(\hat{x}) \leq f_{LS}(x^*) + N\varepsilon.$$

Proof. Since $\hat{x}$ is feasible for CCLS and $(\hat{\alpha}, \hat{\beta})$ is feasible for CCLSD, we have

$$f_{LS}(x^*) \geq \mathcal{L}(\hat{x}, \hat{\alpha}, \hat{\beta}) = f_{LS}(\hat{x}) - \hat{\alpha}^T - \hat{\beta}^T(\mathbf{1} - x) \geq f_{LS}(\hat{x}) - N\varepsilon.$$

Note that the factor is N instead of $2N$ as a result of condition (v). □

An ε-pseudo-optimal pair can be computed with existing iterative methods. For example, the SciPy library's least squares solver has the option to compute such a pair for a user-provided $\varepsilon > 0$, by setting the `gtol` option to ε, and then using the returned $\hat{x}$ to compute the corresponding $\hat{\alpha}$ and $\hat{\beta}$.

We can relate $\hat{\nu}$ to ν^* as follows:

Lemma 4.11. *Let $\varepsilon \geq 0, I \subseteq [N]$ and let $(\hat{x}, (\hat{\alpha}, \hat{\beta}))$ be ε-pseudo-optimal. Let $\hat{\nu} = \hat{\beta} - \hat{\alpha}$ and let ν^* be the optimal solution of BCLSD. Let M_I be the linear map $\mathbb{R}^N \to \mathbb{R}^I$ given by*

$$(M_I(x))_i = x_i.$$

Then

$$\sum_{i \in I} |\nu_i^*| \geq \sum_{i \in I} |\hat{\nu}_i| - \left\|M_I A^T\right\|_1 \sqrt{2MN\varepsilon}.$$

Proof. We derive

$$\sum_{i \in I} |\nu_i^*| = \|M_I \nu^*\|_1 \geq \|M_I \hat{\nu}\|_1 - \|M_I(\nu^* - \hat{\nu})\|_1 .$$

We will show that $\|M_I(\nu^* - \hat{\nu})\|_1 \leq \left\|M_I A^T\right\|_1 \sqrt{2MN\varepsilon}$. First we observe that

$$\nu^* - \hat{\nu} = A^T y - A^T A x^* - A^T y + A^T A\hat{x} = A^T A(x^* - \hat{x}),$$

where x^* is an optimal solution of CCLS. We have

$$\|M_I(\nu^* - \hat{\nu})\|_1 = \left\|M_I A^T A(x^* - \hat{x})\right\|_1 \leq \left\|M_I A^T\right\|_1 \|A(x^* - \hat{x})\|_1 .$$

We use the ε-pseudo-optimality to derive an upper bound on $\|A(x^* - \hat{x})\|_1$. By Lemma 4.10 we have

$$\begin{aligned} 2N\varepsilon &\geq \|A\hat{x} - y\|_2^2 - \|Ax^* - y\|_2^2, \\ &= \|A(\hat{x} - x^*)\|_2^2 + 2(Ax^* - y)^T A(\hat{x} - x^*), \\ &= \|A(\hat{x} - x^*)\|_2^2 - 2(\hat{x} - x^*)^T \nu^*. \end{aligned}$$

Since $\hat{x}$ is feasible and x^* is optimal, $(\hat{x} - x^*)^T \nu^* \leq 0$. So we have $\|A(x^* - \hat{x})\|_2 \leq \sqrt{2N\varepsilon}$. Using the fact that $\|w\|_1 \leq \sqrt{M}\|w\|_2$ for all $w \in \mathbb{R}^M$ we find that

$$\sum_{i \in I} |\nu_i^*| \geq \sum_{i \in I} |\hat{\nu}_i| - \left\|M_I A^T\right\|_1 \|A(x^* - \hat{x})\|_1 \geq \sum_{i \in I} |\hat{\nu}_i| - \left\|M_I A^T\right\|_1 \sqrt{2MN\varepsilon}.$$

□

Note that if $I = \{i\}$, then $\left\|M_{\{i\}} A^T\right\|_1 = \left\|(a^i)^T\right\|_1 = \left\|a^i\right\|_\infty$, which is the element of the i-th column of A with the largest absolute value. We now have all the tools we need to generalise Lemma 4.8, which in turn will allow us to compute a guaranteed persistent partial solution z.

Theorem 4.12. *Let $\varepsilon \geq 0$ and let $(\hat{x}, (\hat{\alpha}, \hat{\beta}))$ be ε-pseudo-optimal. Suppose $I \subseteq [N]$ is such that $|\hat{\nu}_i| > \left\|a^i\right\|_\infty \sqrt{2MN\varepsilon}$ for all $i \in I$. Let $x \in \{0,1\}^N$ satisfy*

$$H(-\hat{\nu}_i) = x_i$$

for all $i \in I$. Then

$$f_{LS}(x) \geq f_{LS}(\hat{x}) + \frac{L^2}{4 f_{LS}(\hat{x})} + L - N\varepsilon,$$

where $L = \sum_{i \in I} |\hat{\nu}_i| - \left\|M_I A^T\right\|_1 \sqrt{2MN\varepsilon}$

Proof. Let x^* be an optimal solution of CCLS. Note that $\left\|M_{\{i\}} A^T\right\|_1 = \left\|a^i\right\|_\infty$. Since $|\hat{\nu}_i| > \left\|a^i\right\|_\infty \sqrt{2MN\varepsilon}$ for all $i \in I$, by Lemma 4.11 we must have $\nu_i^* \neq 0$ and $\operatorname{sign}(\nu_i^*) = \operatorname{sign}(\hat{\nu}_i)$. So $H(-\nu_i^*) = x_i$ for all $i \in I$, and we can apply Lemma 4.8. Combining this with Lemma 4.10 gives

$$\begin{aligned} \frac{1}{2}\|Ax - y\|_2^2 &\geq \frac{1}{2}\|Ax^* - y\|_2^2 + \left(\frac{\sum_{i \in I} |\nu_i^*|}{2\|Ax^* - y\|_2^2} + 1\right) \sum_{i \in I} |\nu_i^*|, \\ &\geq \frac{1}{2}\|A\hat{x} - y\|_2^2 - N\varepsilon + \left(\frac{\sum_{i \in I} |\nu_i^*|}{2\|A\hat{x} - y\|_2^2} + 1\right) \sum_{i \in I} |\nu_i^*| \end{aligned}$$

Applying Lemma 4.11 to derive a lower bound on $\sum_{i \in I} |\nu_i^*|$ gives the stated result. □

We can apply the bound from Theorem 4.12 to compute a partial solution z as follows:

The partial solution z returned by Algorithm 1 is persistent if u is an upper bound on the optimal objective value of BCLS. One way to compute such an upper bound is by using a cheap algorithm to compute an $\tilde{x} \in \{0,1\}^N$ such that $f_{LS}(\tilde{x})$ is fairly low. Since in order to apply Theorem 4.12, we need to approximate an optimal solution of CCLS, one can consider to construct $\tilde{x}$ based on the approximate solution of CCLS $\hat{x}$. This can be done in multiple ways. A simple approach is to round each $\hat{x}_i$ to the nearest binary value. One can take a randomised approach, by setting $\widetilde{x}_i = 1$ with probability x_i^* and setting $\widetilde{x}_i = 0$ otherwise. One can repeat this process a few times to try different versions of $\widetilde{x}$ and use the one that gives the best upper bound. For our experiments discussed in Section 6, we used a somewhat more involved method, given by Algorithm 2. The idea is that by rounding the variables $\hat{x}_i$ that are close to binary first we obtain a binary solution that is similar to $\hat{x}$.

Algorithm 1

Input: A, y, ε, u.
Output: partial solution z

- Compute an ε-pseudo-optimal primal-dual pair $(\hat{x}, (\hat{\alpha}, \hat{\beta}))$, for instance by using SciPy's least squares solver. A lower ε can possibly provide more persistent assignments, but comes at the cost of longer computation times. Set $\hat{\nu} = \hat{\beta} - \hat{\alpha}$.
- For each $i \in [N]$, compute the value
$$L_i = |\hat{\nu}_i| - \left\|a^i\right\|_\infty \sqrt{2MN\varepsilon}.$$
If this value is larger than zero, we can apply Theorem 4.12 with $I = \{i\}$: if
$$u < f_{LS}(\hat{x}) + \frac{L_i^2}{f_{LS}(\hat{x})} + L_i - N\varepsilon,$$
then set $z_i = H(\hat{\nu}_i)$.

Algorithm 2

Input: A, y, $\hat{x}$
Output: $\widetilde{x} \in \{0,1\}^N$

- Find a permutation $\pi : [N] \to [N]$ such that $\min\left(\hat{x}_{\pi(i)}, 1 - \hat{x}_{\pi(i)}\right) \leq \min\left(\hat{x}_{\pi(j)}, 1 - \hat{x}_{\pi(j)}\right)$ for all $i, j \in [N]$, $i < j$. This can be done with a simple sorting algorithm.
- For each $i \in [N]$, compute
$$c_0 = f_{LS}(\hat{x}_1, \ldots, \hat{x}_{\pi(i)-1}, 0, \hat{x}_{\pi(i)+1}, \ldots, \hat{x}_N)$$
and
$$c_1 = f_{LS}(\hat{x}_1, \ldots, \hat{x}_{\pi(i)-1}, 1, \hat{x}_{\pi(i)+1}, \ldots, \hat{x}_N).$$
If $c_0 < c_1$, set $\hat{x}_{\pi(i)} := 0$, otherwise set $\hat{x}_i := 1$.
- Return $\hat{x}$.

Combining methods that use $\hat{x}$ to compute an upper bound on $f_{LS}(\bar{x})$ with Algorithm 2 gives the method described by Algorithm 3.

So far, we only applied Theorem 4.12 using index sets I of size 1. It is possible to extend the procedure to additionally identify constraints that hold for the optimal value $\bar{x}$. For example, one might find that $L_i > 0$ and $L_j > 0$ but that the two values on their own are not high enough to guarantee persistency. It could be possible that applying Theorem 4.12 with $I = \{i, j\}$ does give a positive result. In that case it is guaranteed that $H(\hat{\nu}_i) = \bar{x}_i$ or $H(\hat{\nu}_j) = \bar{x}_j$. This can be useful if the preprocessing is followed by a method that can make use of extra constraints on the variables.

