SIAM J. Optim. © 2000 Society for Industrial and Applied Mathematics
Vol. 11, No. 2, pp. 522–546

APPROXIMATING BINARY IMAGES FROM DISCRETE X-RAYS[*]

PETER GRITZMANN[†], SVEN DE VRIES[†], AND MARKUS WIEGELMANN[†]

Abstract. We study the problem of approximating binary images that are accessible only through few evaluations of their discrete X-ray transform, i.e., through their projections counted with multiplicity along some lines. This inverse discrete problem belongs to a class of generalized set partitioning problems and allows natural packing and covering relaxations. For these (ℕℙ-hard) optimization problems we present various approximation algorithms and provide estimates for their worst-case performance. Further, we report on computational results for various variants of these algorithms. In particular, the corresponding integer programs are solved with only small absolute error for instances up to 250,000 binary variables.

Key words. discrete tomography, set packing, set covering, set partitioning, greedy algorithm, matching, approximation algorithm, worst-case performance, polynomial time

AMS subject classifications. 05B40, 05C35, 05C70, 68R05, 68U10, 82D25, 90C27, 92C55

PII. S105262349935726X

1. Introduction.

The present paper studies various algorithms for finding approximate solutions for an inverse discrete problem that is most prominently motivated by the demand in material science for developing a tool for the reconstruction of 3-dimensional crystalline structures that are accessible only through some images provided by high resolution transmission electron microscopy. In fact, the articles [13] and [18] describe a new technique called *QUANTITEM* for the quantitative analysis of the information provided by transmission electron microscopy that can effectively measure the number of atoms lying on each line parallel to a given set of directions.

Mathematically, this is the inverse problem of reconstructing certain discrete density functions from their discrete X-rays in certain directions. More precisely, the basic question is the following. Can a finite set of points in the integer lattice $\mathbb{Z}^3$ be (approximately) reconstructed from measurements of the number of its points lying on each line parallel to one of a small prescribed number of directions specified by nonzero vectors in $\mathbb{Z}^3$? Here small means 3, 4, or 5 since the energy that is needed to produce the images is about 200 keV so that after a few exposures the object is damaged.

Various approaches have been suggested for solving the general reconstruction problem, and various theoretical results are available; see, e.g., [9] for a survey. In the present paper we concentrate on approximative solutions. Even though most of the resulting combinatorial optimization problems are ℕℙ-hard, we prove in section 3 that some (relatively) simple algorithms yield already very good worst-case bounds. As section 4 indicates, these algorithms perform even better in computational practice.

Let us close the introduction with a word of warning. Typically, when one is dealing with optimization problems in practice it is completely satisfactory to produce solutions that are close to optimal. For instance, a tour for a given instance of the traveling salesman problem that is off by only a few percents is for many practical purposes almost as good as an optimal tour. This is because the particular optimization

[*]Received by the editors May 25, 1999; accepted for publication (in revised form) May 31, 2000; published electronically November 2, 2000.
http://www.siam.org/journals/siopt/11-2/35726.html

[†]Zentrum Mathematik, Technische Universität München, D-80290 München, Germany (gritzman@mathematik.tu-muenchen.de, devries@mathematik.tu-muenchen.de, markus.wiegelmann@ db.com). The first and second authors were supported in part by German Federal Ministry of Education, Science, Research and Technology grant 03-GR7TM1.

is typically just part of a much more complex real world task, and the improvement over existing methods is governed by so many much harder to influence factors that a small error in the optimization step does not really matter by any practical means. This is different in the context of our prime application. The relevant measure for the quality of an approximation to a binary image would of course be the deviation from this image. Hence, in order to devise the most appropriate objective function one would have to know the underlying solution of the given inverse problem. However, the whole point is of course to find this unknown solution. Hence, one can only consider objective functions, with respect to which the approximation is evaluated, that are based on the given input data. While a good approximation in this sense is close to a solution in that its X-ray images in the given directions are close to those of the original set, the approximating set itself may be off quite substantially. In fact, the inverse discrete problem is ill-posed and it is precisely this property that causes additional difficulties. In particular, if the input data do not uniquely determine the image, even a "perfect" solution that is completely consistent with all given data may be quite different from the unknown real object.

Obviously there is more work to be done to handle the ill-posedness of the problem in practice. Hence, the results of this paper should be regarded only as a first (yet reassuring!) step in providing a computational tool that is adequate for the real world applications outlined above. In particular, our approximate algorithms can be used to provide lower bounds in branch-and-cut approaches that incorporate strategies to handle the nonuniqueness of solutions and the presence of noise in the data.

The paper is organized as follows. Section 2 provides the basic notation, states the problems and algorithmic paradigms that are most important in the context of the present paper, and gives a brief overview of our main results. Section 3 studies various polynomial-time iterative improvement strategies for inner and outer approximation. We derive performance ratios that show that in this model the optimum can be approximated up to a relative error that depends only on the number m of directions in which the X-ray data are available. The analysis is based on work of Hurkens and Schrijver [12], Goldschmidt, Hochbaum, and Yu [8], and Halldórsson [11] for set packing and set covering heuristics. Our theoretical worst-case bounds are complemented by extremely satisfactory computational results described in section 4.

2. Preliminaries and results.

2.1. Basics of discrete tomography. We use the general setting of a d-dimensional Euclidean space $\mathbb{E}^d$, with $d \geq 2$, though only the cases $d = 2, 3$ are relevant in practice. Let $\mathcal{S}_{1,d}$ be the set of all 1-dimensional subspaces in $\mathbb{E}^d$, and let $\mathcal{F}^d$ denote the family of finite subsets of $\mathbb{Z}^d$. For $F \in \mathcal{F}^d$ let $|F|$ be the cardinality of F. A vector $v \in \mathbb{Z}^d \setminus \{0\}$ is called a *lattice direction*; $\mathcal{L}_{1,d}$ denotes the subset of $\mathcal{S}_{1,d}$ spanned by a lattice direction. For $S \in \mathcal{S}_{1,d}$ let $\mathcal{A}(S)$ denote the family of all lines parallel to S. The *(discrete)* 1*-dimensional X-ray parallel to* S of a set $F \in \mathcal{F}^d$ is the function $X_S F : \mathcal{A}(S) \to \mathbb{N}_0 = \mathbb{N} \cup \{0\}$ defined by

$$X_S F(T) = |F \cap T| \qquad \text{for } T \in \mathcal{A}(S).$$

Since F is finite, the X-ray $X_S F$ has finite support $\mathcal{T} \subset \mathcal{A}(S)$.

In the inverse reconstruction problem, we are given *data functions* $\phi_i : \mathcal{A}(S_i) \to \mathbb{N}_0$, $i = 1, \dots, m$, with finite support, and we want to find a set $F \subset \mathbb{Z}^d$ with corresponding X-rays. More formally, for $S_1, \dots, S_m \in \mathcal{L}_{1,d}$ pairwise different, the most important algorithmic task in our context can be stated as follows.

Reconstruction($S_1, \dots, S_m$).

Given data functions $\phi_i : \mathcal{A}(S_i) \to \mathbb{N}_0$ for $i = 1, \dots, m$, find a finite set $F \subset \mathbb{Z}^d$ such that $\phi_i = X_{S_i} F$ for all $i = 1, \dots, m$ or decide that no such F exists.

Clearly, when investigating the computational complexity of the above problem in the usual *binary Turing machine model* one has to describe suitable finite data structures. We do not go into such details here but refer the reader to [5]. For the purpose of this paper, handling an input of m data functions $\phi_1, \dots, \phi_m$ with supports $\mathcal{T}_1, \dots, \mathcal{T}_m$, respectively, is facilitated with the aid of a set $G \subset \mathbb{Z}^d$ of candidate points. This set G consists of the intersection of all (finitely many) translates of $\bigcap_{i=1}^m S_i$ that arise as the intersection of m lines parallel to $S_1, \dots, S_m$ with $\mathbb{Z}^d$, respectively, whose data function value is nonzero, i.e.,

$$G = \mathbb{Z}^d \cap \bigcap_{i=1}^{m} \bigcup_{T \in \mathcal{T}_i} T.$$

To exclude trivial cases, in the following we will always assume that $G \neq \emptyset$ and that $\bigcap_{i=1}^m S_i = \{0\}$. Hence, in particular $m \geq 2$.

The incidences of G and $\mathcal{T}_i$ can be encoded by an incidence matrix A_i. To fix the notation, let G consist of, say, N points, and let $M_i = |\mathcal{T}_i|$ and $M = M_1 + \cdots + M_m$. Then the incidence matrices $A_i \in \{0,1\}^{M_i \times N}$ can be joined together to form a matrix $A \in \{0,1\}^{M \times N}$. Identifying a subset of G with its characteristic vector $x \in \{0,1\}^N$, the reconstruction problem amounts to solving the integer linear feasibility program

$$Ax = b \quad \text{s.t. } x \in \{0,1\}^N, \tag{1}$$

where $b^T = (b_1^T, \dots, b_m^T)$ contains the corresponding values of the data functions $\phi_1, \dots, \phi_m$ as the right-hand sides of $A_1, \dots, A_m$, respectively.

