DOI: 10.2478/auom-2018-0021
An. Şt. Univ. Ovidius Constanţa

sciendo
Vol. 26(2),2018, 105–133

DC-Programming versus ℓ_0-Superiorization for Discrete Tomography

Aviv Gibali and Stefania Petra

Abstract

In this paper we focus on the reconstruction of sparse solutions to *underdetermined* systems of linear equations with variable bounds. The problem is motivated by sparse and gradient-sparse reconstruction in binary and discrete tomography from limited data. To address the ℓ_0-minimization problem we consider two approaches: DC-programming and ℓ_0-superiorization. We show that ℓ_0-minimization over bounded polyhedra can be equivalently formulated as a DC program. Unfortunately, standard DC algorithms based on convex programming often get trapped in local minima. On the other hand, ℓ_0-superiorization yields comparable results at significantly lower costs.

1 Introduction

We consider the ℓ_0-minimization problem

$$\min \|Dx\|_0 \qquad \text{subject to} \quad x \in \mathcal{P} \subset \mathbb{R}^n, \tag{1.1}$$

over a polyhedron $\mathcal{P}$. Matrix $D \in \{0,1\}^{n\times n}$ is a diagonal matrix that selects entries in x, $\|x\|_0 := |\operatorname{supp}(x)| := |\{i \in \{1,2,\dots,n\} \colon x_i \neq 0\}|$ counts the nonzeros entries and is typically coined ℓ_0-*norm*. $\|\cdot\|_0$ is not a norm and problem (1.1) is known to be NP-hard. See e.g. [Nat95] for the case of affine linear constraints

Key Words: DC-Programming, Sparse Regularization, Superiorization, Discrete Tomography, Split Convex Feasibility Problem
2010 Mathematics Subject Classification: Primary 90C26, 65K05; Secondary 90C90, 47J25, 49M30.
Received: May, 2017.
Revised: June, 2017.
Accepted: October, 2017.

$\mathcal{P} = \{x \in \mathbb{R}^n \colon Ax = b\}$, with $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$. In this work we focus of polyhedra of the form

$$\mathcal{P} = \{x \in \mathbb{R}^n \colon Ax = b, Cx \geq d\}, \tag{1.2}$$

described by linear equalities and inequalities, with $A \in \mathbb{R}^{m\times n}$, $C \in \mathbb{R}^{k\times n}$, $b \in \mathbb{R}^m$, $d \in \mathbb{R}^k$.

1.1 Motivation

Our work is motivated by the image reconstruction problem from few linear measurements, e.g. discrete tomography [HK99], where typically there are known bounds on the image values. Moreover we aim to reconstruct an image that exhibits some structure, e.g. sparsity/gradient-sparsity.

Recovering a structured signal/image from few linear measurements is a central point in both compressed sensing (CS) [FR13] and discrete tomography [HK99]. In CS the signal structure is described by means of a low complexity model. The most basic model is signal sparsity and the associated ℓ_0-minimization problem writes

$$\min \|x\|_0 \qquad \text{subject to} \quad Ax = b. \tag{1.3}$$

If x is sparse, i.e. $x \in \Sigma_s := \{x \colon \|x\|_0 \leq s\}$ and the *measurement* matrix $A \in \mathbb{R}^{m\times n}$ satisfies certain conditions, like e.g. being well-conditioned when restricted to Σ_s, see [FR13], one can consider instead of (1.3) the easier ℓ_1-minimization problem

$$\min \|x\|_1 \qquad \text{subject to} \quad Ax = b. \tag{1.4}$$

The theory of CS implies that if $m \geq C \cdot s \log(n/s)$ and the entries of A follow a normal distribution. i.e. $a_{ij} \sim \mathcal{N}(0,1)$, the problem (1.3) can be solved in polynomial time via (1.4). If one could solve (1.3) directly, then $m \geq 2s$ measurements would suffice to recover any s-sparse solution. Interestingly, if one approximates the ℓ_0-norm by an ℓ_p-quasi-norm, where $0 < p < 1$, then one can recover a s-sparse vector via ℓ_p-minimization

$$\min \|x\|_p \qquad \text{subject to} \quad Ax = b, \tag{1.5}$$

as soon as $m \geq C_1(p) \cdot s + C_2(p) \cdot p \cdot s \cdot \log(n/s)$, see [CS08]. Hence, although non-convex metrics are generally more challenging to minimize, they have the advantage to enable reconstructing a sparse signal from substantially fewer measurements than the convex ℓ_1-minimzation counterpart. This has been (empirically) observed [YLHX15] also for other ℓ_0-norm substitutes like e.g.

$$\min \|x\|_1 - \|x\|_2 \qquad \text{subject to} \quad Ax = b. \tag{1.6}$$

1.2 Objectives

Solution approaches for (1.3) are typically approximated by related concave programs like (1.5) or (1.6), which are usually solved by DC-programming [THPD05], see [Man96, YLHX15]. In DC-programming, see Sect. 3, one solves a sequence of easier (convex) problems of the same type. Similarly, in a superiorization [Cen15, CZ13] framework, see Sect. 4, one successively applies a basic algorithm, e.g. feasibility seeking, that is *bounded perturbation resilient* to find a "better" feasible solution in terms of a cost function. Our objective is to apply both frameworks to problems (1.1), (1.3), or a related problem like e.g. (1.5) for some $p \in (0, 1)$. In particular are interested in:

- How and why the two approaches differ?
- If superiorization is applicable to sparse recovery and if yes, how?

2 Preliminaries

2.1 Basic Definitions and Notation

The extended real line is denoted by $\overline{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}$. For $n \in \mathbb{N}$, we use the shorthands $[n] = \{1, 2, \ldots, n\}$ and $[n]_0 = \{0, 1, 2, \ldots, n\}$. Vectors $x \in \mathbb{R}^n$ are column vectors and indexed by superscripts. $x^\top$ denotes the transposed vector x and $\langle x, y\rangle$ or $x^\top y$ the Euclidean inner product. $\top$ stands for the transpose. To save space, however, we will sometimes simply write e.g. $z = (x, y)$ instead of correctly denoting $z = \big((x)^\top, (y)^\top\big)^\top$, for $z = \left(\begin{smallmatrix} x \\ y \end{smallmatrix}\right)$. $\mathbb{1} = (1, 1, \ldots, 1)^\top$ denotes the all one-vector whose dimension will always be clear from the context. The dimension of a vector x we denote by $\dim(x)$. The (quasi)-norms $\|x\|_p = \Big(\sum_{i\in[n]} |x_i|^p\Big)^{\frac{1}{p}}$, $p > 0$ will be called ℓ_p-(quasi-)norms. $\|x\|_\infty := \max_{i\in[n]} |x_i|$ is the maximum norm. The ℓ_0-measure (not a norm!) stands for the cardinality of the support of x, i.e. $\|x\|_0 := |\operatorname{supp}(x)|$, with $\operatorname{supp}(x) = \{i \in [n] \colon x_i \neq 0\}$. The sign vector $\operatorname{sign}(x) \in \mathbb{R}^n$ is defined component-wise as

$$(\operatorname{sign}(x))_i = \begin{cases} 1, & x_i > 0, \\ 0, & x_i = 0, \\ -1, & x_i < 0. \end{cases} \tag{2.1}$$

Note that $\|x\|_0 = (\sum_{i\in[n]} \operatorname{sign}(|x_i|)$. The Euclidean unit ball is denoted by $\mathbb{B}(0)$, while we write $\mathbb{B}_\infty(0) := \{x \colon \|x\|_\infty \leq 1\}$ for the unit ball w.r.t. the max-norm.

For some matrix $A \in \mathbb{R}^{m\times n}$, the nullspace is denoted by $\mathcal{N}(A)$ and its range by $\mathcal{R}(A)$.

We consider images $u(x),\ x \in \Omega \subset \mathbb{R}^2$ discretized as follows. Ω is assumed to be a rectangle covered by a *regular grid graph* $G = (V, E)$ of size $|V| = n$. Accordingly, we identify $V = \prod_{i\in[2]}[n_i]_0 \subset \mathbb{Z}^2$, $n_i \in \mathbb{N}$. Thus, vertices $v \in V$ are indexed by $(i,j)^\top \in \mathbb{Z}^2$ with ranges $i \in [n_1]_0, j \in [n_2]_0$, and

$$n = n_1 n_2. \tag{2.2}$$

As a result, discretization of $u(x),\ x \in \Omega$, yields the vector $u \in \mathbb{R}^n$, where we keep the symbol u for simplicity.

Consider the one-dimensional discrete derivative operator

$$\partial\colon \mathbb{R}^m \to \mathbb{R}^{m-1}, \qquad \partial_{i,j} = \begin{cases} -1, & i = j, \\ +1, & j = i+1, \\ 0, & \text{otherwise.} \end{cases} \tag{2.3}$$

Forming corresponding operators ∂_1, ∂_2 for each coordinate, conforming to the ranges of i, j such that $(i,j) \in V$, we obtain the discrete gradient operator

$$\nabla = \begin{pmatrix} \partial_1 \otimes I_2 \\ I_1 \otimes \partial_2 \end{pmatrix} \in \mathbb{R}^{p\times n}, \tag{2.4}$$

where $\otimes$ denotes the Kronecker product and $I_i,\ i = 1, 2$, are identity matrices with appropriate dimensions. The *anisotropic* discretized total variation (TV) is given by

$$\mathrm{TV}(u) := \|\nabla u\|_1. \tag{2.5}$$

The image gradient sparsity is given by $\|\nabla u\|_0$.

For simplicity we will consider as basic *feasible set* the *polyhedron* $\mathcal{P} = \{x \in \mathbb{R}^n\colon Ax = b, Cx \geq d\}$, with $A \in \mathbb{R}^{m\times n}$, $C \in \mathbb{R}^{k\times n}$, $b \in \mathbb{R}^m$, $d \in \mathbb{R}^k$ assumed to be non-empty. This is a particular closed and convex set, which for the applications we have in mind can also be assumed to be bounded.