4.5. **On the conjectures posed in** [12]**.** In this section we will discuss the conjectures given in [12] and how the observations from the previous section relate to them. First we describe the experiment performed in [12]: we consider the lattice set reconstruction problem (see Section 2.1) with $d = 2$ and $n_1 = n_2 =: n$. For $n \in \{2, 3, 4\}$, all the binary images of size $n \times n$ are constructed. For each such image the noiseless projection data is computed for three different

ALGORITHM 3

Input: A, y, ε
Output: Persistent partial solution z

- Compute an ε-pseudo-optimal primal-dual pair $(\hat{x}, (\hat{\alpha}, \hat{\beta}))$, for instance by using SciPy's least squares solver. A lower ε can possibly provide more persistent assignments, but comes at the cost of longer computation times. Set $\hat{\nu} = \hat{\beta} - \hat{\alpha}$.
- Compute a binary $\widetilde{x} \in \{0,1\}^N$ for which the value of $f_{LS}(\widetilde{x})$ is low, for instance by rounding $\hat{x}$ to a binary vector using Algorithm 2.
- For each $i \in [N]$, compute the value

$$L_i = |\hat{\nu}_i| - \left\|a^i\right\|_\infty \sqrt{2MN\varepsilon}.$$

If this value is larger than zero, we can apply Theorem 4.12 with $I = \{i\}$: if

$$f_{LS}(\widetilde{x}) < f_{LS}(\hat{x}) + \frac{L_i^2}{f_{LS}(\hat{x})} + L_i - N\varepsilon,$$

then setting $z_i = H(\hat{\nu}_i)$ preserves the persistency of z.

sets of directions. These sets are give by

$$D^2 = \left\{\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}\right\},\ D^3 = \left\{\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}1\\-1\end{pmatrix}\right\},\ D^4 = \left\{\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}1\\-1\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}\right\}.$$

Then the obtained projection data is used to see if their method is able to reconstruct (parts of) the original images. This method consists of the following steps:

ALGORITHM 4. [12]

Input: A and y
Output: partial solution z

- Find an (approximate) solution $\hat{\mu}$ of the dual formulation (17). This was done in MATLAB, using a convex programming library called CVX [9]. Set $\hat{\nu} = A^T\hat{\mu}$.
- Set

$$z_i = \begin{cases} 1 & \text{if } \hat{\nu}_i \geq 10^{-9}; \\ 0 & \text{if } \hat{\nu}_i \leq -10^{-9}; \\ \text{ø} & \text{otherwise.} \end{cases}$$

- Return z

The result of the experiment was that for all the considered n and direction sets the output z of Algorithm 4 satisfies

$$z_i = \begin{cases} 1 & \text{if } x_i = 1 \text{ for all } x \in \{0,1\}^N \text{ that satisfy } Ax = y; \\ 0 & \text{if } x_i = 0 \text{ for all } x \in \{0,1\}^N \text{ that satisfy } Ax = y; \\ \text{ø} & \text{otherwise.} \end{cases}$$

Note that by construction of y at least one $x \in \{0,1\}^N$ satisfies $Ax = y$. In the previous section, we saw that this implies that the optimal solution ν^* of BCLSD is given by $\nu^* = \mathbf{0}$. This means

that in this case z is determined by the difference between $\hat{\nu}$ and ν^*. We hypothesise that $\hat{\nu}_i$ can be relatively large if $x_i^* \in \{0,1\}$ for all optimal solution x^* of BCLS. The reasoning behind this is that if $\hat{x}$ is ε-pseudo-optimal then the corresponding $\hat{\nu}$ satisfies $|\hat{\nu}_i| \leq \frac{\varepsilon}{\min(\hat{x}_i, 1-\hat{x}_i)}$. So if $\hat{x}_i$ is close to 0 or 1, the corresponding $\hat{\nu}_i$ is allowed to be quite large. In Algorithm 4 $\hat{x}$ is not computed explicitly, but it is plausible that the $\hat{\nu}$ found by the CVX solver approximately satisfies the KKT conditions anyway. As such, we conjecture that for $n \in \{2,3,4\}$, directions sets D^2, D^3 and D^4 and noiseless projection data the following statement holds:

Each optimal solution of CCLS is a convex combination of optimal solutions of BCLS.

It is easy to see that if x^* is a convex combination of optimal solutions of BCLS and y is noiseless, it must be an optimal solution of CCLS: if $\bar{x}^1, \ldots, \bar{x}^k$ are optimal solutions of BCLS and $\lambda_1, \ldots, \lambda_k \in \mathbb{R}_{\geq 0}$ satisfy $\sum_{i\in[k]} \lambda_i = 1$ then

$$A \sum_{i\in[k]} \lambda_i \bar{x}^i = \sum_{i\in[k]} \lambda_i A\bar{x}^i = \sum_{i\in[k]} \lambda_i y = y.$$

The difficulty lies in showing that the optimal solutions of CCLS are contained in the convex hull of optimal solutions of BCLS (or finding an example where this is not the case), and we have not succeeded in finding a conclusive answer for this. However, we were able to verify the following weaker statement (again for $n \in \{2,3,4\}$, directions sets D^2, D^3 and D^4 and noiseless projection data):

If BCLS has a unique optimal solution $\bar{x}$, then $\bar{x}$ is the unique optimal solution of CCLS.

In order to verify the above we use the following lemma:

Lemma 4.13. *Let $\bar{x} \in \{0,1\}^N$ and $x \in [0,1]^N$. If $\bar{x} \neq x$ and $A\bar{x} = Ax = y$, then there exist an $i \in [N]$ and an $x' \in [0,1]^N$ such that $x'_i = 1 - \bar{x}_i$ and $Ax' = y$.*

Proof. Let

$$\lambda' = \max\left\{\lambda \mid \lambda(x - \bar{x}) + \bar{x} \in [0,1]^N\right\} \text{ and } x' = \lambda'(x - \bar{x}) + \bar{x}.$$

We have $\lambda' > 0$ and $Ax' = \lambda A(x - \bar{x}) + A\bar{x} = y$. Furthermore, there must be an $i \in [N]$ such that $x'_i \in \{0,1\}$ and $(\lambda(x - \bar{x}) + \bar{x})_i \neq x'_i$ for all $\lambda < \lambda'$. So $x'_i \neq \bar{x}_i$, and since both are binary, $x'_i = 1 - \bar{x}_i$. □

Our approach is as follows: For each y with a unique binary solution $\bar{x}$, and each $j \in [N]$, we compute an approximate solution $\hat{x}$ to the problem

$$\begin{array}{lll} \text{minimise} & \frac{1}{2}\|Ax - y\|_2^2 & \\ \text{subject to} & \mathbf{0} \leq x \leq \mathbf{1}, & i \in [N], \\ & x_j = 1 - \bar{x}_j. & \end{array} \tag{19}$$

Then we compute dual values $\hat{\alpha}, \hat{\beta} \in \mathbb{R}^N_{\geq 0}$ that satisfy $\hat{\alpha}_i \hat{\beta}_i = 0$ for all $i \in [N]$ and $\hat{\beta} - \hat{\alpha} = A^T y - A^T A\hat{x}$. Then the value of $\mathcal{L}(\hat{x}, \hat{\alpha}, \hat{\beta})$ gives a lower bound on the objective value of (19). If for each $j \in [N]$ this lower bound is larger than zero, by Lemma 4.13 we can conclude that CCLS has $\bar{x}$ as unique optimal solution. By computing a 10^{-5}-pseudo-optimal $\hat{x}$ for each $n \in \{2,3,4\}$, direction set D^2, D^3, D^4 and noiseless y with an unique solution we have found that this is indeed the case for all such discrete tomography problems. To summarise, we have

Proposition 4.14. *If $m, n \in \{2,3,4\}$, A is the projection matrix corresponding to the discrete tomography problem with direction set D^m and y is such that the equation $Ax = y$ has a unique solution $\bar{x} \in \{0,1\}^N$, then CCLS has a unique optimal solution.*

5. Roof duality

We will now shift our focus to *roof duality*, another technique that can be used to find a lower bound and partial solutions of the binary tomography problem (BCLS). Roof duality was introduced in [10] as a means to find an upper bound on *quadratic pseudo-boolean functions*. Here a pseudo-boolean function is a mapping $f : \{0,1\}^N \to \mathbb{R}$. Such a mapping is uniquely represented by a multilinear polynomial

$$f(x) = \sum_{S \subseteq [N]} c_S \prod_{i \in S} x_i,$$

where the c_S, $S \subseteq [N]$ are real numbers. When this polynomial is quadratic we speak of a quadratic pseudo-boolean function. In this case we have

$$f(x) = c_0 + \sum_{i \in [N]} c_j x_i + \sum_{1 \leq i < j \leq N} c_{ij} x_i x_j.$$

In the same paper it was shown that the upper bound given by roof duality can be obtained by some other approaches as well. In order to apply roof duality to binary tomography, we first have to phrase the objective function $f_{LS}(x) = \frac{1}{2} \|Ax - y\|_2^2$ as a quadratic pseudo-boolean function. To do so we have to replace each $\frac{1}{2}(A^T A)_{ii} x_i^2$ term with $\frac{1}{2}(A^T A)_{ii} x_i$. Denote the columns of A by $a^1, \ldots, a^n$. Our objective function becomes

$$f_{PB}(x) = \frac{1}{2} \langle y, y \rangle + \sum_{i \in [N]} \frac{1}{2} \left\langle a^i - 2y, a^i \right\rangle x_i + \sum_{1 \leq i < j \leq N} \left\langle a^i, a^j \right\rangle x_i x_j. \tag{20}$$

While roof duality was introduced to find upper bounds, we are more interested in lower bounds on f and will state the relevant results as such. Keep this in mind when we introduce some of the terminology from [10]. In the next subsection we will describe the original characterisation of roof duality, as well as a maximum flow problem that gives the same lower bound. This maximum flow formulation was introduced in [10] as well, and is considered the most efficient way to compute the roof duality lower bound in most cases [3]. We will also briefly discuss an extension of the maximum flow approach which was introduced in [4]. There are other formulations of roof duality that are less very relevant for our current application; we refer to [3] for a more extensive overview.

5.1. Roof duality concepts.

We will first give the original formulation of the roof dual, and then discuss the maximum flow approach as described in [4].