Let us point out here in passing that more general inverse discrete problems can be modeled in a similar way. In fact, query sets (which are lines in the present paper) could be chosen in various different and meaningful ways. (For instance, if the lines are replaced by the translates of some k-dimensional subspaces, we obtain the reconstruction problem for discrete *k-dimensional X-rays*.)

It is not difficult to see that the matrix A is totally unimodular when $m = 2$. In particular, for $m = 2$ the integer linear program (1) and its linear programming relaxation

$$Ax = b \quad \text{s.t. } x \in [0,1]^N, \tag{2}$$

where the condition $x \in \{0,1\}^N$ is replaced by the weaker constraint $x \in [0,1]^N$, are equivalent in the sense that all vertices of the polytope $\{x\colon Ax = b \wedge x \in [0,1]^N\}$ are 0-1 vectors anyway; see, e.g., [17] for an exposition of the underlying theory. Hence, for $m = 2$ the reconstruction problem is solvable in polynomial time; see [15], [1], and [7] for different proofs that do not rely on the fact that linear programming problems can be solved in polynomial time. Reconstruction($S_1, \dots, S_m$) becomes $\mathbb{NP}$-hard, however, when $m \geq 3$; see [5]. (For an introduction to the theory of computational complexity see [6].) This means that (unless $\mathbb{P} = \mathbb{NP}$) exact solutions of (1) require (in general) a superpolynomial amount of time. In polynomial time only approximate solutions can be expected. We will henceforth always assume that $m \geq 3$.

Let us stress the fact that while the solutions of the polynomial-time solvable LP-relaxation (2) do provide some information about (1) (see [3]), it is our goal to solve (1) rather than (2), since the objects underlying our prime application are crystalline structures forming (physical) sets of atoms rather than "fuzzy" sets; see [10] for some additional discussion of this point.

2.2. Two optimization problems. For measuring the quality of approximation methods, we introduce objective functions so as to formulate the reconstruction problem as optimization problems. Two very natural such formulations are the following problems, Best-Inner-Fit and Best-Outer-Fit.

Best-Inner-Fit$(S_1,\dots,S_m)$ [BIF].
Given data functions $\phi_1,\dots,\phi_m$, find a set $F\subset G$ of maximal cardinality such that

$$X_{S_i}F(T)\le \phi_i(T) \text{ for all } T\in\mathcal{T}_i \text{ and } i=1,\dots,m.$$

Equivalently, [BIF] can be formulated as the integer linear program

$$\begin{aligned} &\max\ \mathbb{1}^T x\ \text{ s.t.}\\ &Ax\le b\ \text{ and } x\in\{0,1\}^N, \end{aligned}\tag{3}$$

where $\mathbb{1}$ is the all-ones vector.

The "outer counterpart" of this inner approximation is defined as follows.

Best-Outer-Fit$(S_1,\dots,S_m)$ [BOF].
Given data functions $\phi_1,\dots,\phi_m$, find a set $F\subset G$ of minimal cardinality such that

$$X_{S_i}F(T)\ge \phi_i(T) \text{ for all } T\in\mathcal{T}_i \text{ and } i=1,\dots,m.$$

Again, the problem is equivalent to an integer linear program, precisely to

$$\begin{aligned} &\min\ \mathbb{1}^T x\ \text{ s.t.}\\ &Ax\ge b\ \text{ and } x\in\{0,1\}^N. \end{aligned}\tag{4}$$

Note that while for any given instance of [BIF] $\emptyset$ is a feasible solution, [BOF] may be infeasible. In order to exclude this degeneracy, we will in the following always assume that

$$\phi_i\le X_{S_i}G \text{ for } i=1,\dots,m.$$

The two problems [BIF] and [BOF] are then complementary to each other in the following sense. The complement $\bar F=G\setminus F$ of a solution $F\subset G$ of an instance of one problem is a solution of the instance with complementary data functions $\bar\phi_i$ defined by $\bar\phi_i(T)=|G\cap T|-\phi_i(T)$ of the other problem. This reflects the fact that reconstructing the "positive" or the "negative" of a binary picture are equivalent. However, as the direct conversion of an approximation result for [BIF] of the form $|V|/|F|\ge\alpha$ (F is an optimal solution and V is some solution) yields a bound $|\bar V|/|\bar F|\le\alpha+(1-\alpha)|G|/|\bar F|$ for [BOF] that is dependent on the "density" $|F|/|G|$ of an optimal solution in the underlying candidate grid, bounds for the relative error of one problem are usually not "identical" to bounds for the other. More importantly, our algorithms for [BOF] are actually insertion methods rather than "dual" deletion methods. Hence, we will consider [BIF] and [BOF] separately in section 3.

Let us remark in passing that one can of course consider other kinds of optimization problems related to Reconstruction$(S_1, \dots, S_m)$. For instance, rather than measuring the approximability in terms of the points inserted into the candidate grid one may count the number of lines on which an X-ray of a solution coincides with the given value of the corresponding data function. An intractability result for this kind of approximation can be found in [10].

2.3. The basic algorithmic paradigm.

In this section we describe a general algorithmic scheme for solving [BIF] and [BOF] that provides the framework for the subsequent approximation algorithms studied in sections 3 and 4. See [10] for a discussion of some other algorithmic paradigms that comprise most of the methods for solving Reconstruction$(S_1, \dots, S_m)$ that have been suggested by various authors in the past.

In the present paper we give a theoretical and computational analysis of various iterative improvement strategies that are built on some greedy method. In the simplest classes of local search algorithms for [BIF] and [BOF] the neighborhood of a set S is defined as the collection of all supersets of S of cardinality $|S| + 1$ or of all subsets of cardinality $|S| - 1$, respectively, and the choice is based on some greedy strategy (that may or may not use weights for breaking ties).

In order to increase the performance of such iterative insertion or deletion algorithms, one can apply *r-improvements* for $r \in \mathbb{N}_0$, where an r-point $\langle(r+1)$-point$\rangle$ subset of a current feasible solution $F \subset G$ for the given instance of [BIF] $\langle$[BOF]$\rangle$ is deleted and $r+1$ $\langle r \rangle$ points of $(G \setminus F)$ are inserted while maintaining feasibility. A feasible set $F \subset G$ is called *t-optimal* for the given instance of [BIF] $\langle$[BOF]$\rangle$ if no r-improvement is possible for any $r \leq t$. Note that 0-optimality agrees with the common greedy-optimality (no point can be inserted without destroying feasibility for [BIF] and no point can be removed without destroying feasibility for [BOF]). However, since our algorithms for [BOF] are based on greedy-type *insertions* rather than greedy-*deletions*, the greedy algorithm of section 3.3 need not produce 0-optimal solutions per se.

The following paradigm comprises a large class of iterative improvement methods for [BIF]. A similar paradigm can be formulated for [BOF]. (A symbolic formulation in the realm of commutative algebra of a general reduction process involving a set of binomials in an appropriate toric ideal is given in [20].)

Paradigm 2.1 (iterative inner approximation).

- INPUT: *Data functions $\phi_1, \dots, \phi_m$ for the given lines $S_1, \dots, S_m$, respectively.*
- OUTPUT: *A feasible set $F \subset G$ for the given instance of [BIF].*
- COMPUTATION:

  Start with $F = \emptyset$ and successively apply r-improvements for $r \leq t$ for some fixed constant $t \in \mathbb{N}_0$ until no further improvement is possible.

Since it is not specified how to select the points for insertion and deletion, Paradigm 2.1 is so general and flexible that it covers a large number of algorithms that incorporate promising refinements. For example, the X-ray data can be used for back-projection-like techniques to express preferences between points to be chosen; see Algorithm 3.7. In addition, connectivity of the solution (in a sense that is justified by the physical structure of the analyzed material) can be rewarded by introducing adjustable weights. Similarly, information from neighboring layers can be taken into account in a layerwise reconstruction of a 3-dimensional object. In fact, the positive results of section 3 will apply to the general paradigm.

2.4. Main results. The simplest algorithm for [BIF] within the framework of Paradigm 2.1 is the plain greedy algorithm which considers the positions of the grid in an arbitrary order and successively fills in points. We will refer to it as GreedyA. GreedyB and GreedyC will be variants with refined insertion order. As a first result (Theorem 3.1, with $t = 0$), we see that

$$|V|/|F| \geq 1/m,$$

where V is the set obtained by GreedyA (or GreedyB or GreedyC), and F is an optimal solution. Recall that m is the number of directions, whence the sharp and (considering that it is hard to think of any algorithm that is simpler than GreedyA) surprisingly good lower bound $1/m$ reflects the fact that the more data are given, the harder it is for a greedy strategy to satisfy them. In our experiments, it turns out that $|V|/|F|$ is typically greater than 0.9 and for large instances greater than 0.96 even for $m = 5$; see section 4.

There are two natural ways to improve this algorithm:

(a) using a better order to visit the candidate points and

(b) using 1-improvements, 2-improvements, etc.

In terms of (a) we use a strategy (GreedyB) that is motivated by a method of [16] for solving consistent [BIF]-instances exactly for $m = 2$. In our computational study GreedyB clearly outperforms GreedyA for all m considered; see Figures 9 and 10.

In GreedyC weights are assigned dynamically to the candidate points to represent the "changing importance" of a point to be included in a solution. Our computational study shows that GreedyC gives smaller relative errors than GreedyA and GreedyB; see Figure 9. In fact, even the *absolute errors* are small; the average case for GreedyC for 250,000 positions and density 50%, i.e., solutions of cardinality 125,000 being 21.62, 64.13, 111.88 missing atoms for 3, 4, 5 directions, respectively. The price to pay for this excellent performance is GreedyC's considerably longer running time; see section 4.