We denote the class of *proper convex and lower semicontinuous* (lsc) functions by

$$\mathcal{F}_0(\mathbb{R}^n) := \{f\colon \mathbb{R}^n \to \overline{\mathbb{R}}\colon f \text{ is proper, convex and lsc }\}. \tag{2.6}$$

Hence, the *indicator function* $\delta_{\mathcal{P}} : \mathbb{R}^n \to \overline{\mathbb{R}}$ of $\mathcal{P}$ defined as

$$\delta_{\mathcal{P}}(x) = \begin{cases} \infty, & x \in \mathcal{P}, \\ 0, & x \notin \mathcal{P}, \end{cases}$$

is lsc since $\mathcal{P}$ is closed. The *epigraph* of $f\colon \mathbb{R}^n \to \mathbb{R}$ is the set of points lying on or above its graph, that is

$$\mathrm{epi}(f) := \{(x,\alpha) \in \mathbb{R}^n \times \mathbb{R}\colon f(x) \leq \alpha\} \subseteq \mathbb{R}^{n+1}. \tag{2.7}$$

For a function $f\colon \mathbb{R}^n \to \overline{\mathbb{R}}$ the function $f^*\colon \mathbb{R}^n \to \overline{\mathbb{R}}$ defined by

$$f^*(p) := \sup_{x\in\mathbb{R}^n} \{\langle p, x\rangle - f(x)\} \tag{2.8}$$

is called the *(Fenchel) conjugate* of f. The conjugate f^* is the pointwise supremum of the affine functions $p \mapsto \langle p, x\rangle - \alpha$ parametrized by $(x, \alpha) \in \operatorname{epi}(f)$, which implies that f^* is lsc convex.

A mapping $F : \mathbb{R}^n \to \mathbb{R}^n$ is called *L-Lipschitz continuous* for some $L \geq 0$ if

$$\|F(x) - F(y)\| \leq L\|x - y\|, \text{ for all } x, y \in \mathbb{R}^n. \tag{2.9}$$

The *unit sphere* in $\mathbb{R}^n$ is defined as

$$S^{n-1} = \{g \in \mathbb{R}^n \mid \|g\| = 1\}. \tag{2.10}$$

The *set of vertices of the unit hypercube* in $\mathbb{R}^n$ is denoted by $G = \{-1, 1\}^n$.

For any $g \in S^{n-1}$ we define $|g_i| = \max\{|g_k| : k \in [n]\}$. The *set of univariate positive infinitesimal functions* is

$$P = \left\{z(\lambda) \in \mathbb{R}\colon z(\lambda) > 0, \lambda > 0, \lambda^{-1}z(\lambda) \to_{\lambda\to 0} 0\right\}. \tag{2.11}$$

Given a set of points $S = \{x^i\}_{i=1}^N \subset \mathbb{R}^n$, the *convex hull* of S, denoted by $\operatorname{conv}(S)$ is the set of all convex combinations of its points, i.e.

$$\operatorname{conv}(S) = \left\{ \sum_{i\in[N]} \alpha_i x^i \ :\ \alpha_i \geq 0,\ \sum_{i\in[N]} \alpha_i = 1,\ x^i \in S \right\}. \tag{2.12}$$

Let us recall the *orthogonal (metric) projection* operator $P_C : \mathbb{R}^n \to C$ onto a non-empty, closed and convex set $C \subseteq \mathbb{R}^n$. For each point $x \in \mathbb{R}^n$, there exists a unique nearest point, denoted by $P_C(x)$. That is,

$$\|x - P_C(x)\| \leq \|x - y\| \text{ for all } y \in C. \tag{2.13}$$

The metric projection P_C is characterized [GR84, Section 3] by the following two properties:

$$P_C(x) \in C \tag{2.14}$$

and

$$\langle x - P_C(x), P_C(x) - y\rangle \geq 0 \text{ for all } x \in \mathbb{R}^n,\ y \in C, \tag{2.15}$$

and if C is a hyperplane, then (2.15) becomes an equality.

The *subdifferential set* denoted by $\partial f(x)$ of $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ at a point $x \in \mathbb{R}^n$ where $f(x) \in \mathbb{R}$, is defined as

$$\partial f(x) = \{\xi \in \mathbb{R}^n\colon f(z) \geq f(x) + \langle \xi, z - x\rangle, \forall z \in \mathbb{R}^n\}. \tag{2.16}$$

We agree that $\partial f(x) = \emptyset$ if $f(x)$ is not finite. Elements of the subdifferential set are called subgradients.

2.2 Equivalence of ℓ_0- and ℓ_p-Minimization over Bounded Polyhedra

We consider here a variation of problem (1.1)

$$\min \|Dx\|_0 \qquad \text{subject to} \quad x \in \mathcal{P} \cap \mathbb{B}_\infty(0) \tag{2.17}$$

along with the associated ℓ_p-minimization problem

$$\min \|Dx\|_p \qquad \text{subject to} \quad x \in \mathcal{P} \cap \mathbb{B}_\infty(0). \tag{2.18}$$

Since we intersect $\mathcal{P} = \{x \in \mathbb{R}^n \colon Ax = b, Cx \geq d\}$ with the $\mathbb{B}_\infty(0)$ ball our feasible set becomes a *bounded* polyhedron, hence a polytope.

Along the lines of [FM11] we establish the equivalence of (2.17) and (2.18) for a *sufficiently small* $p > 0$. As in [FM11] we rewrite the objective of (2.17) as a step function and reformulate (2.18) as a concave optimization problem. Using that $\|Dx\|_0 = \sum_{i\in[n]} D_{ii} \operatorname{sign}(|x_i|)$ and introducing an auxilary variable y_i for $|x_i|$ we rewrite (2.17) as

$$\min(D\mathbb{1})^\top \operatorname{sign}(y) \qquad \text{subject to} \quad x \in \mathcal{P}, -y \leq x \leq y, -\mathbb{1} \leq x \leq \mathbb{1}. \tag{2.19}$$

Similarly we can rewrite (2.18) as

$$\min(D\mathbb{1})^\top y^p \qquad \text{subject to} \quad x \in \mathcal{P}, -y \leq x \leq y, -\mathbb{1} \leq x \leq \mathbb{1}, \tag{2.20}$$

where the p-power should be understood component-wise, i.e. $y^p := (y_1^p, \dots, y_n^p) \in \mathbb{R}^n$. We define now a subset of the bounded feasible region of the two above problems as follows

$$\mathcal{F} := \{(x, y) \in \mathbb{R}^{2n} \colon x \in \mathcal{P}, -y \leq x \leq y, -\mathbb{1} \leq x \leq \mathbb{1}, 0 \leq y \leq \mathbb{1}\}. \tag{2.21}$$

We can consider further w.l.o.g. $\mathcal{F}$ as the feasible set of both minimization problems (2.19) and (2.20), since in view of the inequalities $-y \leq x \leq y, \|x\|_\infty \leq 1$ the inequalities $0 \leq t \leq \mathbb{1}$ do not influence the solutions x.

Observe now that problem

$$\min(D\mathbb{1})^\top y^p \qquad \text{subject to} \quad (x, y) \in \mathcal{F}$$

is a concave problem with a polytope as feasible set. Hence it's infimum is attained at a vertex of $\mathcal{F}$.

Lemma 2.1. *[Roc70, Corollary 32.3.3 and 32.3.4] Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be a concave function, and let $C \subset \mathbb{R}^n$ be a nonempty polyhedral convex set contained in* $\operatorname{dom} f := \{x \colon f(x) < \infty\}$*. Suppose that C contains no lines, and that f is bounded below on C. Then the infimum of f relative to C is attained at one of the (finitely many) extreme points of C.*

Theorem 2.2. *The ℓ_0-norm minimization problem* (2.17) *is equivalent to the ℓ_p-norm minimization problem* (2.18) *for some $p_0 \leq 1$. Furthermore, there exists a vertex of $\mathcal{F}$ that is an exact solution of the ℓ_0-norm minimization problem* (2.17)*, or equivalently of* (2.19)*, and is a global solution of the ℓ_p-norm minimization problem* (2.18)*, or equivalently of the concave problem* (2.20)*, for some $p_0 = 1/q_0$ where $q_0 \in \mathbb{N}$.*

Proof. Note that the objective function of (2.20) is concave for $y \geq 0$, $p =: 1/q < 1$, and that

$$0 \leq (D\mathbb{1})^\top y^p = (D\mathbb{1})^\top y^{1/q} \leq (D\mathbb{1})^\top \operatorname{sign}(y), \qquad \text{for} \quad 0 \leq y \leq \mathbb{1}. \tag{2.22}$$

Since $(D\mathbb{1})^\top y^{1/q} \geq 0$ holds on the bounded polyhedral set $\mathcal{F}$, it follows by Lem. 2.1 that $y \mapsto (D\mathbb{1})^\top y^{1/q}$ has a vertex $(x(q), y(q))$ of $\mathcal{F}$ as a solution for each $q \in \mathbb{N}$. Since $\mathcal{F}$ has a finite number of vertices, one vertex, say $(\tilde{x}, \tilde{y})$, will repeatedly solve (2.20) for some increasing infinite sequence in a subset $Q := \{q_0, q_1, q_2, \dots\} \subset \mathbb{N}$. Hence for $q_i \in Q$

$$(D\mathbb{1})^\top \tilde{y}^{\frac{1}{q_i}} = \min_{(x,y)\in\mathcal{F}} (D\mathbb{1})^\top y^{\frac{1}{q_i}} \leq \inf_{(x,y)\in\mathcal{F}} (D\mathbb{1})^\top \operatorname{sign}(y), \tag{2.23}$$

where the last inequality above follows from (2.22). Letting $i \to \infty$ it follows from (2.23) that

$$(D\mathbb{1})^\top \operatorname{sign}(\tilde{y}) = \lim_{i\to\infty} (D\mathbb{1})^\top \tilde{y}^{\frac{1}{q_i}} \leq \inf_{(x,y)\in\mathcal{F}} (D\mathbb{1})^\top \operatorname{sign}(y).$$

Since $(\tilde{x}, \tilde{y}) \in \mathcal{F}$, it follows that $(\tilde{x}, \tilde{y})$ solves (2.19). Furthermore $(\tilde{x}, \tilde{y})$ is a vertex of $\mathcal{F}$. □

From the proof of Thm. 2.2, in particular (2.22), we also obtain the following result. This will be further used in order to design a descent direction for the superiorization algorithm in Section 4.