5.1.1. *Original formulation.* Let $f : \{0,1\}^N \to \mathbb{R}$ be a quadratic pseudo-boolean function and let $h : \mathbb{R}^N \to \mathbb{R}$ be an affine function. We say that h is a *lower plane* of f if $h(x) \leq f(x)$ for all $x \in \{0,1\}^N$. In this case $\min_{x \in \{0,1\}^N} h(x)$ is an easily computeable lower bound on f. When H is a collection of lower planes of f, then

$$\min_{x \in \{0,1\}} f(x) \geq \max_{h \in H} \min_{x \in \{0,1\}} h(x).$$

The idea of roof duality is to choose H to be the collection of roofs of f. These lower planes are generated by constructing a lower plane for each term of f and then adding them. For the constant and linear terms of f we simply use

$$h_0(x) = c_0 \text{ and } h_i(x) = c_i x_i, \ i \in [N].$$

For quadratic terms of the form $c_{ij} x_i x_j$, a lower plane $h_{ij}(x) = a_{ij} x_i + b_{ij} x_j + d_{ij}$ must satisfy

$$d_{ij} \leq 0, \ a_{ij} + d_{ij} \leq 0, \ b_{ij} + d_{ij} \leq 0 \text{ and } a_{ij} + b_{ij} + d_{ij} \leq c_{ij}. \tag{21}$$

We say that h_{ij} is a *tile* if the sum of the slacks in the above inequalities is minimised, that is a_{ij}, b_{ij}, d_{ij} minimise

$$-4d_{ij} - 2a_{ij} - 2b_{ij} + c_{ij}$$

under the constraints (21). Let

$$\mathcal{I}_+ := \{(i,j) : 1 \leq i < j \leq N, c_{ij} > 0\} \text{ and } \mathcal{I}_- := \{(i,j) : 1 \leq i < j \leq N, c_{ij} < 0\}.$$

It was shown in [10] that h_{ij} is a tile if and only if it is of the form

$$h_{ij}(x) = \begin{cases} \lambda_{ij}(x_i + x_j - 1) & \text{if } (i,j) \in \mathcal{I}_+, \\ -\lambda_{ij}x_i + (c_{ij} + \lambda_{ij})x_j & \text{if } (i,j) \in \mathcal{I}_- \end{cases}$$

for some $0 \leq \lambda_{ij} \leq |c_{ij}|$. Then the *roofs* of f are given by

$$\mathcal{R}(f) := \Bigg\{ c_0 + \sum_{i=1}^{N} c_i x_i + \sum_{(i,j)\in\mathcal{I}_+} \lambda_{ij}(x_i + x_j - 1) + \sum_{(i,j)\in\mathcal{I}_-} -\lambda_{ij}x_i + (c_{ij} + \lambda_{ij})x_j : \\ 0 \leq \lambda_{ij} \leq |c_{ij}| \ \ \forall\ 0 \leq i < j \leq N \Bigg\},$$

and the roof dual of f is given by

$$c_{RD}(f) := \max_{h\in\mathcal{R}(f)} \min_{x\in\{0,1\}^N} h(x). \tag{22}$$

Furthermore, any roof h that maximises (22) can be used to infer persistencies: if the coefficient of x_i is larger than zero, then the assignment $x_i := 0$ is strongly persistent, and if this coefficient is smaller than zero the assignment $x_i := 1$ is strongly persistent (Theorem 4.3 in [10]).

5.1.2. *Maximum flow formulation.* We will first state some general theory regarding graph flows, and then explain how this theory can be applied to compute the roof dual. Let $G = (V, E)$ be a directed graph without loops with edge capacities $c : E \to \mathbb{R}_{\geq 0}$. We can assume without loss of generality that G is a complete graph because we can give arcs we do not want to use capacity 0. Given some $s, t \in V$, we say that $\psi : E \to \mathbb{R}$ is an $s - t$ *flow* on G if it satisfies the following three properties:

- $\psi(u, v) \leq c(u, v)$ for all $(u, v) \in E$;
- $\psi(u, v) = -\psi(v, u)$ for all $(u, v) \in E$;
- $\sum_{u\in V:\psi(u,v)>0} \psi(u, v) = \sum_{u\in V:\psi(v,u)>0} \psi(v, u)$ for all $v \in V \setminus \{s, t\}$.

The nodes s and t are commonly referred to as the *source* and the *sink* of the flow. We say that ψ *saturates* an arc (u, v) if $\psi(u, v) = c(u, v)$. The value of ψ, denoted $|\psi|$, is the net amount of flow leaving s (or, equivalently, entering t), that is

$$|\psi| := \sum_{u\in V} \psi(s, u).$$

The maximum flow problem is the problem of finding an $s-t$ flow ψ with maximum value. This can be done in time polynomial in $|V|$ and the number of edges with non-zeros capacity, for instance by using the Edmonds-Karp algorithm [7].

The maximum flow problem is closely related to the minimum cut problem, which we will introduce shortly. Given $s, t \in V$, an $s - t$ cut of G is a partition (P_1, P_2) of V such that $s \in P_1$ and $t \in P_2$. The capacity of a cut is defined as

$$c(P_1, P_2) := \sum_{u\in P_1} \sum_{v\in P_2} c(u, v).$$

One can see this value as an upper bound on the amount of flow between P_1 and P_2. We say that (P_1, P_2) is a minimum $s-t$ cut of G if its capacity is minimised. The problem of finding such a cut is called the minimum cut problem. A fundamental result in the theory of networks flows, known as the max-flow min-cut theorem, is that the capacity of any minimum $s-t$ cut is equal to the value of any maximum $s-t$ flow.

The *residual graph* of $G = (V, E)$ (with arc capacities c) and $s-t$ flow ψ is a graph $R = (V, E)$ with capacities $r_\psi(u, v) := c(u, v) - \psi(u, v)$. One can show that ψ is a maximum $s-t$ flow if and only if there is no path (over edges with non-zero capacity) from s to t in R.

In [10] it was shown that one can use a pseudo-boolean function f to construct a graph such that computing a maximum flow on that graph gives the roof duality lower bound. In order to describe this construction we first need to introduce a few more concepts. Given a binary variable x_i, $i \in [N]$, we define its *complement* as $\bar{x}_i := 1 - x_i$. We denote $L_N := \{x_1, \bar{x}_1, \ldots, x_N, \bar{x}_N\}$. The elements of L_N are often referred to as *literals*. A quadratic pseudo-boolean function can be represented as

$$f(x) = \phi(x, \bar{x}) = d_0 + \sum_{u \in L} d_u u + \sum_{u,v \in L, u \neq v} d_{uv} uv. \tag{23}$$

This representation is not unique, for instance $f(x) = x_1$ and $f(x) = 1 - \bar{x}_1$ represent the same pseudo-boolean function. We say that a representation of form (23) is a *posiform* if $d_u \geq 0$ and $d_{uv} \geq 0$ for all $u, v \in L, u \neq v$. The constant term of a posiform gives a lower bound on f. The largest number c such that a posiform of f with constant term c exists has been shown to be equal to $c_{RD}(f)$. Moreover, the linear part of a posiform with $c_{RD}(f)$ as constant term is a roof of f. Note that given a pseudo-boolean function f one can easily construct a posiform representation by replacing terms $c_{ij} x_i x_j$ with $c_{ij} x_j - c_{ij} \bar{x}_i x_j$ for all negative c_{ij} and subsequently substituting $x_i = 1 - \bar{x}_i$ in the linear terms whenever necessary.

Given a posiform, its *implication network* [4] $G_\phi = (V, E)$ is given by

$$V = L_N \cup \{1, 0\},$$
$$E = \{(u, \bar{v}) \mid d_{uv} > 0\} \cup \{(v, \bar{u}) \mid d_{uv} \neq 0\},$$

with capacities $c(u, \bar{v}) = c(v, \bar{u}) = \frac{1}{2} d_{uv}$. Conversely, given a graph $G = (V, E)$ with V as above one can construct a posiform

$$\phi_G(x) := \sum_{(u,v) \in E} c_{uv} u \bar{v}.$$

We can use this correspondence to compute the roof dual as a maximum flow problem using the following two results:

Proposition 5.1 (Proposition 15 in [3]). *Let $G = (V, E)$ be an implication network and let ψ be a feasible $1-0$ flow of G with value $|\psi|$. Let R be the residual graph of G and ψ. Then*

$$\phi_G(x, \bar{x}) = \phi_R(x, \bar{x}) + |\psi|$$

for all $x \in \{0, 1\}^N$.

Proposition 5.2 (Proposition 16 in [3]). *If ϕ_1, ϕ_2 are posiform representations of f such that the constant term of ϕ_1 is smaller than the one of ϕ_2, then there exists a $0-1$ flow in G_{ϕ_1} with value larger than zero.*

Combining these two results gives the connection the the roof duality lower bound:

Proposition 5.3 (Theorem 10 in [3]). *If ϕ is a posiform representation of f with constant term d_0, and ψ is a maximum flow in the implication network of ϕ, then*

$$c_{RD}(f) = d_0 + |\psi|\,.$$

Furthermore, the resulting residual graph can be used to construct an autark partial solution:

Proposition 5.4 ([4])**.** *Let ϕ be a posiform representation of f and let ψ be a maximum flow in the implication network G_ψ of ϕ. Let R be the residual graph of G_ψ and ϕ. Then the partial solution given by*

$$z_i = \begin{cases} 1 & \text{if there exists a path from node 1 to node } x_i \text{ in } R; \\ 0 & \text{if there exists a path from node 1 to node } \bar{x}_i \text{ in } R; \\ \emptyset & \text{otherwise} \end{cases}$$

is a strong autarky for f.

In [4] it was shown that one can possibly derive even more persistent assignments by analysing the strongly connected components of R. We refer to [4] for details on this approach.