In terms of (b), Theorem 3.1 shows that, for a t-optimal solution V,

$$\frac{|V|}{|F|} \geq \frac{2}{m} - \epsilon_m(t),$$

where $\epsilon_m(t)$ is given explicitly and approaches 0 exponentially fast. Computationally, it turns out that performing 1-improvements after GreedyA, GreedyB, or GreedyC typically yields substantial improvements. In fact, in our computational study the *absolute* errors go down for ImprovementC to 1.07, 23.28, 64.58 for 3, 4, 5 directions, respectively; see Figure 10.

Theorem 3.9 provides worst-case guarantees for [BOF]. Part (a) shows that a simple greedy-type insertion algorithm yields a solution U such that

$$|U|/|F| \leq H(m),$$

where $H(m) = 1 + 1/2 + \cdots + 1/m$ is the mth harmonic number. If additional matching techniques are applied to obtain a stronger optimality condition ("*matching-optimality*"), then

$$|U|/|F| \leq H(m) - 1/6;$$

see Theorem 3.9(b).

Theorem 3.10(a) shows that the t-optimality of a solution U guarantees that

$$|U|/|F| \leq m/2 + \epsilon_m(t),$$

where again $\epsilon_m(t)$ is given explicitly and tends to 0 exponentially fast. If, finally, the solution is matching-optimal and (what will be defined later) *effect-3-t-optimal* for $t \geq 5$, then

$$|U|/|F| \leq H(m) - 1/3;$$

see Theorem 3.10(c). That is, for $m = 3, 4, 5$ the bounds are $\frac{3}{2}$, $\frac{7}{4}$, and $\frac{39}{20}$.

Note that in the case of single coverings, there is a slightly better bound for a certain semi-local search algorithm due to Duh and Fürer [2]. If their approach could be extended to [BOF] it would read $|U|/|F| \leq H(m) - 1/2$. However, currently it is not known whether such an extension is possible.

Let us close this section with two remarks. First, all our results are formulated within the realm of discrete tomography due to its main objective. It goes without saying that the theoretical performance ratios apply also to more general multiple packing and multiple covering problems. Second, as already pointed out in the introduction, our analysis makes substantial use of ideas of Hurkens and Schrijver [12], Goldschmidt, Hochbaum, and Yu [8], and Halldórsson [11] for set packing and set covering heuristics. There may be a way to axiomize how to extend results for simple packings and coverings to more general settings including our discrete tomography to evoke some of their results directly. In general, however, multiple packing and multiple covering appear to be harder: general reductions to single packing or covering problems are not known and not likely to exist. For this reason (and as a service to the reader) we give a full direct analysis of each of the considered algorithms.

3. Worst-case performance guarantees for iterative improvement algorithms.

3.1. Effects. Let $V \subset G$ and $g \in G \setminus V$. The *effect* $e_V(g)$ of g with respect to V is the number of lines $g + S_i$ through g for which the X-ray bound is not yet achieved by V, i.e.,

$$e_V(g) = |\{i \in \{1, \ldots, m\} : X_{S_i} V(g + S_i) < \phi_i(g + S_i)\}|.$$

Clearly, the effect of a point is an integer between 0 and m. The notion can easily be extended to subsets of $G \setminus V$. More precisely, let $V' \subset G \setminus V$; then the *effect* $e_V(V')$ of V' with respect to V is defined by

$$e_V(V') = \sum_{i=1}^{m} \sum_{T \in \mathcal{T}_i} e_{V,V',i}(T),$$

where

$$e_{V,V',i}(T) = \begin{cases} |V' \cap T| & \text{if } |(V \cup V') \cap T| \leq \phi_i(T); \\ \phi_i(T) - |V \cap T| & \text{if } |V \cap T| < \phi_i(T) \text{ and } |(V \cup V') \cap T| \geq \phi_i(T); \\ 0 & \text{if } |V \cap T| \geq \phi_i(T). \end{cases}$$

Clearly, $e_V(g) = e_V(\{g\})$; also $e_V(V')$ lends itself to a successive evaluation. In fact, if $V' = \{g_1, \dots, g_l\}$,

$$e_V(V') = \sum_{i=1}^{l} e_{V\cup\{g_1,\dots,g_{i-1}\}}(g_i).$$

Furthermore,

$$e = \sum_{i=1}^{m} \sum_{T\in\mathcal{T}_i} \phi_i(T)$$

is called the *total effect* of the given instance. Clearly, if L and U are feasible for the given instance of [BIF] and [BOF], respectively, then $m|L| \le e \le m|U|$. In particular, if F is an exact solution of RECONSTRUCTION$(S_1, \dots, S_m)$, then $e = m|F|$.

3.2. Inner approximation algorithms.

The following result gives worst-case performance guarantees for a wide class of primal algorithms for [BIF] that fit into Paradigm 2.1. In particular, all algorithms of section 4 are covered.

THEOREM 3.1. *Let $t \in \mathbb{N}_0$, let V be t-optimal for a given instance of [BIF], and let F be an optimal solution for that instance. Then*

$$\frac{|V|}{|F|} \ge \frac{2}{m} - \epsilon_m(t),$$

where

$$\epsilon_m(t) = \begin{cases} \dfrac{m-2}{m((m-1)^{s+1}-1)} & \text{if } t = 2s; \\ \dfrac{2(m-2)}{m(m(m-1)^s-2)} & \text{if } t = 2s-1. \end{cases}$$

Observe that $\epsilon_m(t) \to 0$ as $t \to \infty$. To give an impression of how t enters the bound on the right-hand side of Theorem 3.1, we point out that for $t = 0, \dots, 5$ the values of $2/m - \epsilon_m(t)$ are $\frac{1}{3}$, $\frac{1}{2}$, $\frac{5}{9}$, $\frac{3}{5}$, $\frac{13}{21}$, $\frac{7}{11}$ when $m = 3$ and $\frac{1}{4}$, $\frac{2}{5}$, $\frac{7}{16}$, $\frac{8}{17}$, $\frac{25}{52}$, $\frac{26}{53}$ when $m = 4$.

For the proof of the case $t > 0$ of Theorem 3.1 we need the following combinatorial result of Hurkens and Schrijver [12, Theorem 1].

PROPOSITION 3.2 (Hurkens and Schrijver). *Let $p, q \in \mathbb{N}$, let V be a set of size q, and let $E_1, \dots, E_p$ be subsets of V. Furthermore, let $m, t \in \mathbb{N}$ with $m \ge 3$ such that the following hold:*

(i) *Each element of V is contained in at most m of the sets $E_1, \dots, E_p$.*

(ii) *For any $r \le t$, any r of the sets among $E_1, \dots, E_p$ cover at least r elements of V.*

Then

$$\frac{p}{q} \le \begin{cases} \dfrac{m(m-1)^s - m}{2(m-1)^s - m} & \text{if } t = 2s-1; \\ \dfrac{m(m-1)^s - 2}{2(m-1)^s - 2} & \text{if } t = 2s. \end{cases}$$

It is convenient to regard V and $\mathcal{E} = \{E_1, \dots, E_p\}$ as a hypergraph $(V, \mathcal{E})$. It is clear that under the hypothesis of (i) and (ii) there is some bound on the quotient

p/q. The bounds given in Proposition 3.2, however, are not that obvious and were proved by a quite involved induction. (In addition, [12] shows that these bounds are tight.)

Let us point out that Hurkens and Schrijver [12] apply Proposition 3.2 to derive bounds for the approximation error of certain set packing heuristics, while in [11] Halldórsson utilizes it for set covering. Our subsequent analysis is based on the ideas of these papers.

Proof of Theorem 3.1. For a direct proof of the case $t = 0$, note that the effect of V has to be at least $|F|$ since otherwise the effect of $F \setminus V$ with respect to V would be greater than $(m-1)|F|$. In this case some point of F would have effect m and could hence be added to V without violating the constraints of [BIF], in contradiction to the assumption. Since the effect of V is exactly $m|V|$, the result follows.

Turning now to the general result, we note first that it suffices to give a proof under the additional assumption that $V \cap F = \emptyset$. The general case then follows via a reduction of the data functions by the X-rays of $V \cap F$ with the aid of the inequality

$$\frac{|V|}{|F|} \geq \frac{|V| - |V \cap F|}{|F| - |V \cap F|} \quad \text{for } |V| < |F|.$$

We define a hypergraph $H = (V, \mathcal{E})$ on the vertex set V with exactly $|F|$ hyperedges (one for each element of F) that satisfies the conditions (i) and (ii) of Proposition 3.2. Let $F = \{f_1, \ldots, f_p\}$ and $V = \{v_1, \ldots, v_q\}$. The family $\mathcal{E}$ of hyperedges is defined by associating to each $k = 1, \ldots, p$ with $f_k \in F$ a set $E_k \subset V$ which encodes the conflicts which the insertion of f_k would cause with respect to $\{f_1, \ldots, f_{k-1}\}$ and V.

