

# A Benchmark Evaluation of Large-Scale Optimization Approaches to Binary Tomography

Stefan Weber<sup>1</sup>, Antal Nagy<sup>2</sup>, Thomas Schüle<sup>1</sup>,  
Christoph Schnörr<sup>1</sup>, and Attila Kuba<sup>2</sup>

<sup>1</sup> Dept. Math. & Comp. Science, CVGPR-Group, University of Mannheim,  
D-68131 Mannheim, Germany

{wstefan, schuele, schnoerr}@uni-mannheim.de

<sup>2</sup> Department of Image Processing and Computer Graphics,  
University of Szeged, Árpád tér 2. H-6720 Szeged, Hungary

{nagy, kuba}@inf.u-szeged.hu

**Abstract.** Discrete tomography concerns the reconstruction of functions with a finite number of values from few projections. For a number of important real-world problems, this tomography problem involves thousands of variables. Applicability and performance of discrete tomography therefore largely depend on the criteria used for reconstruction and the optimization algorithm applied. From this viewpoint, we evaluate two major optimization strategies, simulated annealing and convex-concave regularization, for the case of binary-valued functions using various data sets. Extensive numerical experiments show that despite being quite different from the viewpoint of optimization, both strategies show similar reconstruction performance as well as robustness to noise.

## 1 Introduction

Discrete tomography (DT) is an active field of research covering a number of important problems across various application areas [1, 2]. A key aspect of DT is the reconstruction of functions under non-standard conditions, in contrast to conventional tomography. A necessary condition for making such reconstructions feasible is to restrict the range of the functions to be reconstructed to a finite set. Challenging application problems that can be naturally modeled in this way include non-destructive testing [3], electron microscopy [5], and medical imaging [4, 6].

A major problem in connection with DT concerns optimization. In fact, most applications like DT in medical imaging involve thousands of variables representing the discrete-valued function to be computed. Solving such large-scale combinatorial problems to reach global optimality is generally not possible, hence, optimization strategies providing a good compromise between the quality of sub-optimal solutions and runtime are of primary interest.

For these reasons, we study in this paper two different optimization strategies that showed promising performance in recent work:

- The first strategy based on the classical approach of simulated annealing (SA). It is a stochastic optimization strategy, a random-search technique that is based on the physical phenomenon of metal cooling [10]. The system of metal particles, here the values of the image pixels, gradually reaches the minimum energy level where the metal freezes into a crystalline structure.
- The second optimization strategy, convex-concave regularization, was proposed in [7]. It combines convex relaxations of reconstruction functionals with concave minimization to enforce discrete decisions. A local minimum is determined by solving a sequence of convex optimization problems, each of which can be solved to global optimality. The method involves a single regularization parameter only requiring an application-dependent choice.

In Section 2 we briefly describe the general reconstruction problem. Section 3 details the algorithms related to the two optimization strategies which are evaluated. The evaluation criteria (data sets, performance measures, parameter settings) are specified in section 4, and our quantitative numerical results are presented and discussed in Section 5. We conclude and outline further work in Section 7.

## 2 Reconstruction Problem

We consider the reconstruction problem of transmission tomography for binary objects. As explained in Fig. 1(a), the imaging process is represented by the algebraic system of equations

$$Ax = b, \quad A \in \mathbb{R}^{m \times n}, \quad x \in \{0, 1\}^n, \quad b \in \mathbb{R}^m, \quad (1)$$

where  $A$  and  $b$  are given, and the binary indicator vector  $x$  representing the unknown object has to be reconstructed. Though we restrict ourselves here to parallel beam geometry, Fig. 1(b), this algebraic representation is general enough to suit other geometries as well.

## 3 Two Optimization Strategies

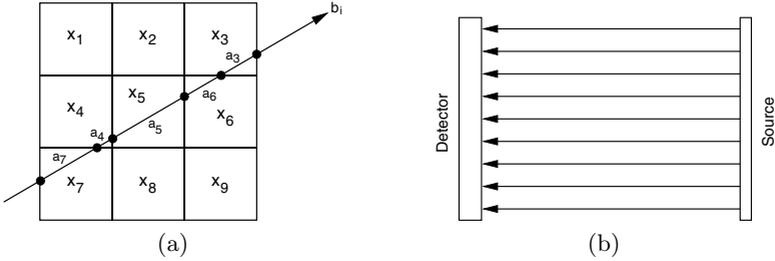
This section describes two approaches capable to numerically solve large-scale instances of the general reconstruction problem (1).