5.2. **Applying roof duality to the discrete tomography problem.** Now that we have established the main concepts of roof duality, we can apply the theory to discrete tomography. Here we will assume that A is entry-wise non-negative. Recall that we then can also assume that

$$\frac{1}{2}\|a\|_2^2 < \langle y, a\rangle < \langle A\mathbf{1}, a\rangle - \frac{1}{2}\|a\|_2^2 \tag{24}$$

for all columns a of A. We can construct an implication network of f_{PB} as follows:

Lemma 5.5. *If A is entry-wise non-negative and $\langle 2a^i, y\rangle \geq \|a^i\|$ for all $i \in [N]$, then*

$$\phi(x) := \frac{1}{2}y^T y + \sum_{i=1}^{N}\left(\frac{1}{2}\|a^i\|_2^2 - \langle a^i, y\rangle\right) + \sum_{i=1}^{N}\left(\langle a^i, y\rangle - \frac{1}{2}\|a^i\|_2^2\right)\bar{x}_i + \sum_{1\leq i<j\leq N}\langle a^i, a^j\rangle x_i x_j$$

is a posiform of f_{PB}. The adjacency matrix of the corresponding implication network is given by

$$\frac{1}{2}\begin{pmatrix} Z_{N,N} & A^TA - \operatorname{diag}(b) & Z_{N,1} & Z_{N,1} \\ Z_{N,N} & Z_{N,N} & Z_{N,1} & A^Ty - \frac{1}{2}b \\ y^TA - \frac{1}{2}b^T & Z_{1,N} & 0 & 0 \\ Z_{1,N} & Z_{1,N} & 0 & 0 \end{pmatrix},$$

where the rows and columns are indexed by $(x_1, \ldots, x_N, \bar{x}_1, \ldots, \bar{x}_n, 1, 0)$

Proof. Since A is entry-wise non-negative and $\langle 2a^i, y\rangle > \|a^i\|$ all the coefficients of ϕ are non-negative, and substituting $\bar{x} = \mathbf{1} - x$ gives f_{PB}, so ϕ is a posiform of f_{PB}. The capacities of the edges are given by

$$c(x_0, x_i) = c(\bar{x}_i, \bar{x}_0) = \frac{1}{2}\langle a^i, y\rangle - \frac{1}{4}\|a^i\|_2^2 = \frac{1}{2}\left(A^Ty - \frac{1}{2}b\right)_i \qquad \text{for all } i \in [N];$$

$$c(x_i, \bar{x}_j) = \langle a^i, a^j\rangle = \frac{1}{2}(A^TA)_{ij} \qquad \text{for } i, j \in [N], i \neq j.$$

This gives the adjacency matrix described above. □

Example 5.6. Consider the lattice set reconstruction problem with a 2×2 image and horizontal and vertical rays. The corresponding projection matrix is

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

Given a $y \in \mathbb{R}^4$, the pseudo-boolean function given by (20) is

$$\begin{aligned} f_{PB}(x) =& \frac{1}{2} y^T y + (1 - y_1 - y_3)x_1 + (1 - y_1 - y_4)x_2 + (1 - y_2 - y_3)x_3 + \\ & (1 - y_2 - y_4)x_4 + x_1x_2 + x_1x_3 + x_2x_4 + x_3x_4. \end{aligned}$$

Assuming that (24) holds for all columns of A. Then f_{PB} can be represented by the posiform

$$\begin{aligned} \phi(x) =& \frac{1}{2} y^T y + 4 - 2y^T \mathbf{1} + (y_1 + y_3 - 1)\bar{x}_1 + (y_1 + y_4 - 1)\bar{x}_2 + (y_2 + y_3 - 1)\bar{x}_3 + \\ & (y_2 + y_4 - 1)\bar{x}_4 + x_1x_2 + x_1x_3 + x_2x_4 + x_3x_4. \end{aligned}$$

The corresponding graph G_ϕ is depicted in Figure 6. The capacities of the edges are

$$\begin{aligned} d_{x_0 x_i} = d_{\bar{x}_i \bar{x}_0} &= \frac{1}{2}(y^T a_i - 1), && i \in \{1, 2, 3, 4\}, \\ d_{x_i \bar{x}_j} = d_{x_j \bar{x}_i} &= \frac{1}{2}, && (i, j) \in \{(1, 2), (1, 3), (2, 4), (3, 4)\}. \end{aligned}$$

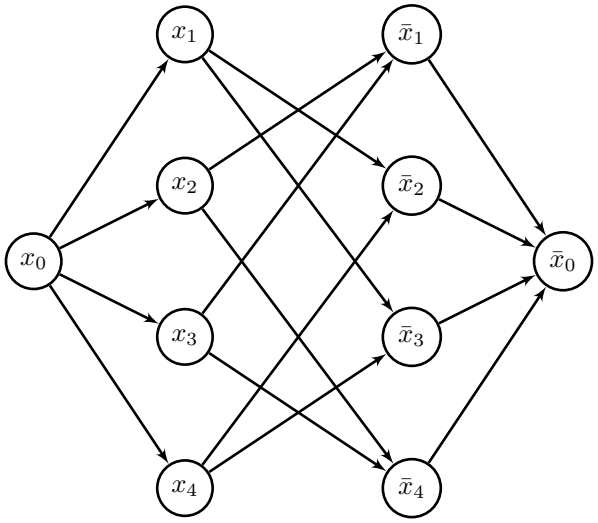


FIGURE 6. The directed graph G_ϕ, where ϕ is defined in Example 5.6.

For the remainder of this section we will focus on the lattice set reconstruction problem. In this case the coefficients of ϕ can be interpreted as follows:

- $\left(A^T y\right)_i$ gives the sum of the projection data of the rays that contain p_i;
- $\left(A^T A\right)_{ij}$ gives the number of rays that intersect both p_i and p_j. In particular $\left\|a^i\right\|_2^2 = \left(A^T A\right)_{ii} = m$ for all $i \in [N]$.

We argue that applying roof duality based methods will often be ineffective in this setting: In the one directional case, roof duality based methods will never be able to find persistent assignments after all size one autarkies are eliminated. In cases with two or more directions, roof duality based methods can only improve on eliminating size one autarkies if there remain y_i such that $y_i \leq \frac{1}{2}$ or $y_i \geq (A\mathbf{1})_i - \frac{1}{2}$. To show this we will use the following lemma:

Lemma 5.7. *Let $n \in \mathbb{N}$, and consider the directed graph $G = (V, E)$ with $V = \{v_1, \ldots, v_{2n}, s, t\}$ and adjacency matrix*

$$\mathcal{A} = \begin{pmatrix} Z_{n,n} & E_{n,n} - I_{n,n} & Z_{n,1} & Z_{n,1} \\ Z_{n,n} & Z_{n,n} & Z_{n,1} & \hat{c} \\ \hat{c}^T & Z_{1,n} & 0 & 0 \\ Z_{1,n} & Z_{1,n} & 0 & 0 \end{pmatrix}, \tag{25}$$

where $0 < \hat{c}_i \leq n-1$, for all $i \in [n]$. Then G has exactly two minimum $s-t$ cuts, given by $(\{s\}, V \setminus \{s\})$ and $(V \setminus \{t\}, \{t\})$, and the capacity of any other cut (P_1, P_2) is at least

$$\hat{c}n + \min(\ell, n - 1 - u)$$

Proof. We see that

$$c(\{s\}, V \setminus \{s\}) = c(V \setminus \{t\}, \{t\}) = \hat{c}n.$$

More generally, let (P_1, P_2) be a $s-t$ cut. Let

$$\alpha = |P_1 \cap \{v_1, \ldots, v_n\}|, \ \beta = |P_1 \cap \{v_{n+1}, \ldots v_{2n}\}| \text{ and } r = |\{i \in I_k \mid v_i \in P_1, v_{i+n} \in P_2\}|.$$

We have $\max(0, \alpha - \beta) \leq r \leq \min(\alpha, n - \beta)$. The value of cut (P_1, P_2) is computed as

$$c(P_1, P_2) = \hat{c}(n - \alpha) + \alpha(n - \beta) - r + \hat{c}\beta.$$

We obtain the value of $c(\{s\}, V \setminus \{s\})$ by setting $\alpha = \beta = 0$ and the value of $c(V \setminus \{t\}, \{t\})$ by setting $\alpha = \beta = n$. We will now assume that $0 < \alpha + \beta < 2n$ and use the fact that $0 < \hat{c} < n - 1$ to show that in this case $c(P_1, P_2) \geq \hat{c}n + \min(\ell, n - 1 - u)$. We have to show that

$$\alpha(n - \hat{c}) \geq \beta(s - \hat{c}) + \max(0, s - t) + \min(\ell, n - 1 - u) \tag{26}$$

for all $\alpha, \beta \in \{0, \ldots, n\}$ such that $0 < \alpha + \beta < 2n$. Given α, the left hand size of (26) is maximised by setting β either as large or as small as possible. This leaves us to check the following cases:

- $\alpha = n, \beta = n - 1$. We have

$$\alpha(n - \hat{c}) = n(n - \hat{c}) = (n-1)(n - \hat{c}) + 1 + (n-1) - \hat{c} \geq \beta(\alpha - \hat{c}) + \max(0, \alpha - \beta) + n - 1 - u.$$

- $\alpha < n, \beta = n$. We have

$$\alpha(n - \hat{c}) = \alpha n - \hat{c}n + (n - \alpha)\hat{c} \geq \alpha n - n\hat{c} + \ell = \beta(\alpha - \hat{c}) + \max(0, \alpha - \beta) + \ell.$$

- $\alpha > 0, \beta = 0$. We have

$$\alpha(n - \hat{c}) \geq n - 1 - \hat{c} + 1 \geq \beta(\alpha - \hat{c}) + \max(0, \alpha - \beta) + n - 1 - u.$$

- $\alpha = 0, \beta = 1$. We have

$$\alpha(n - \hat{c}) \geq \ell - \hat{c} = \beta(\alpha - \hat{c}) + \max(0, s - t) + \ell.$$

So inequality (26) holds and

$$c(P_1, P_2) \geq \hat{c} + \min(\ell, n - 1 - u)$$

for all cuts (P_1, P_2) other than the two minimum cuts. □

We use Lemma 5.7 to show that in the one directional case there will be no paths from node 1 to other nodes in the final residual graph, which means that we can not use Proposition 5.4 to find a non-trivial strong autarky:

Lemma 5.8. *Consider the lattice set reconstruction problem with only one direction. Let the rays be given by $R_1, \dots, R_M$, denote $I_k := \{i \in [N] : p_i \cap R_k \neq \emptyset\}$ and*

$$V_k := \{u \mid u \in \{x_i, \bar{x}_i\}, i \in I_k\}$$

for $k \in [M]$. Suppose

$$\frac{1}{2}\left\|a^i\right\|_2^2 < \left\langle a^i, y\right\rangle < \left\langle A\mathbf{1}, a^i\right\rangle - \frac{1}{2}\left\|a^i\right\|_2^2$$

for all $i \in [N]$. Let ψ be a maximum 1-0 flow of G_ϕ and let R be the residual network of G_ϕ and ψ. Then the strongly connected components of R are given by $\{0\}, \{1\}$, and $V_1, \dots, V_M$.