For each line $T \in \mathcal{T}$ define a map $\iota_T : F \cap T \mapsto (F \cup V) \cap T$. Let $F \cap T = \{f_{i_1}, f_{i_2}, \ldots, f_{i_a}\}$ and $V \cap T = \{v_{j_1}, v_{j_2}, \ldots, v_{j_b}\}$.

Let $k = |F \cap T| - |V \cap T|$. If $k \leq 0$, we set $\iota_T(f_{i_l}) = v_{j_l}$. If $k > 0$, let

$$\iota_T(f_{i_l}) = \begin{cases} f_{j_l} & : \text{for } l \leq k \text{ and} \\ v_{i_{l-k}} & : \text{otherwise.} \end{cases}$$

Now we define the improvement set E_f for a given $f \in F$ by

$$E_f = \{\iota_T(f) \colon T \ni f\} \cap V.$$

We show that the assumptions of Proposition 3.2 are satisfied for $t' = t + 1$. To verify (i) recall that a point $v \in V$ lies in a set E_f if and only if there is a line T_v with $\iota_{T_v}(f) = v$. This can happen only once for each line through v; hence v is contained in at most m different sets E_f.

Next, we show that H has property (ii) of Proposition 3.2. Assume on the contrary that there are sets $E_{k_1}, \ldots, E_{k_{r+1}}$ that cover at most r elements of V for some $r \leq t$. (Here we write E_{k_i} for $E_{f_{k_i}}$ to avoid triple indices.) By choosing r to be minimal with this property, we can assume that $E_{k_1}, \ldots, E_{k_{r+1}}$ cover exactly r elements of V.

Let us consider the set

$$S = \big(V \setminus (E_{k_1} \cup \cdots \cup E_{k_{r+1}})\big) \cup \{f_{k_1}, \ldots, f_{k_{r+1}}\}.$$

We show that the set S is feasible for the given instance of [BIF]. Let $T \in \mathcal{T}$. If $|F \cap T| \leq |V \cap T|$, we have

$$|S \cap T| \leq |V \cap T| \leq \phi(T).$$

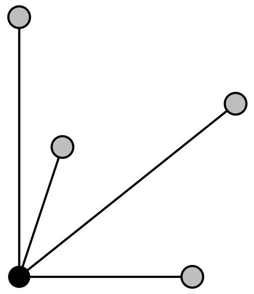

FIG. 1. *The greedy bound is tight. (Grey points belong to F; the black point constitutes V.)*

On the other hand, $|F \cap T| > |V \cap T|$ yields

$$|S \cap T| \leq |F \cap T| \leq \phi(T).$$

(The latter inequalities $|V \cap T| \leq \phi(T)$ and $|F \cap T| \leq \phi(T)$ follow from the fact that both V and F are feasible solutions.) This shows that S is indeed feasible for the given instance of [BIF].

Since S is obtained from V by deleting the r elements of $E_{k_1} \cup \cdots \cup E_{k_{r+1}}$ and inserting the $r+1$ elements $\{f_{k_1}, \ldots, f_{k_{r+1}}\}$, S facilitates an r-improvement, which is a contradiction to the assumption of t-optimality of V.

Summarizing, we have seen that (i) and (ii) of Proposition 3.2 hold for H and $t' = t+1$, and we obtain

$$\frac{p}{q} = \frac{|F|}{|V|} \leq \begin{cases} \dfrac{m(m-1)^s - m}{2(m-1)^s - m} & : \quad t+1 = 2s-1, \\[2ex] \dfrac{m(m-1)^s - 2}{2(m-1)^s - 2} & : \quad t+1 = 2s. \end{cases}$$

Hence

$$\frac{|V|}{|F|} = \frac{2}{m} - \left(\frac{2}{m} - \frac{|V|}{|F|}\right) \geq \frac{2}{m} - \epsilon_m(t)$$

which yields the assertion. □

Deterministic polynomial-time algorithms that meet the requirements of Theorem 3.1 include the greedy algorithm (for $t = 0$), or any other algorithm, according to Paradigm 2.1. In case that t-optimality is guaranteed for some $t \in \mathbb{N}_0$ when the algorithm stops, the results, however, also extend to techniques like *simulated annealing* where, with some probability, changes are allowed that replace a current feasible set by an inferior one.

The following examples show that the bounds given in Theorem 3.1 are tight in the worst case already in the most basic situations.

EXAMPLE 3.3. *Let $m \geq 3$ and let $u_1, \ldots, u_m \in \mathbb{Z}^d$ be m pairwise different lattice directions in $\mathbb{E}^d$. Let $F = \{\nu_1 u_1, \ldots, \nu_m u_m\} \subset \mathbb{Z}^d$ for some scaling factors $\nu_1, \ldots, \nu_m \in \mathbb{Z} \setminus \{0\}$. The X-rays of F in the directions $u_1, \ldots, u_m$ are taken as data functions for an instance of [BIF]. If the factors ν_i are chosen so that $G = F \cup \{0\}$, then $V = \{0\}$ is a greedy-optimal solution for [BIF]. Of course $\frac{|V|}{|F|} = \frac{1}{m}$; see Figure* 1.

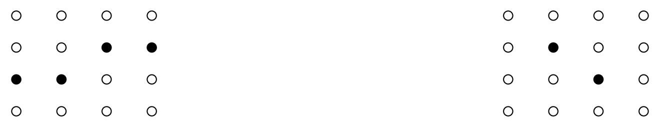

FIG. 2. *The* 1*-optimality bound is tight for three directions. (Black points denote F in the left picture and V in the right picture.)*

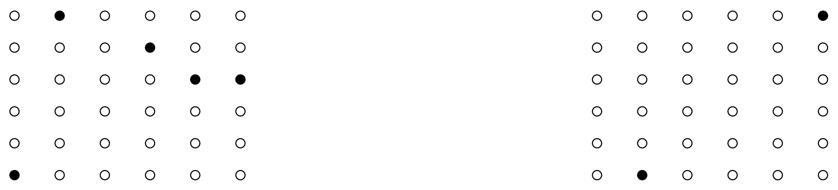

FIG. 3. *The* 1*-optimality bound is tight for four directions. (Black points denote F in the left picture and V in the right picture.)*

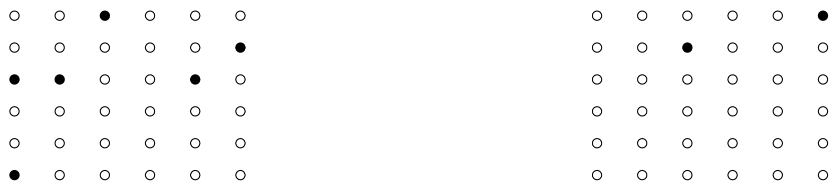

FIG. 4. *The* 1*-optimality bound is tight for five directions. (Black points denote F in the left picture and V in the right picture.)*

EXAMPLE 3.4. *Let $m = 3$ and let $u_1, u_2, u_3 \in \mathbb{Z}^2$ be the directions $(1,0), (0,1), (1,1)$. The X-rays of $F = \{(0,1),(1,1),(2,2),(3,2)\}$ in the directions u_1, u_2, u_3 are taken as data functions for an instance of [BIF]. Then $V = \{(1,2),(2,1)\}$ is* 1*-optimal and $\frac{|V|}{|F|} = \frac{1}{2} = \frac{2}{3} - \epsilon_3(1)$; see Figure* 2.

EXAMPLE 3.5. *Let $m = 4$ and let $u_1, u_2, u_3, u_4 \in \mathbb{Z}^2$ be the directions $(1,0), (0,1), (1,1), (1,2)$. The X-rays of $F = \{(0,0),(1,5),(3,4),(4,3),(5,3)\}$ in the directions u_1, u_2, u_3, u_4 are taken as data functions for an instance of [BIF]. Then $V = \{(1,0), (5,5)\}$ is* 1*-optimal and $\frac{|V|}{|F|} = \frac{2}{5} = \frac{2}{4} - \epsilon_4(1)$; see Figure* 3.

EXAMPLE 3.6. *Let $m = 5$ and let $u_1, \dots, u_5 \in \mathbb{Z}^2$ be the directions $(1,0), (0,1), (1,1), (1,2), (2,1)$. The X-rays of $F = \{(0,0),(0,3),(1,3),(2,5),(4,3),(5,4)\}$ in the directions $u_1, \dots, u_5$ are taken as data functions for an instance of [BIF]. Then $V = \{(2,4),(5,5)\}$ is* 1*-optimal and $\frac{|V|}{|F|} = \frac{1}{3} = \frac{2}{5} - \epsilon_5(1)$; see Figure* 4.

Clearly, there are smarter ways to insert points into the grid than by just greedily putting one in when it fits. A more natural strategy is, for example, to apply a back-projection technique, where each candidate point gets a weight based on the X-ray values of all lines through this point. A typical example is given in Algorithm 3.7 below. In this algorithm, a specific direction S_1 is chosen, which dictates the order in which candidate points are considered for insertion into the set of points L that will eventually form V and the set of holes E (that is disjoint from V). For a fixed line T parallel to S_1, each point g on T gets a weight which depends on the number of points still to be inserted and on the number of candidate points still available on the lines $g+S_i$ for $i \geq 2$; cf. step 2.1 of Algorithm 3.7. The corresponding ratio is a value in $[0,1]$. A value of 0 for a line $g+S_i$ indicates that the point g cannot be inserted into L and a value of 1 indicates that the point must be inserted into L. Therefore, the product over all $m-1$ other lines is a natural indicator for comparing the relative importance of the points on line T.