### 3.1 Simulated Annealing

Actually, a possible way of solving (1) at least approximately is to reformulate it as an optimization problem. Formally, we should find the minimum of the following objective function

$$C(x) = \|Ax - b\|^2 + \gamma \cdot \Phi(x), \quad \text{where } x \text{ is a binary-valued vector.} \quad (2)$$

The first term on the right hand side ensures that we have an  $x$  satisfying (1) at least approximately. The second term allows us to include *a priori* knowledge



**Fig. 1. (a) Discretization model for transmission tomography.** The measured projection data are given in terms of a vector  $b \in \mathbb{R}^m$ . Each component  $b_i$  corresponds to a projection ray measuring the absorption along the ray through the volume which is discretized into cells. The absorption  $a_j$  in each cell is assumed to be proportional to the density of the unknown object.  $x_1, x_2, \dots$  are binary variables indicating whether the corresponding cells belong to the object ( $x_k = 1$ ) or not ( $x_k = 0$ ). Assembling all projection rays into a linear system gives  $Ax = b$ ,  $x \in \{0, 1\}^n$ , from which the unknown binary object, represented by  $x$ , has to be determined. **(b) Parallel beam geometry.** Multiple projections are gathered by rotating the source-detector system around a center point.

about  $x$  into the optimization if there are several good binary vector candidates that keep  $\|Ax - b\|^2$  low. In our experiments we have used the following  $\Phi(x)$  function

$$\Phi(x) = \sum_{j=0}^n \sum_{l \in Q_j^m} g_{l,j} \cdot |\xi_j - \xi_l|, \tag{3}$$

where  $Q_j^m$  is the set of the indexes of the  $m \times m$  adjacent pixels of the  $j$ -th lattice pixel and  $g_{l,j}$  is the corresponding element of a matrix representing a 2D  $m \times m$  Gaussian matrix. The  $g_{l,j}$  scalar weights the differences according to the distance of the two adjacent,  $l$ -th and  $j$ -th pixels. Using this regularization term we can force the optimization algorithm to find binary matrices with possibly compact regions of 0s and 1s.

For solving (2) the *simulated annealing* (SA) optimization method [10] was used.

### 3.2 Convex-Concave Regularization and DC-Programming

We also consider the one-parameter family of functionals introduced in [7]:

$$J_\mu(x) := \|Ax - b\|^2 + \frac{\alpha}{2} \sum_{j=1}^n \sum_{l \in Q_j^1} (x_j - x_l)^2 - \mu \frac{1}{2} \langle x, x - e \rangle, \quad x \in [0, 1]^n. \tag{4}$$

The first terms in (4) and (2) coincide. The second term in (4) is similar to (3), but involves nearest neighbors only, i.e.  $m = 1$ , with uniform weighting. This term is controlled by the regularization parameter  $\alpha$ . Proper values depend

on the application and have to be supplied by the user. The third term in (4), together with the convex domain of definition  $x \in [0, 1]^n$ , pertains to the second optimization strategy, to be explained below, that was used to minimize (4). It is a concave functional which gradually enforces the binary constraint  $x \in \{0, 1\}^n$  by increasing the value of  $\mu$  ( $e$  denotes the vector with all components equal to 1).

---

**Algorithm 1.** SA Algorithm

---

**Require:**  $\gamma \geq 0$  {regularization parameter supplied by the user}  
**Require:**  $T_{\text{start}} > 0$  {start temperature supplied by the user}  
**Require:**  $T_{\text{min}} > 0$  {minimum temperature supplied by the user}  
**Require:**  $1 > T_{\text{factor}} > 0$  {The multiplicative constant for reducing the temperature supplied by the user}  
**Require:**  $1 > R_{\text{objective}} > 0$  {The ratio between the first and the current value of the objective function supplied by the user}