Proof. Let $k \in [M]$ and consider the subgraph of G_ϕ induced by $V_k \cup \{0, 1\}$. Denote this subgraph by G_k. This subgraph has the same form as the graph described by Lemma 5.7: We set $n = |I_k|$ and can correspond s to node 1, t to node 0, each v_i, $i \in [n]$ to a node $x_j \in V_k$ and each v_{i+n}, $i \in [n]$ to a node $\bar{x}_j \in V_k$. Since $(A^T A)_{ij} = 1$ for all $i, j \in I_k$ and

$$0 < \left\langle a^i, y\right\rangle - \frac{1}{2} < \left\langle A\mathbf{1}, a^i\right\rangle - 1 = n - 1$$

for all $i \in I_k$, we can apply Lemma 5.7 to show that G_k has exactly two minimum $1-0$ cuts, given by $(\{1\}, V_k \cup \{0\})$ and $(\{1\} \cup V_k, \{0\})$. This implies that the strongly connected components of the residual graph of G_k and any of its maximum flows are given by $\{0\}, \{1\}$ and V_k. Combining this with the fact that there are no arcs between V_i and V_j for all $i \neq j$ then gives the result. □

The above shows that in the one directional case, the autarky we find by applying Proposition 5.4 will leave all variables undetermined. The above extends to lattice set reconstruction problems with two or more directions in the following way:

Theorem 5.9. *Consider the lattice set reconstruction problem with $m \geq 2$ non-parallel directions $d_1, \dots, d_m$. Suppose*

$$\frac{1}{2} < y_i < (A\mathbf{1})_i - \frac{1}{2}$$

for all $i \in [M]$. Let ψ be a maximum $1-0$ flow of G_ϕ and let R be the residual network G_ϕ and ψ. Then there is no path from node 1 to any other node in R.

Proof. An important observation is that we can see G_ϕ as the sum of the implication networks arising from the one directional case. To see this note that one can see A as a block matrix

$$A = \begin{pmatrix} A_1 \\ \vdots \\ A_m \end{pmatrix},$$

where the rows of A_i correspond to the projections of direction d_i. We can see y in a similar way, we denote the part of y corresponding to direction d_i by y^i. The adjacency matrix $\mathcal{A}$ of G_ϕ can be written as $\mathcal{A} = \sum_{i=1}^m \mathcal{A}_i$ where

$$\mathcal{A}_i := \frac{1}{2}\begin{pmatrix} Z_{N,N} & A_i^T A_i - I_{N,N} & Z_{N,1} & Z_{N,1} \\ Z_{N,N} & Z_{N,N} & Z_{N,1} & A_i^T y^i - \frac{1}{2}\mathbf{1}_{N,1} \\ \left(y^i\right)^T A_i - \frac{1}{2}\mathbf{1}_{1,N} & Z_{1,N} & 0 & 0 \\ Z_{1,N} & Z_{1,N} & 0 & 0 \end{pmatrix}.$$

By applying Lemma 5.8 to each direction, we see that there must be a maximum flow that satisfies all the edges originating from node 1. As such, there is no path from node 1 to any other node in R. □

We conclude that one can only hope for rood duality based methods to find a useful persistent partial solution if some of the values of y are either relatively large or relatively small.

6. Numerical experiments

In this section we report the results of our numerical experiments meant to test the methods described in the previous sections.

6.1. **Setup.** We implemented our methods in Python, making heavy use of the SciPy library [17]. We used the compressed sparse row format to save the projection matrix A. To compute an ε-pseudo-optimal solution of CCLS, we used the `scipy.optimize.least_squares` method. In particular, we selected the Trust Region Reflective algorithm [6]. This method is well suited for sparse problems with box constraints, such as CCLS. In order to obtain an ε-pseudo-optimal solution, we entered ϵ as the `gtol` stopping condition and disabled the others. We used $\frac{1}{2}\mathbf{1}$ as the initial solution. To compute the relevant properties of the implication network we used methods from `scipy.sparse.csgraph`. The method we used to compute the maximum flow, `scipy.csgraph.maximum_flow`, uses the Edmonds–Karp algorithm [7] to do so. The implementation in Scipy only supports integer capacities. In order to apply the method to noisy projections we truncated the projection data to two decimals and scaled the capacities accordingly.

The presented results were obtained by using an MSI laptop with an Intel Core i5 processor, running the Windows 10 operating system. Our code can be found at `https://github.com/s-fleuren/masterThesis`

We will work with the lattice set reconstruction version of discrete tomography, with the direction sets D^2, D^3 and D^4 as defined in Section 4.5. We generated our projection data as follows: we use a selection of binary images from the MPEG-7 Core Experiment CE-Shape-1 data set[5], see Table 1 for some of their properties. For each (vectorised) image χ, we compute the projection data as $y_i = e_i(A\chi)_i$, where the elements of noise term e are drawn from the normal distribution with mean 1 and $\sigma \in \{0.02, 0.04, 0.08\}$. The error term is chosen like this because in practise the noise often increases with the amount of matter (value 1 pixels) an X-ray passes [2]. For each χ and σ we will test the following methods:

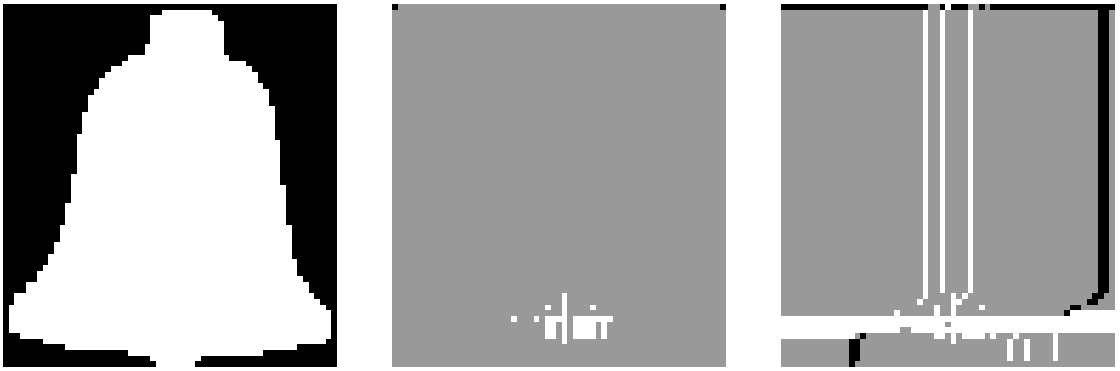
Method a: The size one autarky elimination algorithm as described in Section 3.

Method b: The size one autarky elimination algorithm followed by Algorithm 1 with $\varepsilon = 10^{-6}$ and threshold 0. This gives a method similar to the one introduced in [12]. The differences lie in the way $\hat{\nu}$ is computed and the way it is determined if ν^* can be assumed to be non-zero.

[5]See `http://www.dabi.temple.edu/~shape/MPEG7/dataset.html`, thanks to A. Kadu for pointing out this resource

Image name	n_1	n_2	N	density
bell-2	64	59	3776	0.5874
crown-19	41	72	2952	0.5464
crown-2	43	84	3612	0.5778
crown-4	38	71	2698	0.6197
crown-8	42	80	3360	0.5967
crown-9	41	76	3116	0.6146
hat-5	48	50	2400	0.4746
horseshoe-10	61	59	3599	0.5854
horseshoe-8	61	62	3782	0.4381

Table 1. Properties of the images we used for our experiments. Here the density is given by dividing the number of white (value 1) pixels by the total number of pixels



(A) (B) (C)

FIGURE 7. The partial solutions obtained for the image bell-2 with $\sigma = 0.08$ and $m = 2$. Undetermined pixels are coloured grey. On the left is the original image, in the middle is the partial solution obtained by Method a (and c and d), and on the right the partial solution obtained by Method b.

Method c: The size one autarky elimination algorithm followed by Algorithm 3 with $\varepsilon = 10^{-6}$, where $\widetilde{x}$ is determined by the rounding procedure Algorithm 2.

Method d: The size one autarky elimination algorithm followed by the implication network algorithm described in Section 5.

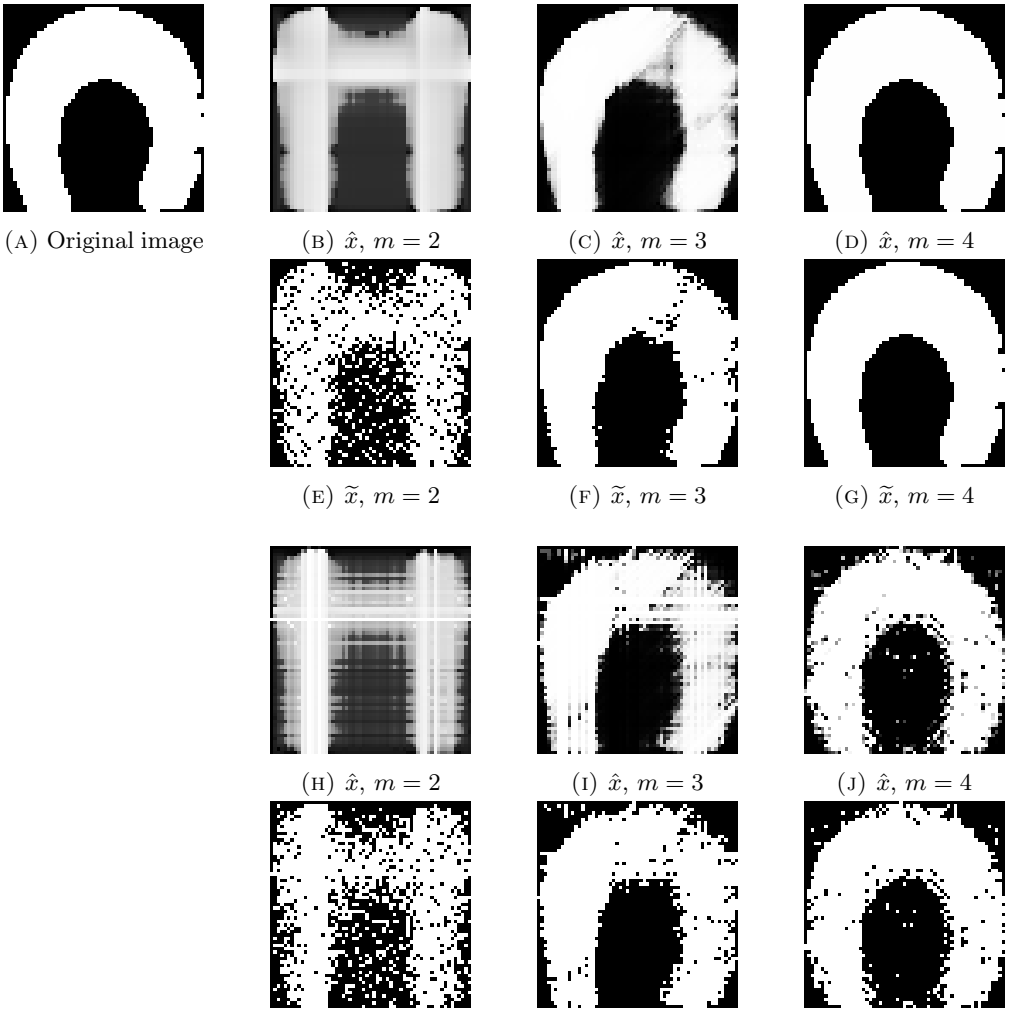
For each method we report the running time, the size of the returned partial solution z, and the quantity $a(z, \chi) := |\{i \in [N] : z_i = \chi_i\}|$, i.e. the number of pixels where z and χ agree. We also report the value of $f_{LS}(\hat{x})$, where $\hat{x}$ is the the output from the least-squares solver used for Method b and Method c, and the values $f_{LS}(\widetilde{x})$ and $a(\widetilde{x}, \chi)$, where $\widetilde{x}$ is the output of Algorithm 2 used for Method c. Our full results can be found in the tables in Appendix A. In the next section we will summarise them and give some illustrations.