Proposition 2.3. *Let $p > 0$, not necessary $p < 1$ and $0 \leq y \leq \mathbb{1}$. Then*

$$\phi_p(y) := (D\mathbb{1})^\top y^p \leq \|Dy\|_0 =: \phi_0(y). \tag{2.24}$$

Moreover, a nonascent direction for ϕ_p is a nonascent direction for ϕ_0.

Further we note that ϕ_p above is differentiable and a nonascent direction for ϕ_p and ϕ_0 is the gradient of $\phi_p = pDy^{p-1}$.

Remark 2.1. Alternatives to the ℓ_p-quasinorms exists throughout the literature, such as,

$$|x|/(|x| + \alpha), \qquad \text{or} \qquad 1 - \exp(-|x|/\alpha),$$

for some small $\alpha > 0$. These functions are particular instances of $f_\alpha : \mathbb{R} \to \mathbb{R}$, of the form $f_\alpha(x) = r_\alpha(|x|)$, where $r_\alpha(0) = 0$, r_α is increasing and concave on $\mathbb{R}_+$, r'_α and $t \mapsto r_\alpha(t)/t$ is non-increasing. Then $\sum_{i=1}^n f_\alpha(x_i)$ can be used as a proxy for the ℓ_p-quasinorm and for ℓ_0 for an appropriate choice of parameter $\alpha > 0$.

2.3 Sparse Regularization for Binary Tomography

In this section we recast the sparse binary tomography problem as (2.17). We consider the problem of reconstructing a vectorized *binary* image $u \in \{0, 1\}^n$ from a limited number of tomographic projections

$$Mu = q. \tag{2.25}$$

Each pixel is associated with some unknown binary variable $u_i \in \{0, 1\}$. Each entry in $q \in \mathbb{R}^m$, called tomographic measurement or single projection, corresponds to the integrated gray values of u along the single ray, see Fig. 2.1. Hence the integral can be split into the sum of products $M_{ij}u_j$, where each matrix entry $M_{ij} \geq 0$ corresponds to the length of the intersection of the i-th ray with the j-th pixel. If ray i and pixel j do not intersect then $M_{ij} = 0$. Stacking all equations for all the rays together leads to the linear equations in (2.25) and provides the following representation of the reconstruction problem

$$Mu = q, \quad u \in \{0, 1\}^n. \tag{2.26}$$

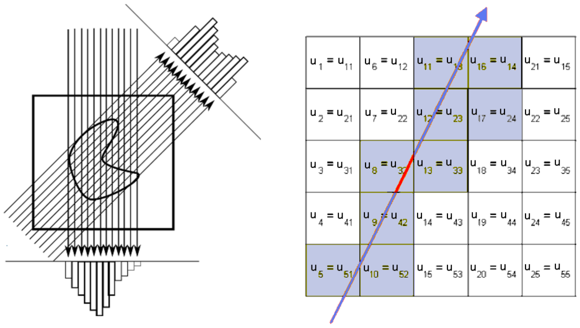


Figure 2.1: Parallelbeam geometry set-up: a set of parallel rays is shot through the object from different directions. These are typically coined as one projection. Two projections are illustrated above. (Left) Illustration of a *single* projection corresponding to a measurement along one ray. The image domain Ω is tiled into pixels or mathematically Haar-basis functions. Hence, a single projection corresponds to the line integral over a piecewise constant function.

The binary reconstruction problem (2.26) is NP-complete for more than two projections. Our approach here is to relax the constraints and to move the combinatorial difficulty into the objective. This leads us to

$$\min \|u\|_0 \qquad \text{subject to} \quad Mu = q, \quad u \in [0, 1]^n. \tag{2.27}$$

Consequently, we obtained a convex feasible set of the form (1.2), with $A := M$, $b := q$, $C := I \in \mathbb{R}^{n \times n}$, $d := 0 \in \mathbb{R}^n$ and m equals the number of rays. Moreover $\|u\|_\infty \leq \mathbb{1}$ and our polyhedron

$$\mathcal{P} = \{u \colon Mu = q, u \in [0,1]^n\}$$

is bounded.

Remark 2.2. Note that problem (2.27) is equivalent to

$$\min \|u\|_p^p \qquad \text{subject to} \quad Mu = q, \quad u \in [0,1]^n$$

for a small but *finite* $p < 1$. In view of the non-negativity of u it is not necessary to introduce auxiliary variables for $|u|$ as done in the previous section.

2.4 Gradient-Sparse Regularization for Discrete Tomography

Our objective in this section is to introduce a second regularization model from discrete tomography and to recast it as (2.17). Like previously, we wish to reconstruct a vectorized image $u \in \mathbb{R}^n$ from a limited number of tomographic projections (2.25). We now have a small range of signal values in u due to few materials only (e.g. bone, fat, water etc.). In particular, we have an upper bound on the maximal absolute value of u,

$$u_i \in [0, u_b], \forall i \in [n].$$

Moreover, we assume that the image to be reconstructed is gradient-sparse, i.e. $\|\nabla u\|_0 \ll n$. This leads us to the following reconstruction problem

$$\min \|\nabla u\|_0 \qquad \text{subject to} \quad Mu = q,\ u \geq 0, \|u\|_\infty \leq u_b. \tag{2.28}$$

Rescaling, with $\tilde{u} := u/u_b$ and $\tilde{M} := u_b M$, leads to

$$\min \|\nabla \tilde{u}\|_0 \qquad \text{subject to} \quad \tilde{M}\tilde{u} = q,\ \tilde{u} \geq 0, \|\tilde{u}\|_\infty \leq 1. \tag{2.29}$$

We further simplify notation by setting $u \leftarrow \tilde{u}$. We note that (2.29) is equivalent to

$$\min_{u,z} \|z\|_0 \qquad \text{subject to} \quad Mu = q,\ \nabla u = z,\ u \in [0,1]^n, \tag{2.30}$$

that can be rewritten as

$$\min_{u,z} \|z\|_0 \qquad \text{subject to} \quad \begin{pmatrix} M & 0 \\ \nabla & -I \end{pmatrix} \begin{pmatrix} u \\ z \end{pmatrix} = \begin{pmatrix} q \\ 0 \end{pmatrix},\ u \in [0,1]^n. \tag{2.31}$$

Using that $\|z\|_\infty = \|\nabla u\|_\infty \leq 1$, in view of the definition of z and the given bounds on u, we can recast (2.31) as

$$\min_x \|Dx\|_0 \qquad \text{subject to} \quad Ax = b, Cx \geq d, \|x\|_\infty \leq 1, \tag{2.32}$$

where

$$A := \begin{pmatrix} M & 0 \\ \nabla & -I \end{pmatrix}, \quad b := \begin{pmatrix} q \\ 0 \end{pmatrix}, \quad x := \begin{pmatrix} u \\ z \end{pmatrix}, \quad D := \begin{pmatrix} 0 & 0 \\ I_{\dim(z)} & 0 \end{pmatrix}$$

and

$$C := \begin{pmatrix} I_{\dim(u)} & 0 \\ 0 & 0 \end{pmatrix}, \quad d := \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

3 DC-Programming

We have seen in Section 2.2 that one can replace ℓ_0-minimization problem over bounded polyhedra with an concave minimization problem. A concave program is a special instance of a *DC-program*. A DC-program is a problem of the form

$$\min_{x \in \mathbb{R}^n} f(x), \qquad f(x) = g(x) - h(x), \tag{3.1}$$

with $g, h \in \mathcal{F}_0(\mathbb{R}^n)$ proper, convex, lsc. We emphasize that problem (3.1) is *non*-convex, since f is decomposed into a convex part g and into a concave part h and in general *non*-convex. Such an f will be called DC function.

As discussed earlier, the conditions for global optimality in DC-programs do not yield efficient general algorithms.

3.1 DC-Algorithm

There are some popular techniques for the global optimization of (3.1) - among them, branch-and-bound and cutting planes algorithms. Here we omit discussion of global optimization techniques for DC-programming, refer to the overview [HT99] and focus instead on an convex-based approach to *local* optimization. The following basic algorithm [PDEB86, PDTH97] computes a non-increasing sequence $\{f(x^{(k)})\}$ converging to a *stationary point* by iteratively minimizing g plus an affine upper bound of h. It is known as the *simplified* DC Algorithm (DCA).