ALGORITHM 3.7 (weighted greedy strategy).

- INPUT: *Data functions* $\phi_1,\dots,\phi_m$ *for the given lines* $S_1,\dots,S_m$.
- OUTPUT: *A set* $L \subset G$ *feasible for the given instance of [BIF].*
- COMPUTATION:

```
1. Initialize $L = E = \emptyset$ and choose a specific direction, say $S_1$.
2. For all $T \in \mathcal{T}_1$ do:
    2.1. For all $g \in G \cap T$ determine
```

$$w_g = \prod_{i=2}^{m} \frac{\phi_i(g+S_i) - |(g+S_i)\cap L|}{|(G \setminus (L \cup E)) \cap (g+S_i)|}.$$

```
    2.2. Sort $G \cap T$ according to decreasing weights $w_g$, $g \in G \cap T$, and add
         the
```

$$\min\{\phi_1(g+S_1), |\{g \in G \cap T \colon w_g > 0\}|\}$$

```
         first elements of $G \cap T$ to $L$ and the remaining ones to $E$.
```

It is a well-known result, and already known by Lorentz [14], that a similar strategy (with a proper ordering of the lines) leads to an exact algorithm for $m = 2$ directions for consistent instances in the plane; cf. [16]. This suggests that Algorithm 3.7 might be substantially better for arbitrary m than the pure greedy algorithm, an expectation that is confirmed by the experiments stated in section 4.

Let us point out that the solutions produced by the variant of Algorithm 3.7 that is obtained by replacing the weights w_g by

$$w'_g = \prod_{i=1}^{m} \frac{\phi_i(g+S_i) - |(g+S_i)\cap L|}{|(G \setminus (L \cup E)) \cap (g+S_i)|}$$

coincide with the solutions produced by Algorithm 3.7. In fact, while w'_g usually differs from w_g, the order of points on a line in direction S_1 produced by these weights are the same.

3.3. Greedy-type insertion for outer approximation. By changing the stopping rule in Paradigm 2.1, an algorithm for solving [BIF] can be extended to an algorithm for solving [BOF]. Instead of inserting points into a set $U \subset G$ only as long as all constraints of [BIF] are satisfied, such an algorithm inserts points until the constraints of [BOF] are satisfied for the first time. As one would never insert a point into the set U that has effect 0, any such heuristic approximates [BOF] by a factor of at most m. This seems to be the dual result to Theorem 3.1 for the case $t = 0$ but it is not since the final set U is not 0-optimal in general.

ALGORITHM 3.8 (greedy insertion strategy for [BOF]).

- INPUT: *Data functions* $\phi_1,\dots,\phi_m$ *for the given lines* $S_1,\dots,S_m$.
- OUTPUT: *A set* $U \subset G$ *feasible for the given instance of [BOF].*
- COMPUTATION:

```
1. Initialize $U = \emptyset$ and $l = m$.
```

2. *Repeat the following step until* $l = 0$*:*
   2.1. *Add points of effect* l *to* U *as long as such points exist.*
   2.2. *Decrease* l *by* 1.

In what follows it will often be necessary to regard the points of U as ordered. This underlying order will always be the point insertion order produced by Algorithm 3.8.

The performance guarantees given in the next theorem are derived by a careful analysis of the m iterations of step 2.1 in Algorithm 3.8. Further, an additional slight refinement of the algorithm is analyzed. This refinement consists of a combined treatment of points of effects 1 and 2 by means of matching techniques. More precisely, for $l = 1, \dots, m$ let $U_l \subset U$ be the set of points constructed for the parameter l in step 2.1. Then, in the modified version, $U_m, \dots, U_1$ are first constructed by step 2 of Algorithm 3.8 and, subsequently, the following computation is appended as step 3 in order to decrease $|U_1 \cup U_2|$.

3. *Repeat the following procedure until no further improvements occur:*
   3.1. *Define a graph* (V, E) *on the vertex set* $V = \cup_{i=1}^{m} \mathcal{T}_i$ *of all lines as follows: For a vertex* $v \in \mathcal{T}_i$, $1 \le i \le m$, *define the degree*

$$b_v = \max\{0, \phi_i(v) - X_{S_i}(U_3 \cup \cdots \cup U_m)(v)\}.$$

The edges E *are given by means of the set* $G' = G \setminus (U_3 \cup \cdots \cup U_m)$ *in the following way: For* $g \in G'$ *let* $e_g = \{v \in V : g \in v \text{ and } b_v > 0\}$*. (Note that* $|e_g| \le 2$ *since there are no points of effect at least* 3 *left in* G'*.) Now construct a minimum b-edge-cover* M *for* (V, E) *and set* $U_{1,2} = \{g \in G' : e_g \in M\}$ *and* $U = U_{1,2} \cup U_3 \cup \cdots \cup U_m$.

Algorithm 3.8 can be implemented so as to have a polynomial running time. Using, e.g., Gabow and Tarjan's [4] weighted perfect matching algorithm to solve the capacitated b-matching problem and the b-edge cover problem, step 3.1 can also be carried out in polynomial time. A feasible set U (together with an insertion order) which does not allow any further improvements by means of the procedure in step 3.1 is called *matching-optimal* (with respect to that order). Note that the iteration of step 3 terminates in one step, since after one call upon 3.1 no further improvements are possible. This will be different after another refinement of Algorithm 3.8 is appended as step 3.2 in subsection 3.4.

THEOREM 3.9. *Let* U *be a set of points constructed by Algorithm* 3.8 *and let* F *be any feasible solution for [BOF].*

(a) *Then*

$$\frac{|U|}{|F|} \le 1 + \frac{1}{2} + \cdots + \frac{1}{m} = H(m) < 1 + \log(m).$$

(b) *If* U *is matching-optimal, e.g., constructed by Algorithm* 3.8 *extended by step* 3*, then*

$$\frac{|U|}{|F|} \le \frac{5}{6} + \frac{1}{2} + \cdots + \frac{1}{m} = H(m) - \frac{1}{6} < \frac{5}{6} + \log(m).$$

The bounds for $|U|/|F|$ in Theorem 3.9(a) are $\frac{11}{6}, \frac{25}{12}, \frac{137}{60}$ for $m = 3, 4, 5$, respectively. In (b) they are $\frac{5}{3}, \frac{23}{12}, \frac{127}{60}$.

Proof of Theorem 3.9. (a) Let U_l be again the set of points inserted in step 2.1 of Algorithm 3.8 for parameter l, i.e., the points which yield an effect of l upon insertion, and let u_l be the cardinality of U_l. The effect e_l of $U_1 \cup \cdots \cup U_l$ with respect to $U_{l+1}, \ldots, U_m$ is given by $e_l = u_1 + 2u_2 + \cdots + lu_l$. On the other hand, we show that e_l is bounded from above by $l|F|$ for some l.

To this end, let e be the total effect to be attained and suppose to the contrary that $e_l > l|F|$. Consider the set $F' = F \setminus (U_{l+1} \cup \cdots \cup U_m)$. The union of F' and $U_{l+1} \cup \cdots \cup U_m$ contains F, which is feasible for the given instance of [BOF], and thus has effect e. Therefore, the effect of F' with respect to $U_{l+1} \cup \cdots \cup U_m$ is exactly e_l and hence, by our assumption, greater than $l|F|$. Since $|F'| \leq |F|$, this implies by the pigeonhole principle that there is at least one point $g \in F'$ with effect at least $l + 1$. This, however, means that the algorithm would have chosen g rather than some point in $U_1 \cup \cdots \cup U_l$ since all these points have effect at most l with respect to $U_{l+1} \cup \cdots \cup U_m$, a contradiction. Thus

$$e_l = u_1 + 2u_2 + \cdots + lu_l \leq l|F| \tag{5}$$

for $l = 1, \ldots, m$. Denoting the inequality (5) for parameter $l \in \{1, \ldots, m\}$ by I_l we consider the positive linear combination

$$\frac{1}{m} I_m + \sum_{l=1}^{m-1} \frac{1}{l(l+1)} I_l \tag{6}$$

of $I_1, \ldots, I_m$. Collecting the terms on the left and on the right of (6) we obtain

$$\sum_{i=1}^{m} u_i \leq |F| + \sum_{l=1}^{m-1} \frac{1}{l+1} |F|,$$

which is equivalent to the assertion in (a).

The proof of (b) uses the same arguments as that of (a) with the difference that appending step 3.1 to Algorithm 3.8 allows us to improve inequality I_2 to $u_1 + u_2 \leq |F|$ (instead of $u_1 + 2u_2 \leq 2|F|$).