6.2. **Results.** Method a was able to determine a small number of pixels for most of the images (usually 15 or less, with the exceptions being bell-2 and crown-9 with $m = 2$ and $\sigma = 0.08$, where method a was able to determine respectively 46 and 47 pixels). As expected, the number of pixels that can be determined by Method a decreases when we add more directions: when $m = 4$ there was only one case (bell-2, $\sigma = 0.08$) where the method was able to determine a pixel. For some images, especially when $m = 2$, Method a is able to determine more pixels when the noise level increases. In all cases, even in the presence of noise, the pixels determined by Method a all agree with the original image χ. The run time of Method a is less than a decisecond for all test cases, which is negligible compared to the run times of the other three methods.

Method b was able to determine more pixels than Method a for most of the cases with $\sigma = 0.08$, topping at 579 determined pixels for image bell-2 with $m = 2$ and $\sigma = 0.08$. However, some of the pixels determined by Method b do not agree with the original image χ. For instance for the bell-2, $m = 2$, $\sigma = 0.08$ case, of the 579 determined pixels 548 agreed with χ. We should note that we do not know the optimal solution(s) $\bar{x}$ of BCLS, and we do not know how many of the pixels determined by Method b agree with $\bar{x}$ (we do know this for the other three methods, since their output is guaranteed to be persistent). For cases with $\sigma < 0.08$, Method b determined more pixels than Method a a single time, being hat-5 with $m = 2$ and $\sigma = 0.04$. The run time of Method b varies quite a lot between cases, even when the original image is the same. For

instance, for image horseshoe-8 the method takes less than 2 deciseconds when $m = 2$ but more than a minute when $m = 4$ and $\sigma = 0.08$.

Method c and d were not able to determine more pixels than method a in any of the cases. The run time of method c is generally only a few deciseconds slower than method b, which indicates that the run time of method c is dominated by computing $\hat{x}$, and that the rounding procedure is relatively cheap. The run time of method d mostly depends on the size of the image and the number of projections, which makes sense because these two factors determine the number of nodes and arcs of the implication network.



(A) Original image (B) $\hat{x}$, $m = 2$ (C) $\hat{x}$, $m = 3$ (D) $\hat{x}$, $m = 4$

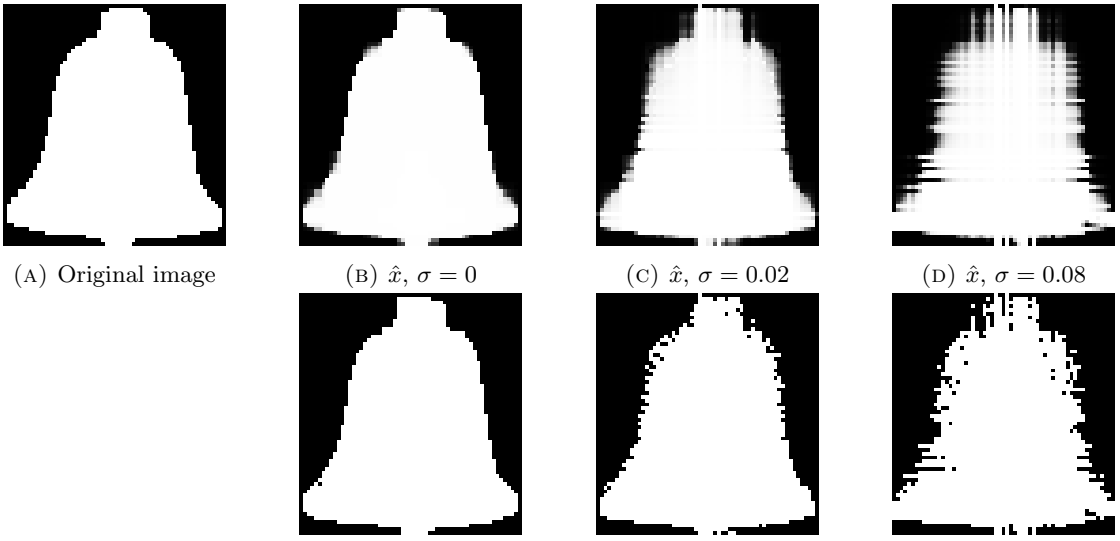
(E) $\widetilde{x}$, $m = 2$ (F) $\widetilde{x}$, $m = 3$ (G) $\widetilde{x}$, $m = 4$

(H) $\hat{x}$, $m = 2$ (I) $\hat{x}$, $m = 3$ (J) $\hat{x}$, $m = 4$

(K) $\widetilde{x}$, $m = 2$ (L) $\widetilde{x}$, $m = 3$ (M) $\widetilde{x}$, $m = 4$

FIGURE 8. The constructed $\hat{x}$ and $\widetilde{x}$ for the image Horseshoe-10. The value of σ is 0 for images (B)-(G) and $\sigma = 0.08$ for the images (H)-(M)

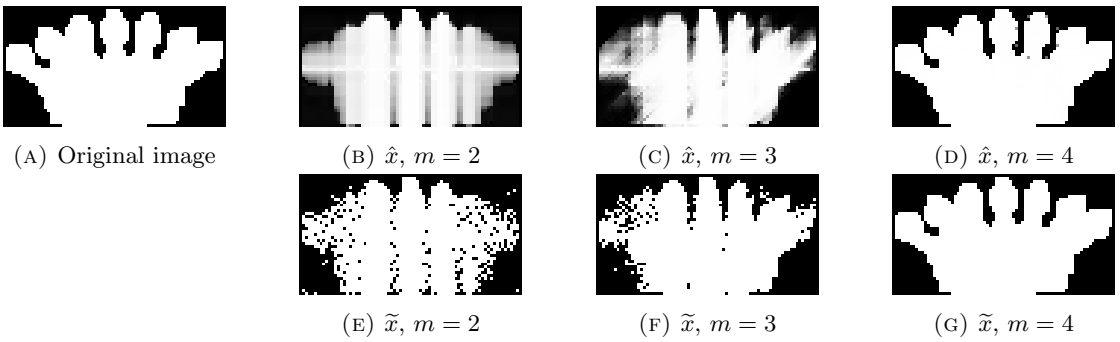
For some images, the $\widetilde{x}$ computed during the execution of method c gives a good approximation of the optimal solution. See for instance Figure 9: with just the horizontal and vertical projections the original image of bell-2 is reconstructed almost completely in the noiseless case (the value of



(A) Original image (B) $\hat{x}$, $\sigma = 0$ (C) $\hat{x}$, $\sigma = 0.02$ (D) $\hat{x}$, $\sigma = 0.08$

(E) $\widetilde{x}$, $\sigma = 0$ (F) $\widetilde{x}$, $\sigma = 0.02$ (G) $\widetilde{x}$, $\sigma = 0.08$

FIGURE 9. The constructed $\hat{x}$ and $\widetilde{x}$ for the image bell-2 with $m = 2$



(A) Original image (B) $\hat{x}$, $m = 2$ (C) $\hat{x}$, $m = 3$ (D) $\hat{x}$, $m = 4$

(E) $\widetilde{x}$, $m = 2$ (F) $\widetilde{x}$, $m = 3$ (G) $\widetilde{x}$, $m = 4$

FIGURE 10. The constructed $\hat{x}$ and $\widetilde{x}$ for the image crown-9 with $\sigma = 0$.

$f_{LS}(\widetilde{x})$ is 5). Even when the noise levels get higher the reconstruction seems still decently faithful to the original image. However, for some images more directions are needed. For instance, with just two directions the shape of horseshoe-10 is not captured quite right by $\hat{x}$ and $\widetilde{x}$, see Figure 8. The same goes for the image crown-9, see Figure 10. However, when $m = 4$ both these images are reconstructed well. In fact, for all tested images the original image is completely reconstructed by $\widetilde{x}$ when $m = 4$ and $\sigma = 0$.

7. Conclusions

We considered a few different methods that can be used as preprocessing techniques for discrete tomography problems. We found that the use of these methods is very limited. Our methods based on the Langrangian dual program can only be expected to identify some pixel values when noise is present, and did poorly in the numerical experiments: Method c was not able to identify any pixel values at all, and while Method b was able to determine some pixels for some images we cannot guarantee that these assignments are correct. Furthermore, we have shown that methods based on roof duality are only able to determine pixel values if there are projections with relatively low or relatively high values. In our numerical experiments, the roof duality based method was not able to determine any pixel values. In conclusion, we do not expect the preprocessing methods to be useful in practise.

However, we also saw in our experiments that computing a close-to-optimal solution of CCLS and then rounding it with Algorithm 2 can give a good approximate solution of BCLS. We expect that this procedure can be further refined (at the cost of additional computation time), by adding intermediate steps in the rounding procedure. In particular, after a portion of the values of $\hat{x}$ have been rounded to binary values, it would be possible to update the remaining values of $\hat{x}$ before continuing with the rounding procedure. For future work we suggest to compare this method with other heuristic methods, like those given in [2] or [16].