Given $x := x^{(k)}$, the first step of each iteration computes an upper bound of f by determining the closest affine upper bound of the concave component h. The second

Algorithm 1: DC Algorithm (DCA)

```
Input: Good initialization $x$.
Output: Approximate local minimizer $\hat{x}$ of problem (3.1).
begin
    repeat
        $y^{(k)} \in \partial h(x^{(k)})$
        $x^{(k+1)} \in \arg\min_x \left\{g(x) - \big(h(x^{(k)}) + \langle y^{(k)}, x - x^{(k)}\rangle\big)\right\} =$
          $\arg\min_x \left\{g(x) - \langle y^{(k)}, x\rangle\right\}$
    until some convergence criteria is met at $x^{(K)}$.
    $\hat{x} \longleftarrow x^{(K)}$.
```

step updates x by minimizing this upper bound.

$$h(x) + h^*(y) \geq \langle x, y\rangle \quad \Leftrightarrow \quad -h(x) \leq -\langle x, y\rangle + h^*(y), \qquad \forall y \tag{3.2a}$$

$$\longrightarrow \qquad y \in \underset{\tilde{y}}{\operatorname{argmin}}\left\{h^*(\tilde{y}) - \langle x, \tilde{y}\rangle\right\} = \partial h(x) \tag{3.2b}$$

$$\longrightarrow \qquad x \in \underset{\tilde{x}}{\operatorname{argmin}}\Big\{ \underbrace{g(\tilde{x}) - \big(\underbrace{h(x) + \langle y, \tilde{x} - x\rangle}_{\leq h(\tilde{x}),\, \forall \tilde{x}}\big)}_{\geq f(\tilde{x})} \Big\} = \partial g^*(y) \tag{3.2c}$$

Hence, the second step in each iteration is equivalent to choosing $x^{(k+1)} \in \partial g^*(y^{(k)})$. As already mentioned, the sequence of iterates is non-increasing. Denote $x := x^{(k)}$, $y := y^{(k)}$, $x' := x^{(k+1)}$ and

$$\begin{pmatrix} y \\ x' \end{pmatrix} = \begin{pmatrix} \partial h(x) \\ \partial g^*(y) \end{pmatrix}. \tag{3.3}$$

Then

$$
\begin{aligned}
\inf_x f(x) &\leq f(x') && && (3.4a)\\
&= g(x') - h(x') && && (3.4b)\\
&\leq g(x') - \big(h(x) + \langle y, x' - x\rangle\big) && \text{since } y \in \partial h(x) && (3.4c)\\
&= g(x') - \langle y, x'\rangle + \langle y, x\rangle - h(x) && \text{use } x' = \operatorname*{argmin}_{\tilde{x}} g(\tilde{x}) - \langle y, \tilde{x}\rangle && (3.4d)\\
&\leq g(x) - \langle y, x\rangle + \langle y, x\rangle - h(x) && && (3.4e)\\
&= f(x). && && (3.4f)
\end{aligned}
$$

Algorithm 1 generalizes subgradient optimization of convex functions to local optimization of DC functions. Accordingly, basic concepts of convex optimization like duality and KKT conditions were extended to DC functions [Tol78]. [PDTH97, PDTH06] guarantees convergence of the sequence $\{x^{(k)}\}_k$ generated by Algorithm 1 to a critical point by starting with $x^{(0)} \in \operatorname{dom}(g)$.

Remark 3.1. The sequences $\{x^{(k)}\}$ and $\{y^{(k)}\}$ are well defined if and only if

$$\operatorname{dom}\partial g \subset \operatorname{dom}\partial h \quad \text{and} \quad \operatorname{dom}\partial h^* \subset \operatorname{dom}\partial g^*.$$

Theorem 3.1. *[PDTH06, Thm. 3.7] Suppose that the sequences $\{x^{(k)}\}$ and $\{y^{(k)}\}$ are defined by the DCA Algorithm 1 and assume DCA is well defined. Then every limit point of the sequence $x^{(k)}$ (resp. $y^{(k)}$) is a critical point of $g - h$ (resp. $h^* - g^*$).*

3.2 DC-Programming for ℓ_p-Minimization

Recall ℓ_p-minimization, $p < 1$ over bounded polyhedra from (2.18), that was rewritten as a concave optimization problem in (2.20).

The DC Algorithm 1 applied to (2.20), also called *Successive Linearization Algorithm (SLA)*, consists of linearizing the differentiable concave objective function of problem (2.20) around a current point $(x^{(k)}, y^{(k)})$ and solving the resulting linear program. The algorithm terminates in a finite number of steps at a stationary point after adding the constraint $y \geq \epsilon\mathbb{1}$ to the minimization problem (2.20) for some small $\epsilon > 0$ to ensure the differentiability of the objective function of (2.20). Hence, we consider

$$\mathcal{F}_\epsilon := \mathcal{F} \cap \{y\colon y \geq \epsilon\mathbb{1}\},\; \mathcal{F} := \{(x,y) \in \mathbb{R}^{2n}\colon x \in \mathcal{P}, -y \leq x \leq y, -\mathbb{1} \leq x \leq \mathbb{1}, 0 \leq y \leq \mathbb{1}\} \tag{3.5}$$

and

$$\min_{x,y} \mathbb{1}^\top D y^p \qquad \text{subject to} \quad (x,y) \in \mathcal{F}_\epsilon. \tag{3.6}$$

Algorithm 2: DCA for (2.20) / Successive Linearization Algorithm (SLA)

```
Input: Choose $p \in (0,1)$, $\epsilon$ small, $(x^{(0)}, y^{(0)})$.
Output: Approximated stationary point of (2.20).
for $k = 1, \dots,$ do
    $(x^{(k+1)}, y^{(k+1)}) = \arg\min_{(x,y)\in\mathcal{F}_\epsilon} \Big(\big(Dy^{(k)}\big)^{p-1}\Big)^\top y$
    Stop when $\Big(\big(Dy^{(k)}\big)^{p-1}\Big)^\top (y^{(k)} - y^{(k+1)}) = 0$
    Otherwise set $k \leftarrow k+1$
```

With $g(x,y) := \delta_{\mathcal{F}_\epsilon}(x,y)$ and $h(x,y) := \mathbb{1}^\top Dy^p$ we now obtain the iteration in Algorithm 2 below. By [Man96, Thm. 4.2] additionally have the following finite termination result for Algorithm 2.

Proposition 3.2. *Algorithm 2 generates a finite sequence $(x^{(k)}, y^{(k)})_{k\in\mathbb{N}}$ with strictly decreasing objective function values for the ℓ_p-minimization problem* (2.20) *with $p \in (0,1)$, and terminating at an iteration $K \in \mathbb{N}$ satisfying the following minimum principle necessary optimality condition*

$$\left(\left(Dy^{(K)}\right)^{1/q-1}\right)^\top (y - y^{(K)}) \geq 0, \qquad (x,y) \in \mathcal{F}_\epsilon, \tag{3.7}$$

which states that $(x^{(K)}, y^{(K)})$ is a stationary point of (3.6).

3.3 DC-Programming for ℓ_{1-2}-Minimization

An solution approach [YLHX15] for (1.3) is to replace the ℓ_0-minimization by minimization of a non-convex yet Lipschitz continuous metric ℓ_{1-2} and consider

$$\min \|x\|_1 - \|x\|_2 \qquad \text{subject to} \quad Ax = b. \tag{3.8}$$

Similarly we can consider for the minimization of (1.1) a proxy

$$\min \|Dx\|_1 - \|Dx\|_2 \qquad \text{subject to} \quad x \in \mathcal{P} \subset \mathbb{R}^n, \tag{3.9}$$

with $\mathcal{P}$ from (1.2) or $\mathcal{P} = \{x \in \mathbb{R}^n \colon Ax = b, Cx \geq d, \|x\|_\infty \leq 1\}$. See Fig. 3.1 for an illustration of different sparsity measures. It is immediate to recast (3.9) as a difference of two convex functions

$$\min \underbrace{\|x\|_1 + \delta_{\mathcal{P}}(x)}_{=:g(x)} - \underbrace{\|x\|_2}_{=:h(x)}. \tag{3.10}$$

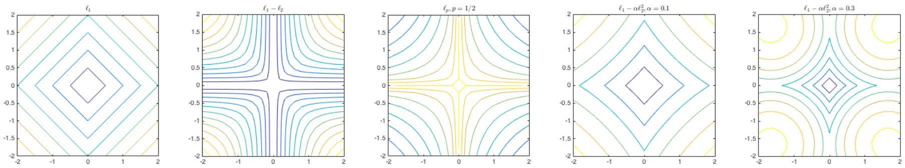


Figure 3.1: Contours of four sparsity metrics: ℓ_1, ℓ_{1-2}, $\ell_{\frac{1}{2}}$ and $\ell_1 - \alpha\ell_2^2$, $\alpha \in \{0.1, 0.3\}$. The level curves of ℓ_{1-2} tend to the axes as the values get small and are more similar to ℓ_0. In Sect. 2.2 we have shown that minimizing ℓ_p, for $p < 1$ small but finite, over bounded polyhedra is equivalent to ℓ_0-minimization. Our numerical experiments will show however that $\ell_1 - \alpha\ell_2^2$-minimization seams to be (numerically) more efficient, since it better avoids local minima.

Using that

$$\partial h(x) = \begin{cases} \frac{x}{\|x\|_2}, & \text{if } x \neq 0, \\ \mathbb{B}_1(0), & \text{if } x = 0, \end{cases} \tag{3.11}$$

we can specialize Algorithm 1 for (3.10) to the following iteration summarized in Algorithm 3 below.

Algorithm 3: DC Algorithm (DCA) for ℓ_{1-2}-Minimization.

```
**Input:** Good initialization $x$.
**Output:** Approximate local minimizer $\hat{x}$ of problem (3.10).
begin
    repeat
        $y^{(k)} = \begin{cases} x^{(k)}/\|x^{(k)}\|_2, & \text{if } x^{(k)} \neq 0, \\ 0, & \text{if } x^{(k)} = 0, \end{cases}$
        $x^{(k+1)} \in \arg\min_{x \in \mathcal{P}} \{\|x\|_1 - \langle y^{(k)}, x \rangle\}$
    until some convergence criteria is met at $x^{(K)}$.
    $\hat{x} \longleftarrow x^{(K)}$.
```

3.4 DC-Programming for $\ell_1 - \ell_2^2$ Minimization

We note that a DC function f has infinitely many DC decompositions. E.g. if $f = g - h$, then $f = (g+k) - (h+k)$ for every $k \in \mathcal{F}_0(\mathbb{R}^n)$. The primal DC corresponding to the two DC decompositions of the objective function f are identical, but their dual programs $h^* - g^*$ are quite different. Hence the DCA relative to these DC decompositions is also different. In other words, there are as many DCAs as there are DC decompositions of the objective function f. It is useful to find a suitable DC decomposition of f since it may have an important influence on the efficiency of the DCA. It is common to make the DC components of the primal objective function $f = g - h$ strongly convex. A straight-forward approach is to choose above $k(x) = \frac{1}{2}\|x\|_2^2$. If $f(x) = (\|x\|_1 + \frac{1}{2}\|x\|_2^2) - (\|x\|_2 + \frac{1}{2}\|x\|_2^2)$ like in Sect. 3.3, then DC Algorithm 1 will require to solve in each iteration a quadratic program. We prefer however to solve linear programs instead and consider the problem

$$\min \|Dx\|_1 - \alpha\|Dx\|_2^2 \qquad \text{subject to} \quad x \in \mathcal{P} \subset \mathbb{R}^n, \tag{3.12}$$

with $\mathcal{P}$ from (1.2) or $\mathcal{P} = \{x \in \mathbb{R}^n \colon Ax = b, Cx \geq d, \|x\|_\infty \leq 1\}$. We recast (3.12) as a DC-program

$$\min \underbrace{\|x\|_1 + \delta_{\mathcal{P}}(x)}_{=:g(x)} - \underbrace{\alpha\|x\|_2^2}_{=:h(x)}. \tag{3.13}$$

Using that

$$\partial h(x) = \nabla h(x) = 2\alpha x \tag{3.14}$$

we can specialize Algorithm 1 for (3.13) to Algorithm 4 below.