$x := (0, \dots, 0)^\top$   
 $T := T_{\text{start}}$   
 $C_{\text{start}} := C_{\text{old}} := \|Ax - b\|^2 + \gamma \cdot \Phi(x)$

**repeat**  
    **for**  $i = 0$  to  $\text{sizeof}(x)$  **do**  
        choose a random position  $j$  in the vector  $x$   
         $\tilde{x} := x$   
         $\tilde{x}[j] := 1 - x[j]$  {change the value of  $x$  in the position  $j$ }  
         $C_{\text{new}} := \|A\tilde{x} - b\|^2 + \gamma \cdot \Phi(\tilde{x})$   
         $z := \text{random}()$   
         $\Delta C := C_{\text{new}} - C_{\text{old}}$   
        **if**  $\Delta C < 0$  or  $\exp(-\Delta C/T) > z$ , **then**  
             $x := \tilde{x}$  {accept changes}  
             $C_{\text{old}} := C_{\text{new}}$   
        **end if**  
    **end for**  
     $T := T * T_{\text{factor}}$   
**until**  $T > T_{\text{min}}$  or  $C_{\text{old}}/C_{\text{start}} > R_{\text{objective}}$

---

Functional (4) can be represented by the sum of a convex and a concave function

$$J_\mu(x) = g(x) - h_\mu(x) , \quad x \in [0, 1]^n , \tag{5}$$

where

$$g(x) := \|Ax - b\|^2 + \frac{\alpha}{2} \sum_i^n \sum_{j \in Q_i^n} (x_i - x_j)^2 , \tag{6}$$

$$=: \|Ax - b\|^2 + \alpha \langle x, L^\top Lx \rangle , \tag{7}$$

$$h_\mu(x) := \mu \frac{1}{2} \langle x, x - e \rangle . \tag{8}$$

As a consequence, (4) naturally belongs to the class of dc-programs (dc: difference of convex functions) and thus provides a basis for algorithm design. It is shown in [7] that the following algorithm converges to a *binary* local minimum of the criterion  $J_\mu$ :

---

**Algorithm 2.** DC Algorithm
 

---

**Require:**  $\alpha \geq 0$  {regularization parameter supplied by the user}  
**Require:**  $\epsilon_{in} > 0$  {termination criterion for the inner loop}  
**Require:**  $\epsilon_{out} > 0$  {termination criterion for the outer loop}  
**Require:**  $\epsilon_\mu > 0$  {determines the increment  $\mu_\Delta$  by eqn. (9)}  
 $x := (0, \dots, 0)^\top$   
 $\mu = 0$   
**repeat**  
  **repeat**  
     $\tilde{x} := x$   
     $x := \operatorname{argmin}_{x \in [0,1]^n} \{g(x) - \langle x, \nabla h_\mu(\tilde{x}) \rangle\}$   
  **until**  $\|x - \tilde{x}\|_2 < \epsilon_{in}$   
   $\mu := \mu + \mu_\Delta$   
**until**  $\max_i \{\min\{x_i, 1 - x_i\}\} < \epsilon_{out}$

---

We point out that each  $x$  computed in the inner loop is the global optimum of a *convex* optimization problem. Our current implementation involves [9] for this step, but many other convex optimization techniques could be applied as well.

Furthermore, while the decomposition (5) with (7) and (8) is the most natural one, a range of alternative decompositions of the functional  $J_\mu$  are possible to which algorithm 1 can be applied. We refer to [8] for further details.

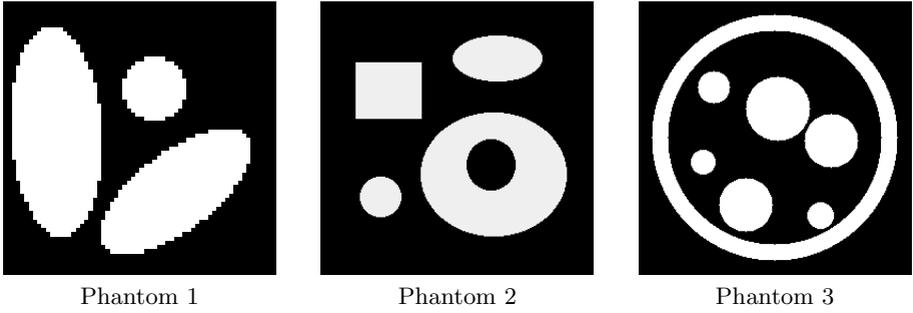
## 4 Evaluation

### 4.1 Data Sets

For evaluation purposes both reconstruction algorithms were tested on the same data set of binary images. The images are software phantoms consisting of discretized versions of geometrical objects like circles, ellipses, etc. – see Fig. 2.