References

[1] S. Barnett. *Matrices: methods and applications*. Clarendon Press, 1990.

[2] K. J. Batenburg. A network flow algorithm for reconstructing binary images from discrete X-rays. *Journal of Mathematical Imaging and Vision*, 27(2):175–191, 2007.

[3] E. Boros and P. L. Hammer. Pseudo-Boolean optimization. *Discrete Applied Mathematics*, 123(1-3):155–225, 2002.

[4] E. Boros, P. L. Hammer, and G. Tavares. Preprocessing of Unconstrained Quadratic Binary Optimization. *Rutgers Research Report*, 2006.

[5] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[6] M. A. Branch, T. F. Coleman, and Y. Li. A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM Journal on Scientific Computing*, 21(1):1–23, 1999.

[7] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*, 19(2):248–264, 1972.

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

[9] M. Grant, S. Boyd, and Y. Ye. CVX: Matlab software for disciplined convex programming, 2009.

[10] P. L. Hammer, P. Hansen, and B. Simeone. Roof duality, complementation and persistency in quadratic 0–1 optimization. *Mathematical programming*, 28(2):121–155, 1984.

[11] G. T. Herman and A. Kuba. *Advances in discrete tomography and its applications*. Springer Science & Business Media, 2008.

[12] A. Kadu and T. Van Leeuwen. A convex formulation for binary tomography. *IEEE Transactions on Computational Imaging*, 2019.

[13] V. Kolmogorov. Generalized roof duality and bisubmodular functions. *Discrete Applied Mathematics*, 160(4-5):416–426, 2012.

[14] V. Kolmogorov and C. Rother. Minimizing nonsubmodular functions with graph cuts - A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1274–1279, 2007.

[15] M. Planitz. Inconsistent systems of linear equations. *The Mathematical Gazette*, 63(425):181–185, 1979.

[16] T. Schüle, C. Schnörr, S. Weber, and J. Hornegger. Discrete tomography by convex-concave regularization and D.C. programming. *Discrete Applied Mathematics*, 151(1-3):229–243, 2005.

[17] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, b. Polat, Y. Feng, E. W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. . . Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 2020.

Appendix A. Full results

Image name	m	σ	$\lvert I_z\rvert$	$a(z,\chi)$	$\lvert I_z\rvert$	$a(z,\chi)$	$\lvert I_z\rvert$	$a(z,\chi)$	$\lvert I_z\rvert$	$a(z,\chi)$
bell-2	2	0.02	3	3	3	3	3	3	3	3
bell-2	2	0.04	3	3	3	3	3	3	3	3
bell-2	2	0.08	46	46	579	548	46	46	46	46
bell-2	3	0.02	1	1	1	1	1	1	1	1
bell-2	3	0.04	1	1	1	1	1	1	1	1
bell-2	3	0.08	5	5	217	210	5	5	5	5
bell-2	4	0.02	0	0	0	0	0	0	0	0
bell-2	4	0.04	0	0	0	0	0	0	0	0
bell-2	4	0.08	1	1	133	132	1	1	1	1
crown-19	2	0.02	1	1	1	1	1	1	1	1
crown-19	2	0.04	2	2	2	2	2	2	2	2
crown-19	2	0.08	7	7	73	69	7	7	7	7
crown-19	3	0.02	1	1	1	1	1	1	1	1
crown-19	3	0.04	1	1	1	1	1	1	1	1
crown-19	3	0.08	1	1	47	47	1	1	1	1
crown-19	4	0.02	0	0	0	0	0	0	0	0
crown-19	4	0.04	0	0	0	0	0	0	0	0
crown-19	4	0.08	0	0	72	66	0	0	0	0
crown-2	2	0.02	2	2	2	2	2	2	2	2
crown-2	2	0.04	2	2	2	2	2	2	2	2
crown-2	2	0.08	9	9	49	46	9	9	9	9
crown-2	3	0.02	1	1	1	1	1	1	1	1
crown-2	3	0.04	1	1	1	1	1	1	1	1
crown-2	3	0.08	2	2	128	121	2	2	2	2
crown-2	4	0.02	0	0	0	0	0	0	0	0
crown-2	4	0.04	0	0	0	0	0	0	0	0
crown-2	4	0.08	0	0	54	54	0	0	0	0
crown-4	2	0.02	1	1	1	1	1	1	1	1
crown-4	2	0.04	9	9	9	9	9	9	9	9
crown-4	2	0.08	8	8	176	161	8	8	8	8
crown-4	3	0.02	1	1	1	1	1	1	1	1
crown-4	3	0.04	1	1	1	1	1	1	1	1
crown-4	3	0.08	1	1	46	46	1	1	1	1
crown-4	4	0.02	0	0	0	0	0	0	0	0
crown-4	4	0.04	0	0	7	7	0	0	0	0
crown-4	4	0.08	0	0	91	91	0	0	0	0
crown-8	2	0.02	1	1	1	1	1	1	1	1
crown-8	2	0.04	1	1	1	1	1	1	1	1
crown-8	2	0.08	15	15	314	309	15	15	15	15
crown-8	3	0.02	1	1	1	1	1	1	1	1
crown-8	3	0.04	1	1	1	1	1	1	1	1
crown-8	3	0.08	1	1	47	46	1	1	1	1
crown-8	4	0.02	0	0	0	0	0	0	0	0
crown-8	4	0.04	0	0	0	0	0	0	0	0
crown-8	4	0.08	0	0	49	49	0	0	0	0
crown-9	2	0.02	2	2	2	2	2	2	2	2
crown-9	2	0.04	2	2	2	2	2	2	2	2
crown-9	2	0.08	47	47	379	312	47	47	47	47
crown-9	3	0.02	1	1	1	1	1	1	1	1
crown-9	3	0.04	1	1	1	1	1	1	1	1
crown-9	3	0.08	1	1	12	12	1	1	1	1
crown-9	4	0.02	0	0	0	0	0	0	0	0
crown-9	4	0.04	0	0	0	0	0	0	0	0
crown-9	4	0.08	0	0	76	73	0	0	0	0

Table 2. Partial solution properties, part 1

Image name	m	σ	$\lvert I_z\rvert$	$a(z,\chi)$	$\lvert I_z\rvert$	$a(z,\chi)$	$\lvert I_z\rvert$	$a(z,\chi)$	$\lvert I_z\rvert$	$a(z,\chi)$
hat-5	2	0.02	14	14	14	14	14	14	14	14
hat-5	2	0.04	14	14	63	62	14	14	14	14
hat-5	2	0.08	14	14	144	141	14	14	14	14
hat-5	3	0.02	1	1	1	1	1	1	1	1
hat-5	3	0.04	1	1	1	1	1	1	1	1
hat-5	3	0.08	1	1	101	99	1	1	1	1
hat-5	4	0.02	0	0	0	0	0	0	0	0
hat-5	4	0.04	0	0	0	0	0	0	0	0
hat-5	4	0.08	0	0	3	3	0	0	0	0
horseshoe-10	2	0.02	1	1	1	1	1	1	1	1
horseshoe-10	2	0.04	1	1	1	1	1	1	1	1
horseshoe-10	2	0.08	7	7	60	56	7	7	7	7
horseshoe-10	3	0.02	1	1	1	1	1	1	1	1
horseshoe-10	3	0.04	1	1	1	1	1	1	1	1
horseshoe-10	3	0.08	2	2	2	2	2	2	2	2
horseshoe-10	4	0.02	0	0	0	0	0	0	0	0
horseshoe-10	4	0.04	0	0	0	0	0	0	0	0
horseshoe-10	4	0.08	0	0	0	0	0	0	0	0
horseshoe-8	2	0.02	15	15	15	15	15	15	15	15
horseshoe-8	2	0.04	15	15	15	15	15	15	15	15
horseshoe-8	2	0.08	15	15	15	15	15	15	15	15
horseshoe-8	3	0.02	5	5	5	5	5	5	5	5
horseshoe-8	3	0.04	5	5	5	5	5	5	5	5
horseshoe-8	3	0.08	5	5	5	5	5	5	5	5
horseshoe-8	4	0.02	0	0	0	0	0	0	0	0
horseshoe-8	4	0.04	0	0	0	0	0	0	0	0
horseshoe-8	4	0.08	0	0	0	0	0	0	0	0

TABLE 3. Partial solution properties, part 2

Image name	m	σ	time (s)	time (s)	time (s)	time (s)
bell-2	2	0.02	0.02	1.48	1.67	127.59
bell-2	2	0.04	0.02	0.80	1.00	126.59
bell-2	2	0.08	0.06	5.88	5.69	127.66
bell-2	3	0.02	0.03	17.70	17.94	235.48
bell-2	3	0.04	0.02	3.19	3.44	236.00
bell-2	3	0.08	0.03	15.91	16.13	237.75
bell-2	4	0.02	0.00	14.09	14.53	350.64
bell-2	4	0.04	0.02	25.94	26.47	350.03
bell-2	4	0.08	0.02	21.27	21.67	375.59
crown-19	2	0.02	0.02	0.25	0.39	53.73
crown-19	2	0.04	0.02	0.20	0.36	54.25
crown-19	2	0.08	0.03	0.28	0.39	59.17
crown-19	3	0.02	0.03	3.69	3.88	82.98
crown-19	3	0.04	0.02	7.38	7.47	90.63
crown-19	3	0.08	0.02	2.69	2.83	95.78
crown-19	4	0.02	0.02	24.83	25.28	130.36
crown-19	4	0.04	0.00	56.69	56.77	138.06
crown-19	4	0.08	0.02	101.02	101.67	154.78
crown-2	2	0.02	0.02	1.75	1.98	111.58
crown-2	2	0.04	0.03	1.42	1.58	106.66
crown-2	2	0.08	0.02	1.55	1.70	110.06
crown-2	3	0.02	0.02	5.52	5.78	171.44
crown-2	3	0.04	0.03	50.30	50.09	175.27
crown-2	3	0.08	0.03	8.69	8.78	189.88
crown-2	4	0.02	0.02	11.38	11.77	260.20
crown-2	4	0.04	0.02	8.42	8.81	272.30
crown-2	4	0.08	0.02	31.77	32.19	293.44
crown-4	2	0.02	0.02	1.13	1.27	54.33
crown-4	2	0.04	0.03	0.97	1.09	54.33
crown-4	2	0.08	0.02	0.64	0.64	53.52
crown-4	3	0.02	0.02	17.03	17.22	76.19
crown-4	3	0.04	0.02	8.13	8.34	82.25
crown-4	3	0.08	0.02	13.72	13.75	79.81
crown-4	4	0.02	0.00	5.02	5.14	118.78
crown-4	4	0.04	0.00	6.02	6.20	124.03
crown-4	4	0.08	0.00	19.06	19.13	132.69
crown-8	2	0.02	0.02	0.55	0.73	83.53
crown-8	2	0.04	0.02	0.77	0.98	78.11
crown-8	2	0.08	0.03	1.02	0.97	93.94
crown-8	3	0.02	0.02	3.13	3.34	142.03
crown-8	3	0.04	0.02	3.30	3.53	147.48
crown-8	3	0.08	0.02	3.31	3.48	155.57
crown-8	4	0.02	0.00	65.86	66.00	235.78
crown-8	4	0.04	0.02	7.47	7.70	228.95
crown-8	4	0.08	0.02	33.39	33.72	265.10
crown-9	2	0.02	0.02	0.73	0.91	68.55
crown-9	2	0.04	0.02	0.42	0.60	68.13
crown-9	2	0.08	0.06	2.08	2.00	74.66
crown-9	3	0.02	0.02	2.34	2.56	106.41
crown-9	3	0.04	0.03	3.77	3.95	105.88
crown-9	3	0.08	0.03	5.78	6.00	118.19
crown-9	4	0.02	0.00	136.02	134.17	162.86
crown-9	4	0.04	0.00	115.16	115.30	172.81
crown-9	4	0.08	0.02	32.05	32.43	199.28