Algorithm 4: DC Algorithm (DCA) for $\ell_1 - \alpha\ell_2^2$-Minimization.

```
Input: Good initialization $x$, $\alpha \in (0,1)$.
Output: Approximate local minimizer $\hat{x}$ of problem (3.13).
begin
    repeat
        $x^{(k+1)} \in \arg\min_{x\in\mathcal{P}} \left\{\|x\|_1 - \langle 2\alpha x^{(k)}, x\rangle\right\}$
    until some convergence criteria is met at $x^{(K)}$.
    $\hat{x} \longleftarrow x^{(K)}$.
```

4 Superiorization

4.1 Superiorization: Basic Concepts.

Superiorization is a recently introduced methodology which gains increasing interest and recognition, as evidenced by the dedicated special issue entitled: “Superiorization: Theory and Applications”, in the journal *Inverse Problems* [CHE34]. The state of current research on superiorization can best be appreciated from the “Superiorization and Perturbation Resilience of Algorithms: A Bibliography compiled and continuously updated by Yair Censor”[Cenml]. In addition, [Her14], [Cen15] and

[RP17, Section 4] are recent reviews of interest. Research works in this bibliography include a variety of reports ranging from new applications to new mathematical results of the foundation of superiorization.

This methodology is heuristic and its goal is to find certain good, or superior, solutions to optimization problems. More precisely, suppose that we want to solve a certain optimization problem, for example, minimization of a convex function under constraints (for an approach which considers the superiorization methodology in a much broader form, see [RP17, Section 4]) then, solving the full problem can be rather demanding from the computational point of view, but solving part of it, say the feasibility part (namely, finding a point which satisfies all the constraints) is, in many cases, less demanding. Suppose further that our algorithmic scheme which solves the feasibility problem is *perturbation resilient*, that is, it converges to a solution of the feasibility problem despite perturbations which may appear in the algorithmic steps due to noise, computational errors, and so on.

Under these assumptions, the superiorization methodology claims that there is an advantage in considering perturbations in an active way during the performance of the scheme which tries to solve the feasibility part. What is this advantage? It may simply be a solution (or an approximation solution) to the feasibility problem which is found faster thanks to the perturbations; it may also be a feasible solution x' which is better than (or superior) feasible solutions x which would have been obtained without the perturbations, where this "superiority "is measured with respect to some given cost/merit function ϕ, namely $\phi(x') \leq \phi(x)$ is required (and hopefully $\phi(x')$ will be much smaller than $\phi(x)$).

Observe that in the case of convex optimization problem, when the objective ϕ is a convex function, the perturbation can be chosen in a non-ascending direction, as the following example: take $s^{(k,n)} \in \partial\phi(y^{(k,n)})$ where $\partial\phi(z)$ is the subgradient set of ϕ at z, and define $v^{(k,n)} = -s^{(k,n)}/\left\|s^{(k,n)}\right\|$, if $0 \notin \partial\phi(y^{(k,n)})$, and $v^{(k,n)} = 0$, if $0 \in \partial\phi(y^{(k,n)})$; See for example [BDHK07, CDH$^+$14].

Before we introduce Censor et al. superiorization "without gradients" [CHS16], which is more general and relevant to our results, we wish to present the superiorization methodology in the convex settings. It is clear that in the non-convex case, which is of our interest in this paper, the major question is how to choose a nonascent direction for ϕ in Algorithm 5 below.

Let $\phi : \mathbb{R}^n \to \mathbb{R}$ be a real-valued convex continuous function and let for simplicity of presentation, assume here that $\Gamma = \mathbb{R}^n$. The Superiorized Version of the Basic Algorithm $\mathfrak{A}$ that we consider here is based on [CZ15, Algorithm 4.1] and is presented in Algorithm 5.

Algorithm 5: Superiorized Version of the Basic Algorithm $\mathfrak{A}$

```
Input: $N \in \mathbb{N}$ and $y^{(0)} \in \mathbb{R}^n$ a user-chosen vector.
Output: A solution to problem $\mathbb{T}$ which is superior with respect to $\phi$.
begin
    Given a current iteration vector $y^{(k)}$ pick an $N_k \in \{1, 2, \dots, N\}$:
    repeat
        $y^{(k,0)} \leftarrow y^{(k)}$.
        Pick a $0 < \beta_{k,n} \leq 1$ such that $\sum_{k=0}^{\infty} \sum_{n=0}^{N_k - 1} \beta_{k,n} < \infty$.
        Pick a $v^{(k,n)}$ that is a direction of non-ascend for $\phi$ at $y^{(k,n)}$.
        Calculate the perturbed iterate $y^{(k,n+1)} \leftarrow y^{(k,n)} + \beta_{k,n} v^{(k,n)}$.
        set $n \leftarrow n + 1$
    until $n < N_k$
    Exit the loop with the vector $y^{(k,N_k)}$.
    Calculate $y^{(k+1)} \leftarrow \mathfrak{A}(y^{(k,N_k)})$ and set $k \leftarrow k + 1$.
```

4.2 Basic Algorithms for SFP

The *Convex Feasibility Problem* (CFP) stands at the core of the modeling of many inverse problems in various areas of mathematics and the physical sciences; for example in sensor networks, in radiation therapy treatment planning, in color imaging and in adaptive filtering, see e.g., [CAP88, Byr99, CDH10] and references therein. The CFP formulation is given next.

Problem 4.1. The Convex Feasibility Problem (CFP).
For $i = 1, \dots, p$ let $C_i \subseteq \mathbb{R}^n$ be closed and convex sets. The CFP is:

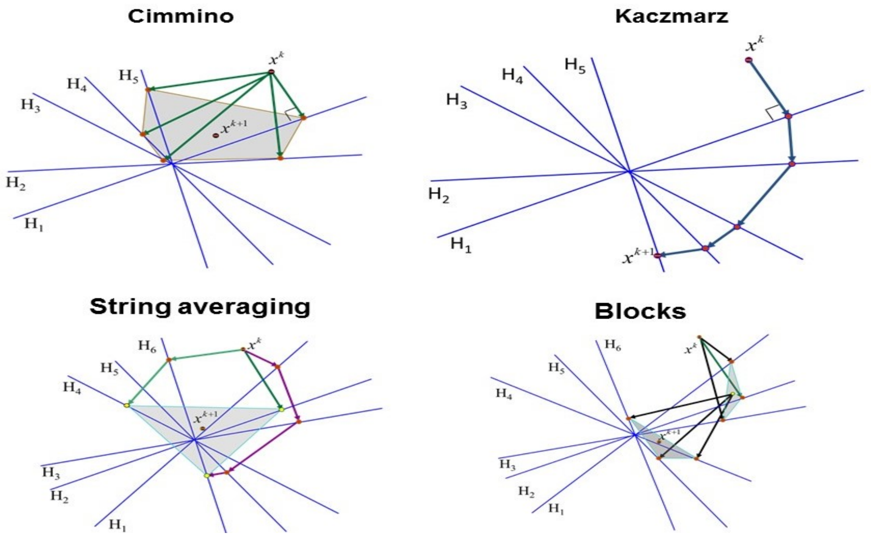
$$\text{find a point } x^* \in \boldsymbol{C} := \cap_{i=1}^{p} C_i. \tag{4.1}$$

In 1994, Censor and Elfving [CE94] introduced the *Split Convex Feasibility Problem* (SFP). This reformulation was employed for solving an inverse problem in intensity-modulated radiation therapy (IMRT) treatment planning, see [CBMAT06]. The problem formulates as follows.

Problem 4.2. The Split Convex Feasibility Problem (SFP).
Let $\mathbb{R}^n$ and $\mathbb{R}^m$ be two Euclidean spaces. Let $C \subseteq \mathbb{R}^n$ and $Q \subseteq \mathbb{R}^m$ be two non-empty, closed and convex sets, in addition given a bounded linear operator $A : \mathbb{R}^n \to \mathbb{R}^m$, the SFP is:

$$\text{find a point } x^* \in C \text{ such that } y^* = Ax^* \in Q. \tag{4.2}$$

A useful tool which is used frequently for solving CFPs as well as SFPs is the class of projection methods, (see, e.g., [Byr08, Ceg12, CZ97]). These are iterative algorithms that use projections onto sets, relying on the principle that when a family of sets is present, then projections onto the given individual sets are easier to perform than projections onto other sets (intersections, image sets under some transformation, etc.) that are derived from the given individual sets. See illustrations in Figure 4.1 which is taken from [CEH01]. Their main advantage, which makes them successful in real-world applications, is computational. They commonly are able to handle huge-size problems of dimensions beyond which more sophisticated methods cease to be efficient or even applicable due to memory requirements.


Cimmino

Kaczmarz

String averaging

Blocks

Figure 4.1: Different projection methods for the linear case. The figure is reproduced from [CEH01].