For each phantom, the image reconstruction problem (1) was compiled by taking parallel projections from different directions. The number  $p$  of projections ranged between 2, 3, 5, and 6. For  $p \in \{2, 3, 5\}$ , directions were uniformly chosen within  $[0^\circ, 90^\circ]$ , and within  $[0^\circ, 150^\circ]$  for  $p = 6$ . For each direction, the number of measurements was 96 for phantom 1 and 384 for phantom 2 and 3.

In addition to noiseless projection data, we also used noisy data for the evaluation. To this end, the projection data were superimposed by noise with Gaussian distribution  $\mathcal{N}(0, \sigma)$ ,  $\sigma \in \{0.5, 1.5, 5\}$ . Negative values that may rarely be generated in this way, do not make sense physically and were clipped to the value zero.



**Fig. 2.** Phantom images of size  $64 \times 64$ ,  $256 \times 256$ , and  $256 \times 256$  used for the experimental evaluation

## 4.2 Performance Measures

Let  $x^*$  be the ground truth image and  $x$  be a solution to the reconstruction problem (1) computed by either optimization algorithm. We use the following error measures for a quantitative evaluation:

$$E_1(x) := \|Ax - b\|_2 ,$$

$$E_2(x) := \frac{1}{\sum_{i=1}^n x_i} \|x - x^*\|_1 .$$

For interpreting the corresponding numerical results in the tables below, readers should keep in mind that these two measures scale quite differently. While a single pixel error results in a change of  $E_1$  of about  $10^1$ , say, the order of change of  $E_2$  will be  $10^{-2}$  only.

## 4.3 Parameter Settings

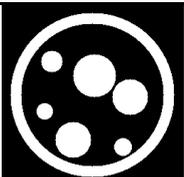
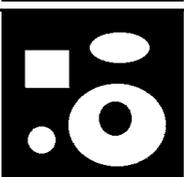
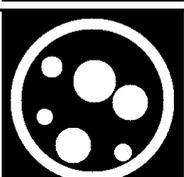
To compare both approaches numerically, we used a fixed parameter set for each reconstruction algorithm. These values were used throughout all experiments.

### Simulated Annealing Algorithm:

$$\begin{aligned} \gamma &= 14.0 \\ T_{\text{start}} &= 4.0 \\ T_{\text{min}} &= 10^{-14} \\ T_{\text{factor}} &= 0.97 \\ R_{\text{objective}} &= 0.00001 \end{aligned}$$

### DC Algorithm:

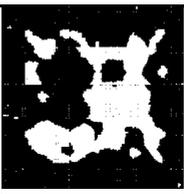
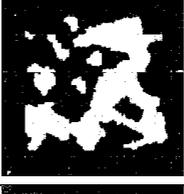
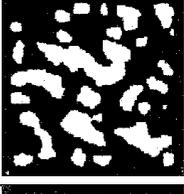
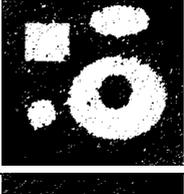
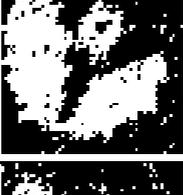
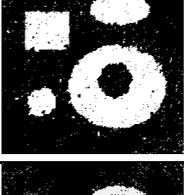
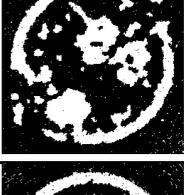
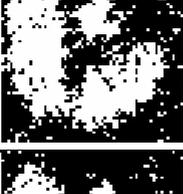
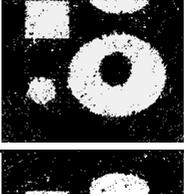
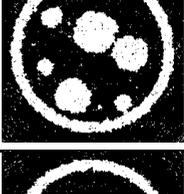
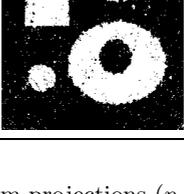
$$\begin{aligned} \alpha &= 0.25 \\ \epsilon_{in} &= 0.1 \\ \epsilon_{out} &= 0.01 \\ \epsilon_{\mu} &= 10 \end{aligned}$$