TABLE 4. Method run times, part 1

Image name	m	σ	time (s)	time (s)	time (s)	time (s)
hat-5	2	0.02	0.03	0.17	0.30	22.47
hat-5	2	0.04	0.02	0.55	0.63	22.39
hat-5	2	0.08	0.03	0.53	0.55	24.17
hat-5	3	0.02	0.02	8.30	8.45	42.73
hat-5	3	0.04	0.02	16.34	16.42	43.31
hat-5	3	0.08	0.02	22.67	22.64	44.14
hat-5	4	0.02	0.02	5.48	5.70	53.06
hat-5	4	0.04	0.00	10.77	11.02	54.36
hat-5	4	0.08	0.02	3.17	3.41	53.67
horseshoe-10	2	0.02	0.02	0.14	0.36	85.52
horseshoe-10	2	0.04	0.02	0.16	0.36	84.88
horseshoe-10	2	0.08	0.03	0.33	0.48	85.14
horseshoe-10	3	0.02	0.02	4.78	5.06	156.78
horseshoe-10	3	0.04	0.03	8.27	8.42	158.69
horseshoe-10	3	0.08	0.03	6.47	6.67	166.81
horseshoe-10	4	0.02	0.00	16.11	16.14	243.67
horseshoe-10	4	0.04	0.02	28.08	28.61	244.00
horseshoe-10	4	0.08	0.02	13.31	13.72	255.13
horseshoe-8	2	0.02	0.03	0.14	0.38	76.22
horseshoe-8	2	0.04	0.03	0.14	0.39	76.61
horseshoe-8	2	0.08	0.05	0.14	0.38	78.95
horseshoe-8	3	0.02	0.03	5.89	6.19	118.03
horseshoe-8	3	0.04	0.02	5.27	5.48	118.36
horseshoe-8	3	0.08	0.03	3.17	3.41	117.88
horseshoe-8	4	0.02	0.02	51.98	52.95	178.36
horseshoe-8	4	0.04	0.02	17.92	18.64	180.52
horseshoe-8	4	0.08	0.02	79.06	80.13	180.23

TABLE 5. Method run times, part 2

Image name	m	σ	$f_{LS}(\hat{x})$	$f_{LS}(\tilde{x})$	$a(\tilde{x},\chi)$
bell-2	2	0.00	0.00	5.00	3747
bell-2	2	0.02	1.59	14.61	3675
bell-2	2	0.04	4.87	17.84	3671
bell-2	2	0.08	117.54	134.56	3559
bell-2	3	0.00	0.00	0.00	3776
bell-2	3	0.02	3.52	19.06	3692
bell-2	3	0.04	7.42	27.96	3602
bell-2	3	0.08	122.31	140.56	3502
bell-2	4	0.00	0.00	0.00	3776
bell-2	4	0.02	6.59	34.00	3701
bell-2	4	0.04	16.28	49.70	3613
bell-2	4	0.08	195.23	225.53	3494
crown-19	2	0.00	0.00	12.00	2428
crown-19	2	0.02	0.60	12.72	2440
crown-19	2	0.04	0.99	11.96	2420
crown-19	2	0.08	4.13	16.15	2392
crown-19	3	0.00	0.00	23.00	2752
crown-19	3	0.02	0.65	29.52	2758
crown-19	3	0.04	3.13	25.09	2729
crown-19	3	0.08	27.76	57.92	2603
crown-19	4	0.00	0.00	0.00	2952
crown-19	4	0.02	1.72	26.96	2897
crown-19	4	0.04	12.94	41.13	2816
crown-19	4	0.08	65.28	97.30	2698
crown-2	2	0.00	0.00	13.00	3474
crown-2	2	0.02	1.00	18.17	3465
crown-2	2	0.04	2.85	17.30	3398
crown-2	2	0.08	20.15	33.68	3302
crown-2	3	0.00	0.00	0.00	3612
crown-2	3	0.02	3.43	22.97	3543
crown-2	3	0.04	7.67	22.53	3452
crown-2	3	0.08	61.16	81.94	3320
crown-2	4	0.00	0.00	0.00	3612
crown-2	4	0.02	4.59	35.09	3554
crown-2	4	0.04	22.59	54.26	3465
crown-2	4	0.08	82.31	113.23	3359
crown-4	2	0.00	0.00	11.00	2572
crown-4	2	0.02	0.03	11.51	2550
crown-4	2	0.04	4.79	14.09	2520
crown-4	2	0.08	25.92	37.58	2441
crown-4	3	0.00	0.00	0.00	2698
crown-4	3	0.02	3.04	15.44	2631
crown-4	3	0.04	8.88	23.00	2584
crown-4	3	0.08	37.95	52.20	2516
crown-4	4	0.00	0.00	0.00	2698
crown-4	4	0.02	6.12	24.86	2673
crown-4	4	0.04	35.99	65.44	2596
crown-4	4	0.08	106.41	133.20	2511
crown-8	2	0.00	0.00	13.00	2916
crown-8	2	0.02	0.92	15.23	2902
crown-8	2	0.04	3.35	14.84	2823
crown-8	2	0.08	19.59	31.38	2822
crown-8	3	0.00	0.00	31.00	3106
crown-8	3	0.02	1.32	34.24	3085
crown-8	3	0.04	2.08	34.09	3080
crown-8	3	0.08	42.70	74.86	3029
crown-8	4	0.00	0.00	0.00	3360
crown-8	4	0.02	3.32	31.04	3262
crown-8	4	0.04	7.97	42.35	3210
crown-8	4	0.08	148.39	178.15	3039
crown-9	2	0.00	0.00	13.00	2696
crown-9	2	0.02	0.54	14.10	2672
crown-9	2	0.04	6.75	22.23	2679
crown-9	2	0.08	120.92	132.20	2600
crown-9	3	0.00	0.00	27.50	2877
crown-9	3	0.02	2.07	33.25	2867
crown-9	3	0.04	9.64	38.99	2821
crown-9	3	0.08	40.97	72.10	2720
crown-9	4	0.00	0.00	0.00	3116
crown-9	4	0.02	1.45	27.39	3001
crown-9	4	0.04	9.09	41.94	2897
crown-9	4	0.08	150.38	180.65	2813

TABLE 6. Properties of constructed $\hat{x}$ and $\tilde{x}$, part 1

Image name	m	σ	$f_{LS}(\hat{x})$	$f_{LS}(\tilde{x})$	$a(\tilde{x}, \chi)$
hat-5	2	0.00	0.00	8.00	2260
hat-5	2	0.02	0.37	7.79	2252
hat-5	2	0.04	2.66	13.79	2235
hat-5	2	0.08	14.41	25.57	2223
hat-5	3	0.00	0.00	6.50	2379
hat-5	3	0.02	0.77	13.71	2355
hat-5	3	0.04	5.49	16.41	2310
hat-5	3	0.08	80.93	93.13	2260
hat-5	4	0.00	0.00	0.00	2400
hat-5	4	0.02	2.99	21.33	2374
hat-5	4	0.04	11.35	33.61	2317
hat-5	4	0.08	43.91	66.96	2280
horseshoe-10	2	0.00	0.00	10.00	2605
horseshoe-10	2	0.02	0.27	10.52	2631
horseshoe-10	2	0.04	0.85	12.10	2605
horseshoe-10	2	0.08	5.52	17.87	2573
horseshoe-10	3	0.00	0.00	29.00	3397
horseshoe-10	3	0.02	0.40	30.43	3376
horseshoe-10	3	0.04	5.07	31.26	3351
horseshoe-10	3	0.08	16.92	44.00	3248
horseshoe-10	4	0.00	0.00	0.00	3599
horseshoe-10	4	0.02	3.71	32.99	3513
horseshoe-10	4	0.04	14.32	44.36	3444
horseshoe-10	4	0.08	30.67	60.34	3338
horseshoe-8	2	0.00	0.00	12.00	2762
horseshoe-8	2	0.02	0.06	11.65	2780
horseshoe-8	2	0.04	1.50	11.13	2776
horseshoe-8	2	0.08	4.13	13.12	2776
horseshoe-8	3	0.00	0.00	30.50	3447
horseshoe-8	3	0.02	0.22	30.19	3446
horseshoe-8	3	0.04	0.54	30.41	3402
horseshoe-8	3	0.08	4.72	37.30	3295
horseshoe-8	4	0.00	0.00	0.00	3782
horseshoe-8	4	0.02	3.08	26.85	3758
horseshoe-8	4	0.04	8.08	39.79	3678
horseshoe-8	4	0.08	68.91	103.11	3559

TABLE 7. Properties of constructed $\hat{x}$ and $\tilde{x}$, part 2