To prove the new inequality, note that the subset $U_{1,2}$ of G' is determined in step 3.1 as a minimum b-edge-cover of (V, E). By construction it follows that $U = U_{1,2} \cup U_3 \cup \cdots \cup U_m$ is feasible for the given instance of [BOF]. Moreover, with $F' = F \setminus (U_3 \cup \cdots \cup U_m)$ the set $\{e_g : g \in F'\}$ is also a b-edge-cover of (V, E). Since $U_{1,2}$ is the disjoint union of (the new sets) U_1 and U_2 and is a minimum b-edge-cover, it follows that

$$|U_{1,2}| = u_1 + u_2 \leq |F'| \leq |F|. \tag{7}$$

With this inequality (instead of inequality I_2) we are led to consider a positive linear combination of type (6) with the coefficient 1/2 of I_1 replaced by 1/3. This reduces the contribution of I_1 to the coefficient of F on the right-hand side by 1/6. Since the other factors remain unchanged, the bound of (b) follows. □

3.4. Outer approximation via *r*-improvements. The aim of this subsection is to analyze an additional refinement of Algorithm 3.8 by means of r-improvements. The first step on the way to improved bounds is to study the impact of r-improvements separately (Theorem 3.10(a)). Afterwards, the additional gain of r-improvements applied to a matching-optimal configuration is considered by appending to Algorithm 3.8 the following step 3.2 for some (fixed) $t \in \mathbb{N}_0$.

3.2. *Apply all r-improvements for $r \leq t$ to $U_1 \cup U_2 \cup U_3$ that decrease U without destroying its feasibility.*

Note that the notation U_1, U_2, U_3 refers to the *updated* sets that are produced in the course of the algorithm. As in this variant r-improvements are applied only to the set $U_1 \cup U_2 \cup U_3$, the resulting algorithm is faster than a general r-improvement algorithm.

Clearly, since $t \in \mathbb{N}$ is a fixed parameter, step 3.2 can be performed in polynomial time. A trivial upper bound for the running time is $O(|G|^{2t+2})$. The geometry of discrete tomography, however, allows us to significantly reduce this bound for many values of t. The reason is that we do not need to consider all pairs of t- and $t+1$-subsets of $U_1 \cup U_2 \cup U_3$ but only those which satisfy certain compatibility conditions.

A set $U \subset G$ (together with an insertion order) is called *effect-3-t-optimal* (with respect to this order), if it cannot be decreased by the procedure of step 3.2 above, i.e., by any r-improvement, on the points of effects 1, 2, and 3.

THEOREM 3.10. *Let F be a minimum solution for a given instance of [BOF] and let $t \in \mathbb{N}_0$.*

(a) *Let U be t-optimal for that instance; then*

$$\frac{|U|}{|F|} \leq \frac{m}{2} + \epsilon_m(t)\,, \text{ where } \epsilon_m(t) = \begin{cases} \dfrac{m(m-2)}{4(m-1)^{s+1} - 2m} & : \text{ if } t = 2s; \\[2ex] \dfrac{(m-2)}{2(m-1)^s - 2} & : \text{ if } t = 2s-1. \end{cases}$$

(b) *Let $m = 3$ and $t = 2s+1$, $s \in \mathbb{N}$. Furthermore, assume that U is matching-optimal and t-optimal (that is, effect-3-t-optimal); then*

$$\frac{|U|}{|F|} \leq \frac{7}{5} + \epsilon'(t)\,, \text{ where } \epsilon'(t) = \begin{cases} \dfrac{6}{25 \cdot 2^{r+1} - 15} & : \text{ if } s = 2r-1; \\[2ex] \dfrac{2}{5(5 \cdot 2^r - 1)} & : \text{ if } s = 2r. \end{cases}$$

(c) *Let $t \geq 5$ and let U be matching-optimal and effect-3-t-optimal; then*

$$\frac{|U|}{|F|} \leq \frac{2}{3} + \frac{1}{2} + \cdots + \frac{1}{m} < \frac{2}{3} + \log(m).$$

The values of $m/2 + \epsilon_m(t)$ in Theorem 3.10(a) for $m = 3$ and $t = 0, \ldots, 5$ are $3, 2, \frac{9}{5}, \frac{5}{3}, \frac{21}{13}, \frac{11}{7}$ and for $m = 4$ they are $4, \frac{5}{2}, \frac{16}{7}, \frac{17}{8}, \frac{52}{25}, \frac{53}{26}$. The values of $7/5 + \epsilon'(t)$ for $t = 3, 5, 7, 9, 11$ in (b) are $\frac{11}{7}, \frac{3}{2}, \frac{25}{17}, \frac{13}{9}, \frac{53}{37}$. Note that $\epsilon_m(t), \epsilon'_m(t) \to 0$ for $t \to \infty$ for all $m \geq 3$. The upper bound for $|U|/|F|$ in (c) is $\frac{3}{2}, \frac{7}{4}, \frac{39}{20}$ for $m = 3, 4, 5$, respectively.

Proof of Theorem 3.10. (a) is proved by defining a hypergraph $H = (V, \mathcal{E})$ on the vertex set $V = F$ with edges defined for each $g \in U$ that satisfies (i) and (ii) of Proposition 3.2. As in the proof of Theorem 3.1, it suffices to prove the result for $U \cap F = \emptyset$. Again, we define a map $\iota_T : U \cap T \mapsto (U \cup F) \cap T$. This time $\iota_T(u)$ encodes the information which point on T is added to compensate the deletion of u. For each line $T \in \mathcal{T}$ let $U \cap T = \{u_{i_1}, u_{i_2}, \ldots, u_{i_a}\}$ and $F \cap T = \{f_{j_1}, f_{j_2}, \ldots, f_{j_b}\}$.

If $|F \cap T| \geq |U \cap T|$ we set $\iota_T(u_{i_l}) = f_{j_l}$ for $l = 1, 2, \ldots, a$. If $|F \cap T| < |U \cap T|$ let

$$\iota_T(u_{i_l}) = \begin{cases} f_{j_l} : \text{for } l \leq |F \cap T| \text{ and} \\ u_{i_l} : \text{otherwise.} \end{cases}$$

Now we define the "improvement sets" E_u for a given $u \in U$ by

$$E_u = \{\iota_T(u)\colon T \ni u\} \cap F.$$

As in the proof of Theorem 3.1, the number m of directions gives the bound in (i) and the t-optimality implies condition (ii) of Proposition 3.2 for $t' = t + 1$. Thus Proposition 3.2 can be applied, and the bound given in (a) follows.

In order to prove (b), let $U = U_1 \cup U_2 \cup U_3$ be a partition of U into subsets of points of effect 1, 2, and 3, respectively. As each point u of U_1 has effect 1 we can associate with it the line $T(u)$ to which it contributes. For $T \in \mathcal{T}$ let

$$U_T = \{u \in U_1 \cap T\colon T = T(u)\}.$$

Since $|U_T| \le \phi(T) \le |F \cap T|$ for $T \in \mathcal{T}$ we can define an injection $\kappa_T : U_T \mapsto F \cap T$. Now $U_1 = \bigcup_{T \in \mathcal{T}} U_T$, and let $\kappa : U_1 \mapsto F$ be the map induced by the injections κ_T. We show that κ is injective. In fact, if there were $u_1, u_2 \in U_1$ with $\kappa(u_1) = \kappa(u_2)$, then $T(u_1) \ne T(u_2)$, whence

$$(U \setminus \{u_1, u_2\}) \cup \{\kappa(u_1)\}$$

was feasible for the given instance of [BOF] contradicting the 1-optimality of U. It follows that

$$|U_1| = |F_1|, \tag{8}$$

where $F_1 = \kappa(U_1)$.

For the set of remaining points $F_0 = F \setminus F_1$, we use the fact that there is no r-improvement for U for any $r \le 2s + 1$ in order to show

$$|U_2| + |U_3| \le \left(\frac{3}{2} + \epsilon_3(s-1)\right) |F_0|. \tag{9}$$

To this end, let us first define the *reduced* X-ray functions

$$\gamma_i(T) = \min\{\phi_i(T), X_{S_i}F_0(T)\} \quad \text{for } T \in \mathcal{T}_i \text{ and } i = 1, 2, 3,$$

set $U_{2,3} = U_2 \cup U_3$, and note that $U_{2,3}$ is feasible for the instance $I = \{\gamma_1, \gamma_2, \gamma_3\}$ of [BOF]. Next we define a hypergraph $H = (F_0, \mathcal{E})$ with $|U_{2,3}|$ edges, again with the aid of maps ι_T for $T \in \mathcal{T}$. This time $\iota_T : U_{2,3} \cap T \mapsto (U_{2,3} \cup F_0) \cap T$, and $\iota_T(u)$ encodes the information which point on T is added to compensate for the deletion of u in the reduced problem. Let $T \in \mathcal{T}$ and $U_{2,3} \cap T = \{u_{i_1}, u_{i_2}, \dots, u_{i_a}\}$, $F_0 \cap T = \{f_{j_1}, f_{j_2}, \dots, f_{j_b}\}$.

If $|F_0 \cap T| \ge |U_{2,3} \cap T|$, we set $\iota_T(u_{i_l}) = f_{j_l}$ for $l = 1, 2, \dots, a$. If $|F_0 \cap T| < |U_{2,3} \cap T|$, let

$$\iota_T(u_{i_l}) = \begin{cases} f_{j_l} : \text{for } l \le |F_0 \cap T| \text{ and} \\ u_{i_l} : \text{otherwise.} \end{cases}$$

Now we define the "improvement sets" E_u for a given $u \in U_{2,3}$ by

$$E_u = \{\iota_T(u)\colon T \ni u\} \cap F_0.$$

To obtain (9), we want to apply Proposition 3.2 to H. Clearly, condition (i) of Proposition 3.2 holds with $m = 3$. Next we show that condition (ii) holds with parameter s. Assume on the contrary that there are $l+1$ sets $E_{u_{i_1}}, \dots, E_{u_{i_{l+1}}}$, with $l+1 \le s$, that cover only l elements $f_1, \dots, f_l \in F_0$, and let l be minimal with this property.