Projections	Algorithm	Phantom 1	Phantom 2	Phantom 3
2	DC			
	SA			
5	DC			
	SA			
6	DC			
	SA			

**Fig. 3.** The phantom images reconstructed from noise free projections ( $p = 2, 5, 6$ )

The  $\mu$ -increment was computed by evaluating the following equation

$$\mu_{\Delta} := \frac{\epsilon_{\mu} n^{1/2} \lambda_{\min}(Q)}{\|x - \frac{1}{2}e\|}, \quad Q := A^{\top}A + \alpha L^{\top}L. \tag{9}$$

Projections	Algorithm	Phantom 1	Phantom 2	Phantom 3
2	DC			
	SA			
5	DC			
	SA			
6	DC			
	SA			

**Fig. 4.** The phantom images reconstructed from projections ( $p = 2, 5, 6$ ) with additive 5 % noise

Here,  $x$  denotes the solution of the very first inner loop for  $\mu = 0$ , and  $\lambda_{\min}$  is the smallest eigen value of the matrix  $Q$  that can be computed offline and beforehand. For details and an interpretation of (9), we refer to [8].

## 5 Results

The aim of the experiments was to make a comparison between the two reconstruction methods. Both methods used the same input projections. Also the same formulas were applied for measuring the errors.

We computed the reconstruction images for phantoms 1-3 from 2, 3, 5, and 6 projections with and without additive noise. Figure 3 shows the reconstruction results in the case of noise-free projections (the reconstructions from 3 projections are omitted, because of the lack of space). From all images reconstructed from noisy projections we present here only those having 5 % additive noise (see Fig. 4). The cases of 0.5 % and 1.5 % additive errors show something similar behavior).

The tables 1-3 contain the error values of the measures  $E_1(x)$  and  $E_2(x)$  for all reconstruction scenarios and for both algorithms. Although the same experiments were repeated with 0 %, 0.5 %, 1.5 %, and 5 % additive noise, we present here all the tables except the case 0.5 % (which gave similar results as in the case of 0 %).

**Table 1.** The error values  $E_1(x)/E_2(x)$  measured on the reconstructed images in noise free case

Projections	Algorithm	Phantom 1	Phantom 2	Phantom 3
2	DC	3.464/0.537	6.782/0.835	5.477/1.108
	SA	8.173/0.480	15.870/0.841	16.901/1.198
3	DC	0.000/0.000	8.351/0.471	7.804/0.751
	SA	6.779/0.020	19.028/0.524	20.453/0.882
5	DC	0.000/0.000	0.005/0.000	14.761/0.545
	SA	0.000/0.000	9.040/0.001	26.478/0.537
6	DC	0.000/0.000	0.005/0.000	0.004/0.000
	SA	0.000/0.000	10.134/0.001	9.632/0.001

**Table 2.** The error values  $E_1(x)/E_2(x)$  measured on the reconstructed images in the case of 1.5 % additive noise

Projections	Algorithm	Phantom 1	Phantom 2	Phantom 3
2	DC	9.708/0.492	21.375/0.829	21.101/1.181
	SA	12.707/0.442	26.854/0.853	26.391/1.188
3	DC	11.892/0.080	24.024/0.489	23.414/0.761
	SA	15.993/0.093	31.156/0.565	30.878/0.918
5	DC	19.020/0.080	31.135/0.026	29.182/0.551
	SA	23.323/0.059	41.052/0.021	39.057/0.536
6	DC	18.795/0.102	31.298/0.034	33.203/0.045
	SA	25.324/0.058	45.537/0.020	43.371/0.042