In order to discuss a more general case, we assume (standard assumption) that the set C (also Q for the SFP) can be represented as a sublevel set of a convex and subdifferential function $c : \mathbb{R}^n \to \mathbb{R}$, that is

$$C = \{x \in \mathbb{R}^n \colon c(x) \leq 0\}. \tag{4.3}$$

Next, we define the set

$$\widetilde{C} = \{y \in \mathbb{R}^n \colon c(x) + \langle \xi, y - x \rangle \leq 0\}, \tag{4.4}$$

where $\xi \in \partial c(x)$ is the subgradient of c at a point x. In this case the subgradient projection onto $\widetilde{C}$, which is an outer approximation to the set C, since $C \subseteq \widetilde{C}$ is calculated as follows. Let $\xi \in \partial c(x)$

$$P_{\widetilde{C}}(y) := \begin{cases} y - \frac{c(x)}{\|\xi\|^2}\xi & \text{if } \xi \neq 0, \\ y & \text{otherwise.} \end{cases} \tag{4.5}$$

Observe that if $\xi \neq 0$ then $\widetilde{C}$ is a half-space and the whole space otherwise.

So, if all the sets of the CFP, Problem 4.1, have a sublevel sets representation, that is for all i

$$C_i = \{x \in \mathbb{R}^n \colon c_i(x) \leq 0\} \tag{4.6}$$

the *Cyclic Subgradient Projection* (CSP) method [CL82] can be used successfully. The CSP iterative step is formulated as follows.

$$x^{k+1} = \begin{cases} x^{(k)} - \lambda_k \frac{c_{i(\nu)}(x^{(k)})}{\|\xi^{(k)}\|^2}\xi^{(k)} & c_{i(\nu)}(x^{(k)}) > 0 \\ x^{(k)} & c_{i(\nu)}(x^{(k)}) \leq 0 \end{cases} \tag{4.7}$$

where $\xi^{(k)} \in \partial c_{i(\nu)}(x^{(k)})$, $\lambda_k \in [\epsilon_1, 2 - \epsilon_2]$ are called *relaxation parameters* for arbitrary $\epsilon_1, \epsilon_2 > 0$ and $\{i(\nu)\}$ is a sequence of indices according to which individual sets C_i are chosen, in this case cyclic: $i(\nu) = \nu \bmod m + 1$. In the linear case CFP, this method reduces to the well known method of Kaczmarz [Kac37] or *Algebraic Reconstruction Technique* (ART) and *POCS* in the field of image reconstruction from projection, see [BGH70].

Observe that our interest here is the binary reconstruction problem (2.26) which is phrased as (2.27), this means that we use projection methods to find a feasible point within the bounded polyhedron $\mathcal{P}$.

4.3 Descent and Nonascent Directions for ℓ_0

Further we apply the superiorization methodology to the objective $\phi(x) = \|x\|_0$. Since ϕ is not differentiable in this case we need to introduce non-ascending directions for ϕ. This is a special case of an idea called superiorization ”without gradients”, which was suggested to us based on his work in [CHS16], by Censor in a private communication. Let $x \in \mathbb{R}^n$, set $I = \operatorname{supp}(x)$. Choose one $j \in I$ and define the vector $w = (w_i) \in \mathbb{R}^n$ by

$$w_i = \begin{cases} x_i & \text{if } i \neq j \\ 0 & \text{else} \end{cases} . \tag{4.8}$$

Clearly $\|w\|_0 \leq \|x\|_0$. If $\|x\|_0 = s$ for some $0 < s \leq n$, then (4.8) reduces to the projection of the vector x onto the non-convex set

$$Q := \{y \in \mathbb{R}^n \colon \|y\|_0 \leq s - 1\} \tag{4.9}$$

see [PZC+17].

An alternative for choosing a non-ascend direction for $\phi(x) = \|x\|_0$ is based on (2.24) from Prop. 2.3. We assume that $0 \leq x \leq \mathbb{1}$ as motivated in Section 2.3 and Section 2.4, and choose for simplicity $D = I$. First observe that for any $p \in (0,1)$, the gradient of $x \mapsto \mathbb{1}^\top x^p$ is $px^{p-1} := (px_1^{p-1}, \cdots, px_n^{p-1})$, where $x^p := (x_1^p, \cdots, x_n^p)$ (element-wise). Further choose $\alpha \in (0,1)$ and define $w = x - \alpha p x^{p-1}$. Then $\|w\|_0 \leq \|x\|_0$. Indeed, if for some $i \in [n]$ we have $x_i = 0$ then clearly $w_i = 0$. Otherwise it might happen that $x_i - \alpha p x_i^{p-1} = 0$.

Further work will include superiorization for $\phi(x) = \|x\|_1 - \|x\|_2$. In this case we have a DC function which is non-convex but Lipschitz continuous. For such functions we wish to present a general technique (see [BKS08]) to compute non-ascend/descent directions for nonsmooth but Lipschitz continuous objective functions.

Algorithm 6: Computation of descent directions for nonsmooth but Lipschitz continuous functions.

```
Input: Choose any $g^1 \in S_1$ and $e \in G$. Given $z \in P$, $\lambda > 0$, $\alpha \in (0,1]$,
    $c \in (0,1)$ and a tolerance $\delta > 0$. Compute
    $v^1 = \Gamma^i\left(x, g^1, e, z, \lambda, \alpha\right) = \left(\Gamma_1^i, \ldots, \Gamma_n^i\right)$, where

    $\Gamma_j^i := \begin{cases} \left[z(\lambda)\alpha^j e_j\right]^{-1}\left[\phi\left(x^j\right) - \phi\left(x^{j-1}\right)\right] & \text{if } i \neq j \\ (\lambda g_i)^{-1}\left[\phi\left(x + \lambda g\right) - \phi\left(x\right) - \lambda \sum_{j=1, j\neq i} \Gamma_j^i g_j\right] & i = j \end{cases}$

    and set $\overline{D}_1(x) = \left\{v^1\right\}$.
Output: A descent direction $g$ of $\phi$ at a point $x \in \mathbb{R}^n$.
begin
    Given a current iteration vector $g^k$ and the set $\overline{D}_k(x)$
    repeat
        Compute $i = \arg\max\left\{\left|g_j^k\right| \mid j = 1, \ldots, n\right\}$ and
          $v^k = \Gamma^i\left(x, g^k, e, z, \lambda, \alpha\right)$.
        Find the vector $\left\|w^k\right\|^2 = \min\left\{\|w\|^2 \mid w \in \overline{D}_k(x)\right\}$.
        repeat
            Compute $g^{k+1} = -\left\|w^k\right\|^{-1} w^k$ and
              $i = \arg\max\left\{\left|g_j^{k+1}\right| \mid j = 1, \ldots, n\right\}$.
            Set $v^{k+1} = \Gamma^i\left(x, g^{k+1}, e, z, z, \alpha\right)$ and
              $\overline{D}_{k+1}(x) = \text{conv}\left\{\overline{D}_k(x) \cup \left\{v^{k+1}\right\}\right\}$.
        until $\phi\left(x + \lambda g^{k+1}\right) - \phi\left(x\right) > -c\lambda\left\|w^k\right\|$
    until $\left\|w^k\right\| \leq \delta$
    Exit the loop with the vector $g^{k+1}$.
    Calculate $g^{k+1}$ and set $k \leftarrow k + 1$.
```

5 Experiments, Discussion

In this section we validate the *local* minimization approach based on DC-programming on the two image reconstruction models from Sect. 2.3 and Sect. 2.4 but also compare DC minimization to ℓ_0-superiorization.

5.1 Experimental Set-Up

We consider the reconstruction of three test images from few tomographic projections (2.25). The first and second image are the binary test images from [Bat07] and [HSH12] representing a vascular system containing larger and smaller vessels, see Fig. 5.1 (left), (middle left). The third image is the Shepp-Logan MATLAB phantom, as shown in Fig. 5.1 (middle right).

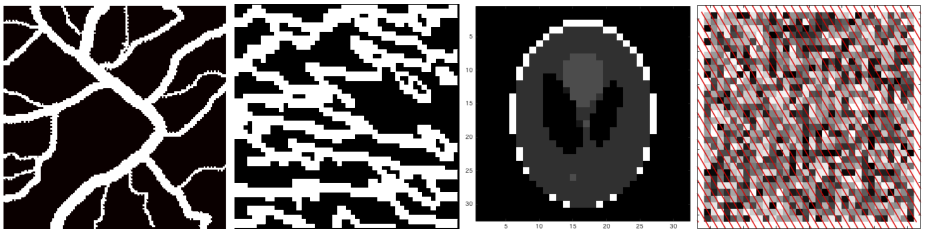


Figure 5.1: Vessel test image from [Bat07] (left). Vessel test image from [HSH12] (middle left). The 32×32 Shepp-Logan phantom (middle right). We illustrate how such a 32×32 image is sampled (right) along 45 parallel and equidistant lines that are all perpendicular to $\theta := (\cos(30°), \sin(30°))^\top$.

The measurements are described by $q = Mu$, where

$$M = \begin{pmatrix} M_{\theta_1}^\top & M_{\theta_2}^\top \dots & M_{\theta_{\mathtt{nA}}}^\top \end{pmatrix}^\top$$

and each block matrix M_{θ_i} corresponds to a different projecting angle, see Fig. 5.1 (right) for a parallel ray geometry that corresponds to one such angle. We use the MATLAB routine `paralleltomo.m` from the AIR Tools package [HSH12] that implements such a tomographic matrix for a given vector of angles. We set $\mathtt{N} \in \{32, 64, 128\}$ the image size and use the default value of `p`, i.e. the number of parallel rays for each angle `p = round(sqrt(2)*N)`.