Let $\hat{U} = \{u_{i_1}, \dots, u_{i_{l+1}}\}$, $\hat{F} = \{f_1, \dots, f_l\}$ and set $S = (U_{2,3} \setminus \hat{U}) \cup \hat{F}$. Of course, S results from $U_{2,3}$ via an l-improvement. Let $e_{U_{2,3}\setminus\hat{U}}(\hat{U})$ $\langle e_{U_{2,3}\setminus\hat{U}}(\hat{F})\rangle$ denote the effect of $\hat{U}$ $\langle\hat{F}\rangle$, with respect to $U_{2,3} \setminus \hat{U}$ and the original data (ϕ), and let $\bar{e}$ be the corresponding effect-function for the reduced data (γ). We show that

$$e_{U_{2,3}\setminus\hat{U}}(\hat{U}) \le e_{U_{2,3}\setminus\hat{U}}(\hat{F}) + l + 3. \tag{10}$$

Of course, $e_{U_{2,3}\setminus\hat{U}}(\hat{U}) \le 3l+3$ and, since S is feasible for I, $\bar{e}_{U_{2,3}\setminus\hat{U}}(\hat{U}) = \bar{e}_{U_{2,3}\setminus\hat{U}}(\hat{F})$. Further, it follows from the minimality of l that $\bar{e}_{U_{2,3}\setminus\hat{U}}(\hat{U}) \ge 2l$. In fact, if $\bar{e}_{U_{2,3}\setminus\hat{U}}(\hat{U}) \le 2l-1$, then there must exist an $f \in \hat{F}$ of effect 1 with respect to $U_{2,3} \setminus \hat{U}$ and the reduced data; hence

$$(U_{2,3} \setminus (\hat{U} \setminus \{u_f\}) \cup (\hat{F} \setminus \{f\}),$$

where u_f is an element of $\hat{U}$ on the line T that carries the effect of f, would constitute an $(l-1)$-improvement. This contradiction implies that

$$e_{U_{2,3}\setminus\hat{U}}(\hat{F}) + l + 3 \ge \bar{e}_{U_{2,3}\setminus\hat{U}}(\hat{F}) + l + 3 = \bar{e}_{U_{2,3}\setminus\hat{U}}(\hat{U}) + l + 3 \ge 3(l+1),$$

as claimed.

Next we want to lift the l-improvement for $U_2 \cup U_3$ to an r-improvement for $U_1 \cup U_2 \cup U_3$ with $r \le 2l$. From (10) we know that $e_\emptyset(((U_1 \cup U_{2,3})\setminus\hat{U})\cup\hat{F}) \ge e-(l+3)$. Hence, it suffices to add at most $l+3$ suitable elements $\{g_1, g_2, \dots, g_{l'}\}$ of F_1 to ensure that

$$\Big((U_1 \cup U_{2,3}) \setminus \hat{U}\Big) \cup \Big(\hat{F} \cup \{g_1, g_2, \dots, g_{l'}\}\Big)$$

is feasible. Furthermore, the points $\kappa^{-1}(g_1), \dots, \kappa^{-1}(g_l)$ can be deleted from U_1 without destroying feasibility; i.e.,

$$\Big((U_1 \cup U_{2,3}) \Big\backslash \Big(\hat{U} \cup \{h_1, h_2, \dots, h_{l'}\}\Big)\Big) \cup \Big(\hat{F} \cup \{g_1, g_2, \dots, g_{l'}\}\Big)$$

is feasible for the (original) data (ϕ). Let $r = l + l'$; then $r \le 2l+3 \le 2s+1 = t$. Hence, the existence of this lifted r-improvement contradicts the t-optimality of U. So, property (ii) holds, Proposition 3.2 can be applied, and (9) follows.

In order to derive the bound of (b), inequality (8), matching-optimality (i.e., inequality (7)), and the bound $3|F|$ on the total effect of U are combined to obtain

$$3|U| = |U_1| + \underbrace{(|U_1| + |U_2|)}_{\le|F|} + \underbrace{(|U_1| + 2|U_2| + 3|U_3|)}_{\le 3|F|} \le |F_1| + 4|F|. \tag{11}$$

Furthermore, inequality (9) implies

$$|U| = |U_1| + (|U_2| + |U_3|) \le |F_1| + \left(\frac{3}{2} + \epsilon_3(s-1)\right)|F_0|. \tag{12}$$

Multiplying (11) with $\frac{1}{2}+\epsilon_3(s-1)$, adding (12), and using $|F_0|+|F_1|=|F|$ then give

$$\left(\frac{5}{2}+3\epsilon_3(s-1)\right)|U| \le \left(\frac{7}{2}+5\epsilon_3(s)\right)|F|,$$

which implies assertion (b).

Note that the proof provides a result that is slightly stronger than assertion (b). In fact, the argument does not use the assumption $m=3$ "globally" but only "locally." More precisely, let $m \ge 3$, $s \in \mathbb{N}$, $t=2s+1$, and let U be matching-optimal and effect-3-t-optimal. Then

$$\begin{array}{rcl} |U_1| & = & |F_1| \quad \text{and} \\ |U_1|+|U_2|+|U_3| & \le & |F|+\left(\frac{1}{2}+\epsilon_3(s-1)\right)|F_0|. \end{array} \tag{13}$$

Finally, we turn to assertion (c). First, we form the positive linear combination

$$\frac{1}{m}I_m+\sum_{l=4}^{m-1}\frac{1}{l(l+1)}I_l$$

of the inequalities (5) derived in the proof of Theorem 3.9. Collecting terms for $U_1,\dots,U_m$ yields

$$\frac{1}{4}|U_1|+\frac{2}{4}|U_2|+\frac{3}{4}|U_3|+|U_4|+\cdots+|U_m| \le \left(1+\frac{1}{5}+\cdots+\frac{1}{m}\right)|F|.$$

Thus it remains to show that

$$\frac{3}{4}|U_1|+\frac{2}{4}|U_2|+\frac{1}{4}|U_3| \le \frac{3}{4}|F|.$$

Since $5 \le t = 2s+1$, we can apply (13) for $s=2$. This yields

$$2|U_1|+|U_2|+|U_3| \le 2|F|.$$

Matching-optimality implies again

$$|U_1|+|U_2| \le |F|,$$

whence addition of these inequalities gives

$$3|U_1|+2|U_2|+|U_3| \le 3|F|.$$

This concludes the proof of Theorem 3.10. ☐

4. Computational results.

In this section we report on computational results for implementations of the different algorithms outlined in the previous sections.

4.1. Description of the implementations.

We implemented six different algorithms for [BIF]. The first algorithm (GreedyA) is the plain greedy algorithm (see Figure 5) which considers all positions in a random order and tries to place atoms at these positions. The second algorithm (GreedyB) is a variant of the line following greedy algorithm, Algorithm 3.7 (Figure 6). The algorithm chooses a direction with

```
procedure GreedyA
Calculate a random permutation of all points
For each point in the order of this permutation do
    Check whether any line passing through this point is saturated
    If no line is saturated then
        Add the point to the solution set
        Update the sums of the lines passing through this point
```

FIG. 5. *The plain greedy solver.*

maximal support $|\mathcal{T}_i|$. Suppose—in accordance with the notation in Algorithm 3.7—that $i = 1$. The lines $T \in \mathcal{T}_1$ are then considered with respect to decreasing line-weights

$$\phi_1(T)/|G \cap T|.$$

The algorithm usually performs quite well. However, if one regards the "en block" point insertion procedure successively, i.e., as a point-by-point insertion, then the adapted line-weights change and at some point—possibly long before the last point of the block has been inserted—another line might be more profitable. This idea is pursued in a third greedy algorithm (GreedyC) which changes the weights of all lines and uninspected points after a new point is placed; see Figure 7. The initial problem is, of course, that after each insertion a complete search for the next position of maximum weight is necessary. This increases the computation times dramatically. A good data structure for keeping the points (partially) ordered according to their weights is a heap. After a point insertion, it suffices to update the weights of points on lines through the new point. While a heap can perform this quite efficiently, this procedure is still pretty time consuming since the weights of points change frequently, without the element even being close to the top of the heap. We decided therefore to use a lazy-update. For this we take the top element of the heap and recompute its weight. Then we compare its stored weight with its actual weight (they might differ due to recent insertions). If the weights are equal, this is still the top element of the heap, and we can try to insert it. If the weights differ, the candidate point gets the new weight and the heap needs to be restructured. After the restructuring we start again with the (new) top element.

The last type of algorithm is the 1-improvement algorithm according to Paradigm 2.1. We tried three different variants (ImprovementA, ImprovementB, ImprovementC) depending on the greedy algorithm (GreedyA, GreedyB, GreedyC) used first; see Figure 8 As the 1-improvement algorithm already needs 22 minutes on average for some instances and the results are very good, we did not implement higher improvement algorithms (like 2-improvement, etc.).