**Table 3.** The error values  $E_1(x)/E_2(x)$  measured on the reconstructed images in the case of 5 % additive noise

Projections	Algorithm	Phantom 1	Phantom 2	Phantom 3
2	DC	25.599/0.479	59.701/0.857	59.270/1.160
	SA	28.633/0.519	63.004/0.839	62.437/1.162
3	DC	27.276/0.305	68.470/0.525	68.802/0.798
	SA	31.936/0.295	73.966/0.563	73.522/0.903
5	DC	44.607/0.292	82.858/0.114	80.945/0.589
	SA	48.423/0.265	91.049/0.103	87.514/0.597
6	DC	47.855/0.342	86.354/0.123	86.269/0.151
	SA	53.585/0.287	98.575/0.102	95.214/0.145

## 6 Discussion

Both algorithm perform very similar on the tested reconstruction problems. Consider first the noise free reconstructions. The methods were able to reconstruct Phantom 1 from 3 or more projections. Phantom 2 was more difficult, 5 or more projections are necessary for the almost perfect reconstruction. The most difficult object was Phantom 3, it needs 6 projections for a good quality reconstruction.

The DC method gives smaller errors in almost all cases in Table 1. It is interesting that the measure  $E_1(x)$  was smaller for DC than SA in every cases. The reason can be explained as follows.  $E_1(x)$  measures the differences between the input projections ( $b$ ) and the projections of the reconstructed object ( $Ax$ ). For this reason  $E_1(x)$  takes into account only the projections and not the original object. (That is,  $E_1(x)$  can be very small even if the object  $x$  is far from the original one.) Our results shows that the difference between the projections is not so strongly weighted in the objective function of SA (2). At the same time SA reaches similarly low values of  $E_2(x)$  as DC does.

Consider now the results of noisy projections. It is clear that DC gives again better  $E_1(x)$  values. The differences in the  $E_2(x)$  values are small, if we have 5 or more projections then SA seems to give solutions being nearer to the original object.

## 7 Conclusion

Summarizing the results we can say that there is no huge difference between the qualities of the reconstructed images of the two methods.

## Acknowledgments

The CVGPR-group gratefully acknowledges support by Siemens Medical Solutions, Forchheim, Germany. The fifth author was supported during his stay in Mannheim by the Alexander von Humboldt-Foundation. This work was partially supported by the NSF Grant DMS 0306215 and the OTKA Grant T048476.

## References

1. Herman, G.T., Kuba, A., eds.: *Discrete Tomography: Foundations, Algorithms, and Applications*. Birkhäuser, Boston (1999)
2. Herman, G.T., Kuba, A., eds.: *Advances in Discrete Tomography and Its Applications*. Birkhäuser, Boston (2006). To appear
3. S. Krimmel, J. Baumann, Z. Kiss, A. Kuba, A. Nagy, J. Stephan: Discrete tomography for reconstruction from limited view angles in non-destructive testing. *Electronic Notes in Discrete Mathematics* **20** (2005) 455–474
4. G. T. Herman, A. Kuba: Discrete tomography in medical imaging. *Proc. of the IEEE* **91** (2003) 1612–1626
5. Liao, H.Y., Herman, G.T.: A method for reconstructing label images from a few projections, as motivated by electron microscopy. *Annals of Operations Research*. To appear
6. Weber, S., Schüle, T., Schnörr, C., Hornegger, J.: A linear programming approach to limited angle 3d reconstruction from dsa projections. *Special Issue of Methods of Information in Medicine* **4** (2004) 320–326
7. Schüle, T., Schnörr, C., Weber, S., Hornegger, J.: Discrete tomography by convex-concave regularization and D.C. programming. *Discrete Applied Mathematics* **151** (2005) 229–243
8. Schnörr, C., Schüle, T., Weber, S.: Variational Reconstruction with DC-Programming. In Herman, G.T., Kuba, A., eds.: *Advances in Discrete Tomography and Its Applications*. Birkhäuser, Boston (2006). To appear
9. Birgin, E.G., Martínez, J.M., Raydan, M.: Algorithm 813: SPG - software for convex-constrained optimization. *ACM Transactions on Mathematical Software* **27** (2001) 340–349
10. Metropolis, N., A. Rosenbluth, A. T. Rosenbluth, M., Teller, E.: Equation of state calculation by fast computing machines. *J. Chem. Phys.* **21** (1953), 1087–1092.