For the description of the superiorization results we define some of the used parameters. We denote by **N** the dimension of the image (that is $N \times N$ pixels image), **nA** is the number of projections which are used to sample the image, **OLI** represent the number of iterations in the outer loop, **ILI** is the number of iterations of the basic algorithm in the inner loop, **SI** is the total number of superiorizations per outer iteration. In addition, **SR** when the stopping rule $\|Ax - b\| < 10^{-5}$ achieved, **Res** stands for the residual, that is for the reconstructed solution x **Res** is $\|Ax - b\|$, closely related, **Diff** is the norm difference between the original image (IOrg) and the reconstructed (IRec) one, that is $\|$IOrg-IRec$\|_2$.

5.2 DC-Programming Results

5.2.1 Sparse Binary Tomography

We aim to reconstruct *exactly* the two *binary* test images $u \in \{0,1\}^n$ via the three DC programs from Sect. 3.2, 3.3 and 3.4 from a *minimal* number of equidistant projection angles θ_i within the interval $[0°, 180°]$. To find the minimal number of projections we vary the number of projecting directions and certify a successful reconstruction if the (relative) distance of the reconstruction, further denoted by u_{rec} to the original signal u is *below* a numerical "zero", i.e. $\frac{\|u_{rec}-u\|_2}{\mathtt{N}} \leq 10^{-6}$. The starting point for each DC algorithm is the default starting point for the LP solver [ApS15].

We also determine the minimal number of projecting angles `nA` such that the matrix M is overdetermined or recovery of u via least-squares

$$\min \|Mu - q\|_2^2 \tag{5.1}$$

is successful. Results are summarized in the first six rows of Table 1.

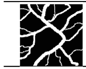
	dim	`nA` (overdet)	`nA` (LS)	`nA` (ℓ_1)	`nA` (ℓ_p)	`nA` (ℓ_{1-2})	`nA` ($\ell_1 - \alpha\ell_2^2$)
	$N = 32$	23	29	10	10	10	10
	$N = 64$	46	56	13	13	13	13
	$N = 128$	91	110	17	17	17	**15** ($\alpha = 0.3$)
	$N = 32$	23	30	12	12	12	12
	$N = 64$	46	56	16	16	16	16
	$N = 128$	91	110	20	20	20	**19** ($\alpha = 0.1$)
	$N = 32$	23	29	11	11	**10**	**9** ($\alpha = 0.1$)
	$N = 64$	46	56	14	14	**13**	**12** ($\alpha = 0.3$)
	$N = 128$	91	110	14	**13**	**11**	**11** ($\alpha = 0.1$)

Table 1: Results for DC-programming. The 1st column shows the test image used and the 2nd column its dimension. The 3rd column indicates the numbers of projecting angles that are necessary for obtaining an overderdetermined system. Column '`nA` (LS)' shows the minimal number of projecting angles such that each test images is recovered via (5.1). Column '`nA` (ℓ_1)' gives the minimal `nA` such that ℓ_1-regularization is exact. Column '`nA` (ℓ_p)' gives the minimal `nA` such that ℓ_p-regularization is exact. Several values of $p \in \{1/2, 1/5, 1/10, 1/15\}$ yield similar results. Column '`nA` (ℓ_{1-2})' gives the minimal `nA` such that ℓ_{1-2}-regularization is exact, while the last column gives the minimal `nA` such that $\ell_1 - \alpha\ell_2^2$-regularization is exact. The used α values are also shown. We observe that $\ell_1 - \alpha\ell_2^2$-regularization always improved on the ℓ_1-results. For the gradient-sparse Shepp-Logan test image DC-programming always improves on the convex model.

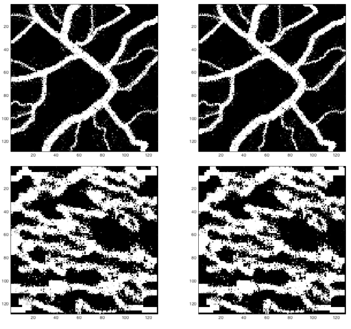


Figure 5.2: *Non-exact* recovery of binary images by DC-programming. The first row shows the Batenburg vessel image reconstructed from nA $= 15$ projections via ℓ_p-minimization (left column) and via ℓ_{1-2}-minimization (right column). The absolute errors are $\|u_{rec} - u\|_2 \approx 17.4929$ for ℓ_p-minimization and $\|u_{rec} - u\|_2 \approx 18.2209$ for ℓ_{1-2}-minimization. The AIR Tools vessel is denser and requires more projections. We show reconstructions from nA $= 19$ projections via ℓ_p-minimization (left column) and via ℓ_{1-2}-minimization (right column). The errors are $\|u_{rec} - u\|_2 \approx 34.0289$ and $\|u_{rec} - u\|_2 \approx 34.5571$, respectively.

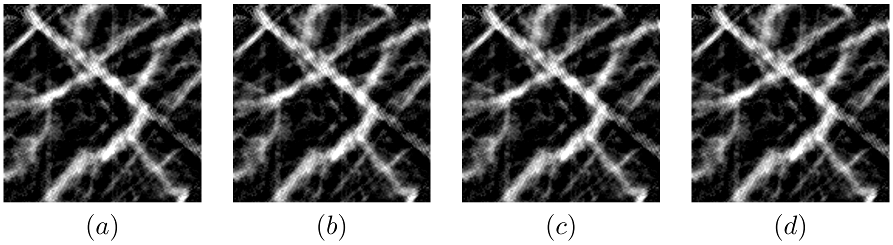
5.2.2 Gradient-Sparse Discrete Tomography

Here we approximate the model presented in 2.4 using the three proxies for ℓ_0-minimization: ℓ_p, with $p \in \{1/2, 1/5, 1/10, 1/15\}$, ℓ_{1-2} and $\ell_1 - \alpha\ell_2^2$, with $\{\alpha \in \{0.1, 0.3\}\}$. The feasible set is described by *both* projecting and box constraints. Results for the gradient-sparse Shepp-Logan phantom are summarized in the last three rows of Table 1. We note that DC programming *always* reduced the number of projecting angles nA when compared to $\|\nabla u\|_1$-minimization subject to projecting and box constraints.

5.3 Superiorization Results

In this subsection we illustrate the ℓ_0-superiorization approach on the image reconstruction models from Sect. 2.3. We consider only the Batenburg vesel image. We choose two variants for the non-ascend directions as explained in Subsection 4.3.

Our experiments are devided into two parts. In the first part, in Figure 5.3, we choose the following parameters N=128, nA=15, OLI=30 and ILI=10m, where m is the number of rows of the matrix A. As explained above, one of natural question when applying the superiorization methodology, is the so-called Balancing Question: How to divide the efforts that a superiorization algorithm invests in target function reduction steps (perturbations) versus the efforts invested in feasibility-seeking. Trying to answer this question our experiments in Figure 5.3 are divided into four. In the first, that is in (a) we dont use any superiorization, just feasibility-seeking algorithm (here Kaczmarz), see Figure 4.1. In (b) one superiorization (=one perturbation) per outer iteration is used. In this case we noticed that the computational time/effort of the total procedure is higher but the reconstructed image is better with respect to step **Res** and **Diff**. In (c) we push that even further and tried to used a minimum of 10 superiorization steps for each outer iteration. In this case, the stopping criteria is achieved only after 13 outer iterations, compared to 3 in the pervious scenarios but again we witness that the result is slightly better with respect to **Res**. Finally, in (d), similarly to (c), we used a minimum of 10 superiorization steps for each outer iteration but also included the addition $[0, 1]^n$ box constraints for the feasibility-seeking algorithm. Again there are improvements in **Res** and **Diff** and even the stopping criteria is reached after 11 compared to 13 in (c).



(a) (b) (c) (d)

Figure 5.3: (a) SI=0, SR=3 and Res=$8.2449 * 10^{-6}$, Diff=54.9293. (b) SI=1, SR=3 and Res=$8.0233 * 10^{-6}$, Diff=49.2490. (c) SI=10, SR=13 and Res=$9.656 * 10^{-6}$. (d) SI=10, SR=11 and Res=$9.656 * 10^{-7}$, Diff=47.9286.

In the second part of our experiments the second non-ascend direction (2.24) is used with different choices of the image's size N. In the first line, (a) and (b), we show how for a relatively small image (N=16) using the even one perturbation per outer iteration improves significantly the reconstructed image and yield a sparser image. In (b) we choose $p = 2$. In the second line, (c) and (d) we again compared the performances with and without perturbations for an image of size 32 and see in (d) that **Diff** is slightly better, but remember that only one "nonexpansive" perturbation is used. Finally, in (e) we decided to test our scheme for a larger image of size 128 but with a fewer number of projections (nA=6) and also decrease the total number of the outer and inner iterations. Although the reconstructed image contains many artifacts, it is still very impressive to see that the measurements **Res** and **Diff** are quite good which suggests that this approach can be used an alternative for the DC-programming.

6 Conclusion

Motivated by sparse and gradient-sparse image recovery from few tomographic projections we compare two recent strategies for solving the related ℓ_0-minimization

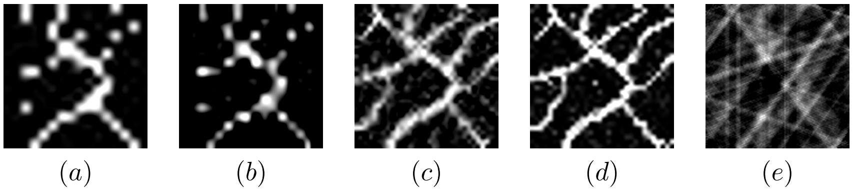


Figure 5.4: (a) SI=0, $N = 16, nA = 15, OLI = 10, ILI = 5m$, Diff=7.5427. (b) SI=1, N=16, nA=15, OLI=10, ILI=5m, $p = 2$, Diff=6.3964. (c) SI=0, N=32, nA=15, OLI=10, ILI=5m, Diff=14.1781 (d) SI=1, N=32, nA=15, OLI=10, ILI=5m, $p = 2$, threshold $\epsilon = 10^{-5}$, Diff=13.1946. (e) SI=1, N=128, nA=6, OLI=3, ILI=2m, $p = 0.01$ threshold $\epsilon = 10^{-5}$, Res=$5.6340 * 10^{-9}$, Diff=53.4470.

problem (1.1): the DC-programming approach and the ℓ_0-superiorization. DC-programming for non-convex ℓ_0-proxies not always improves on the convex ℓ_1-relaxation of ℓ_0, but often gets trapped in local minima. On the other hand ℓ_0-superiorization yields comparable results with significantly less computational effort. For ℓ_0-superiorization, we suggest two new non-ascend directions and test their performances for different parameter. Hence, our preliminary results shows that the ℓ_0-superiorization can be used as an alternative approach.

Acknowledgment

AG is supported by the EU FP7 IRSES program STREVCOMS, grant no. PIRSES-GA-2013-612669.

References

[ApS15] MOSEK ApS, *The mosek optimization toolbox for matlab manual. version 7.1 (revision 28).*, 2015.

[Bat07] K. J. Batenburg, *A Network Flow Algorithm for Reconstructing Binary Images from Discrete X-rays*, J. Math. Imaging Vis. **27** (2007), no. 2, 175–191.

[BDHK07] D. Butnariu, R. Davidi, G. T. Herman, and I. G. Kazantsev, *Stable convergence behavior under summable perturbations of a class of projection methods for convex feasibility and optimization problems*, IEEE Journal of Selected Topics in Signal Processing **1** (2007), 540–547.

[BGH70] R. Bender, R. Gordon, and G. T. Herman, *Algebraic reconstruction techniques (art) for three-dimensional electron microscopy and x-ray photography*, Journal of Theoretical Biology **29** (1970), 471–81.

[BKS08] A. M. Bagirov, B. Karasöen, and M. Sezer, *Discrete gradient method: drivative-free method for nonsmooth optimization*, Journal of Optimization Theory and Applications **137** (2008), no. 2, 317334.

[Byr99] C. L. Byrne, *Iterative projection onto convex sets using multiple Bregman distances*, Inverse Problems **15** (1999), no. 5, 1295–1313. MR 1715366

[Byr08] C. L. Byrne, *Iterative methods for fixed point problems in hilbert spaces*, Applied Iterative Methods, AK Peters, Wellsely, MA, USA, 2008.

[CAP88] Y. Censor, M. D. Altschuler, and W. D. Powlis, *On the use of Cimmino's simultaneous projections method for computing a solution of the inverse problem in radiation therapy treatment planning*, Inverse Problems **4** (1988), no. 3, 607–623. MR 965639

[CBMAT06] Y. Censor, T. Bortfeld, B. Martin, and A A. Trofimov, *unified approach for inversion problems in intensity-modulated radiation therapy*, Physics in Medicine and Biology **51** (2006), 2353–2365.

[CDH10] Y. Censor, R. Davidi, and G. T. Herman, *Perturbation resilience and superiorization of iterative algorithms*, Inverse Problems **26** (2010), no. 6, 065008, 12. MR 2647162

[CDH$^+$14] Y. Censor, R. Davidi, G. T. Herman, R. W. Schulte, and L. Tetruashvili, *Projected subgradient minimization versus superiorization*, Journal of Optimization Theory and Applications **160** (2014), 730–747.

[CE94] Y. Censor and T. Elfving, *A multiprojection algorithm using Bregman projections in a product space*, Numer. Algorithms **8** (1994), no. 2-4, 221–239. MR 1309222

[Ceg12] A. Cegielski, *Iterative methods for fixed point problems in hilbert spaces*, Lecture Notes in mathematics 2057, Springer-Verlag, Berlin, Heidelberg, Germany, 2012.

[CEH01] Y. Censor, T. Elfving, and G. T. Herman, *Averaging strings of sequential iterations for convex feasibility problems*, Inherently parallel algorithms in feasibility and optimization and their applications (Haifa, 2000), Stud. Comput. Math., vol. 8, North-Holland, Amsterdam, 2001, pp. 101–113. MR 1853219

[Cen15] Y. Censor, *Weak and Strong Superiorization: Between Feasibility-Seeking and Minimization*, An. Şt. Univ. Ovidius Constanţa **23** (2015), 41–54.

[Cenml] Y. Censor, *Superiorization and perturbation resilience of algorithms: A bibliography compiled and continuously updated*, http://math.haifa.ac.il/yair/bib-superiorization-censor.html.

[CHE34] Y. Censor, G. T. Herman, and M. Jiang (Guest Editors), *Superiorization: Theory and applications*, Special Issue of the journal *Inverse Problems*, Volume 33, Number 4, 2017, http://iopscience.iop.org/issue/0266-5611/33/4.

[CHS16] Y. Censor, H. Heaton, and R. Schulte, *Superiorization with component-wise perturbations and its application to image reconstruction*, November 4, 2016.

[CL82] Y. Censor and A. Lent, *Short Communication: Cyclic subgradient projections*, Math. Programming **24** (1982), no. 1, 233–235. MR 1552955

[CS08] R. Chartrand and V. Staneva, *Restricted Isometry Properties and Non-convex Compressive Sensing*, Inverse Problems **24** (2008), no. 3, 035020.

[CZ97] Y. Censor and S. A. Zenios, *Parallel optimization: Theory, algorithms, and applications*, Oxford University Press, New York, NY, USA, 1997.

[CZ13] Y. Censor and A. J. Zaslavski, *Convergence and perturbation resilience of dynamic string-averaging projection methods*, Computational Optimization and Applications **54** (2013), no. 1, 65–76.

[CZ15] ______, *Strict Fejér monotonicity by superiorization of feasibility-seeking projection methods*, J. Optim. Theory Appl. **165** (2015), no. 1, 172–187. MR 3327420

[FM11] G. M. Fung and O. L. Mangasarian, *Equivalence of Minimal ℓ_0- and ℓ_p-Norm Solutions of Linear Equalities, Inequalities and Linear Programs for Sufficiently Small p*, J. Optim. Theory. Appl. **151** (2011), no. 1, 1–10.

[FR13] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*, Springer, 2013.

[GR84] K. Goebel and S. Reich, *Uniform convexity, hyperbolic geometry, and nonexpansive mappings*, Monographs and Textbooks in Pure and Applied Mathematics, vol. 83, Marcel Dekker, Inc., New York, 1984. MR 744194

[Her14] G. T. Herman, *Superiorization for image analysis, in:*, 2014.

[HK99] G. T. Herman and A. Kuba, *Discrete Tomography: Foundations, Algorithms, and Applications*, Applied and Numerical Harmonic Analysis, Birkhauser, December 1999 1999.

[HSH12] P. C. Hansen and M. Saxild-Hansen, *AIR Tools – A MATLAB Package of Algebraic Iterative Reconstruction Methods*, J. Comput. Appl. Math. **236** (2012), no. 8, 2167–2178.

[HT99] R. Horst and N. Thoai, *DC programming: Overview*, J. Optim. Theory Appl. **103** (1999), no. 1, 1–43.

[Kac37] S. Kaczmarz, *Angenöherte auflösung von systemen linearer gleichungen*, Bulletin de l'Académie Polonaise des Sciencesat Lettres **A35** (1937), 355–357.

[Man96] O. L. Mangasarian, *Machine Learning via Polyhedral Concave Minimization*, Applied Mathematics and Parallel Computing: Festschrift for Klaus Ritter (H. Fischer, B. Riedmüller, and S. Schäffler, eds.), Physica-Verlag HD, Heidelberg, 1996, pp. 175–188.

[Nat95] B. K. Natarajan, *Sparse Approximate Solutions to Linear Systems*, SIAM J. Computing **24** (1995), no. 5, 227–234.

[PDEB86] T. Pham Dinh and S. El Bernoussi, *Algorithms for Solving a Class of Nonconvex Optimization Problems. Methods of Subgradients*, Fermat Days 85: Mathematics for Optimization (J.-B. Hiriart-Urruty, ed.), North-Holland Mathematics Studies, vol. 129, North-Holland, 1986, pp. 249–271.

[PDTH97] T. Pham Dinh and A. L. Thi Hoai, *Convex Analysis Approach to D.C. Programming: Theory, Algorithms and Applications*, Acta Math. Viet. **22** (1997), no. 1, 289–355.

[PDTH06] ______, *A D.C. Optimization Algorithm for Solving the Trust-Region Subproblem*, SIAM J. Optim. **8** (2006), no. 2, 476–505.

[PZC$^+$17] S. Penfold, R. Zalas, M. Casiraghi, M. Brooke, Y. Censor, and R. Schulte, *Sparsity constrained split feasibility for dose-volume constraints in inverse planning of intensity-modulated photon or proton therapy*, Physics in Medicine and Biology **62** (2017), 3599–3618.

[Roc70] R. T. Rockafellar, *Convex Analysis*, Princeton Mathematical Series, Princeton University Press, 1970.

[RP17] D. Reem and A. De Pierro, *A new convergence analysis and perturbation resilience of some accelerated proximal forward-backward algorithms with errors*, Inverse Problems **33** (2017), no. 2, 28pp.

[THPD05] A. L. Thi Hoai and T. Pham Dinh, *The DC (Difference of Convex Functions) Programming and DCA Revisited with DC Models of Real World Nonconvex Optimization Problems*, Annals of Operations Research **133** (2005), no. 1, 23–46.

[Tol78] J. F. Toland, *Duality in Nonconvex Optimization*, J Math Anal Appl. **66** (1978), no. 2, 399–415.

[YLHX15] P. Yin, Y. Lou, Q. He, and J. Xin, *Minimization of ℓ_{1-2} for Compressed Sensing*, SIAM J. Sci. Computing **37** (2015), no. 1, A536–A563.

Aviv Gibali
Department of Mathematics,
ORT Braude College, P.O. Box 78,
Karmiel 2161002, Israel.
Email: avivg@braude.ac.il

Stefania Petra
Mathematical Imaging Group,
Heidelberg University, Germany.
Email: petra@math.uni-heidelberg.de