4.2. Performance of the implemented algorithms. In this subsection we report on different experiments we conducted with the algorithms described in the previous section. We performed several tests for problems of size 20×20 to 500×500, with 2 to 5 directions and of density between 10% and 90%. After analyzing the different experiments, we observed that the experiments with varying numbers of directions, but a fixed density of 50%, are most representative and the other series behave similarly. (For more data on the computational performance of the evaluated heuristics for other densities of 1%, 5%, and 20% see de Vries [19].)

Even though our program can solve problems in three dimensions and on arbitrary crystal-lattices, we decided to present here only results for 2-dimensional problems on the square lattice, as in the physical application all directions belong to a single plane

```
procedure GreedyB
Determine a direction with maximal support
Sort the lines parallel to that direction by descending line-weights
For each of these lines ($T$) in this order do
    For each point on $T$ do
        Calculate its weight (the product of the line-weights)
    Sort the points on $T$ with respect to descending weights
    For each point in this order do
        Check whether any line passing through this point is saturated
        If no line is saturated then
            Add the point to the solution set
            Update the sums of the lines passing through this point
```

FIG. 6. *The line following greedy solver.*

```
procedure GreedyC
For each point do
    Calculate the weight of the point (the product of the relative line capacities)
        and insert it into the heap
While there are still points in the heap do
    Find the maximum weight and a corresponding point and remove it from the heap
    Check whether any line passing through this point is saturated
    If no line is saturated then
        Add the point to the solution set
        Update the sums of the lines passing through this point
```

FIG. 7. *The dynamically reordering greedy solver.*

(therefore the problem can be solved in a slice-by-slice manner); furthermore, this restriction should facilitate the comparison with other, less general codes currently under development by various research groups.

Whenever we report either running-times or performances, we report the average of 100 randomly generated instances. We decided to use random instances for two reasons. The first reason is that we still lack sufficient experimental data from the physicists. On the other hand, it is typically easy to detect and then eliminate invariant points, i.e., points that either must belong to every solution or do not belong to any solution. Since the invariant points carry much of the physical a priori knowledge, the reduced problem tends to be quite unstructured.

To obtain a random configuration of prescribed density, we generate a random permutation of the positions of the candidate grid and then place atoms in this order until the described density is reached. After calculating the lines and their sums we discard the configuration itself. Then we preprocessed the problem by calculating the incidence tables, which are necessary for all algorithms. The running-times we report were obtained on an SGI Origin 200 computer with four MIPS R10000 processors at 225MHz with 1GB of main memory and by running three test programs at the same time.

Note that all instances are consistent. There are two reasons for this. First, for inconsistent problems we need the exact solution to evaluate the performance of the heuristics. But for the relevant dimensions there are at present no algorithms available that produce exact solutions in reasonable time. The second reason is that the true nature of the error distribution for the real physical objects has not yet been experimentally determined by the physicists. So it is not clear how to perturb an exact instance to obtain inconsistent problems in a physically reasonable manner.

The performance plotted in Figure 9 is the quotient of the cardinality of the

```
procedure Improvement[ABC]
Calculate a solution $U$ according to GreedyA, GreedyB, or GreedyC
Repeat
    For each point ($p_1$) of the candidate grid do
        If $p_1 \in U$ then continue with the next point
        If no line passing through $p_1$ is saturated then
            Add $p_1$ to $U$
            Update the sums of the lines passing through $p_1$
            Continue with the next point
        If more than one line passing through $p_1$ is saturated then continue with the next point
        For each point ($p_2$) of $U$ on the saturated line do
            For each nonsaturated line ($T_1$) through $p_1$ do
                Calculate the line ($T_2$) parallel to $T_1$ passing through $p_2$
                For each point ($p_3$) on $T_2$ not in $U$ do
                    If the lines passing through $p_3$ and
                        not containing $p_1$ or $p_2$ are nonsaturated and
                        the line passing through $p_3$ and $p_1$ (if existent)
                        has at least one point not in $U$ then
                        Perform the improvement:
                            Remove $p_2$ from $U$
                            Add $p_1$ to $U$
                            Add $p_3$ to $U$
                        Update the sums of all lines passing through $p_1$, $p_2$, or $p_3$
                        Continue with the next point
Until no improvement was done in the last loop
```

FIG. 8. *The improvement solvers.*

approximate solution to that of an optimal solution. The closer it is to 1 the better the result. It turns out that the larger the problems, the better every algorithm performs in terms of relative errors (see Figure 9). Obviously, postprocessing the output of some greedy algorithm with an improvement algorithm cannot decrease the performance (usually it improves the performance). However, it turns out that GreedyB outperforms ImprovementA (for four and five directions) and that GreedyC performs better than ImprovementB (for five directions; for four directions they are similar and for three directions ImprovementB is better).

The running-times for the algorithms GreedyA and GreedyB are less than 4 seconds for all instances (of size up to 500×500). The application of the 1-improvements to their results increases the running-time to up to 110 seconds.

The running-times of GreedyC and ImprovementC increase much faster than those for the other algorithms. Still, they take only up to 1320 seconds. This is long, but in fact these algorithms provide very close approximations, while presently available exact algorithms seem incapable of solving 500×500 problems in less than a century. Furthermore, knowing a solution for a neighboring slice should speed up the solution of the next slice by a good amount; so there is hope of solving even $500 \times 500 \times 500$ real world problems in time that is acceptable in practice.

The better of the presented algorithms are so good that we also compare their *absolute* errors (see Figure 10). As can be seen, the absolute error for ImprovementC seems constant for three directions. (Of course, it follows from [5] that asymptotically there must be a more than constant worst-case error unless $\mathbb{P} = \mathbb{NP}$.) For four and five directions the absolute error appears to be $O(\sqrt{|G|})$.

Another (practically) important issue is that of the distribution of errors among different lines. For this we counted for 100 problems of size 500×500 how many constraints were satisfied with equality, how many needed only one more point for

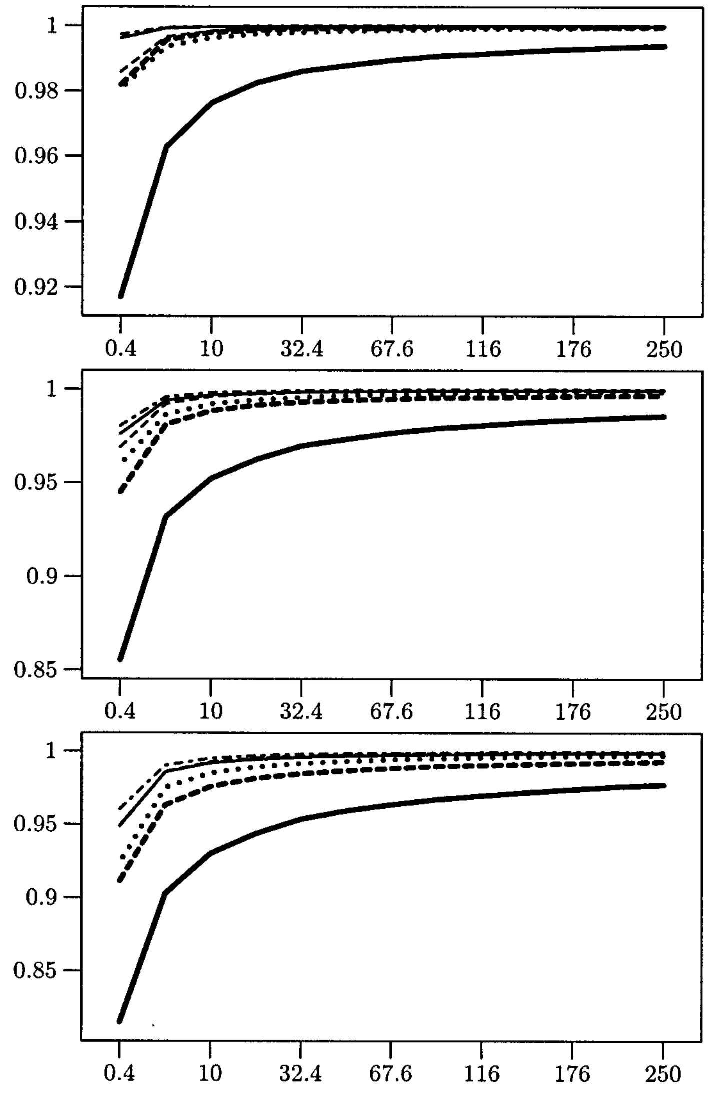

FIG. 9. *Relative performance for* 3 *(top),* 4 *(middle), and* 5 *(bottom) directions on instances of* 50% *density for GreedyA (/), ImprovementA (⁄), GreedyB (·˙), ImprovementB (/), GreedyC (·′), and ImprovementC (·˙′). The abscissa depicts the number of variables in thousands at a quadratic scale and the ordinate depicts the relative performance.*

equality, and so on. Again, it turned out that the algorithms GreedyC and ImprovementC have the best error distribution. In particular, for GreedyC no line occurred with error greater than 1 for 3 and 4 directions; for 5 directions the worst case was 1 instance with a single line of error 2. For ImprovementC the worst cases were 3 instances with one line of error 2 for 3 directions, for 4 directions 1 instance with a single line of error 4, and for 5 directions 1 instance with a single line of error 5. Of course, while 1-improvements never decrease the number of points placed, the variation of errors over the single lines may increase, as it may happen that in a number of improvement steps atoms from the same line are removed.

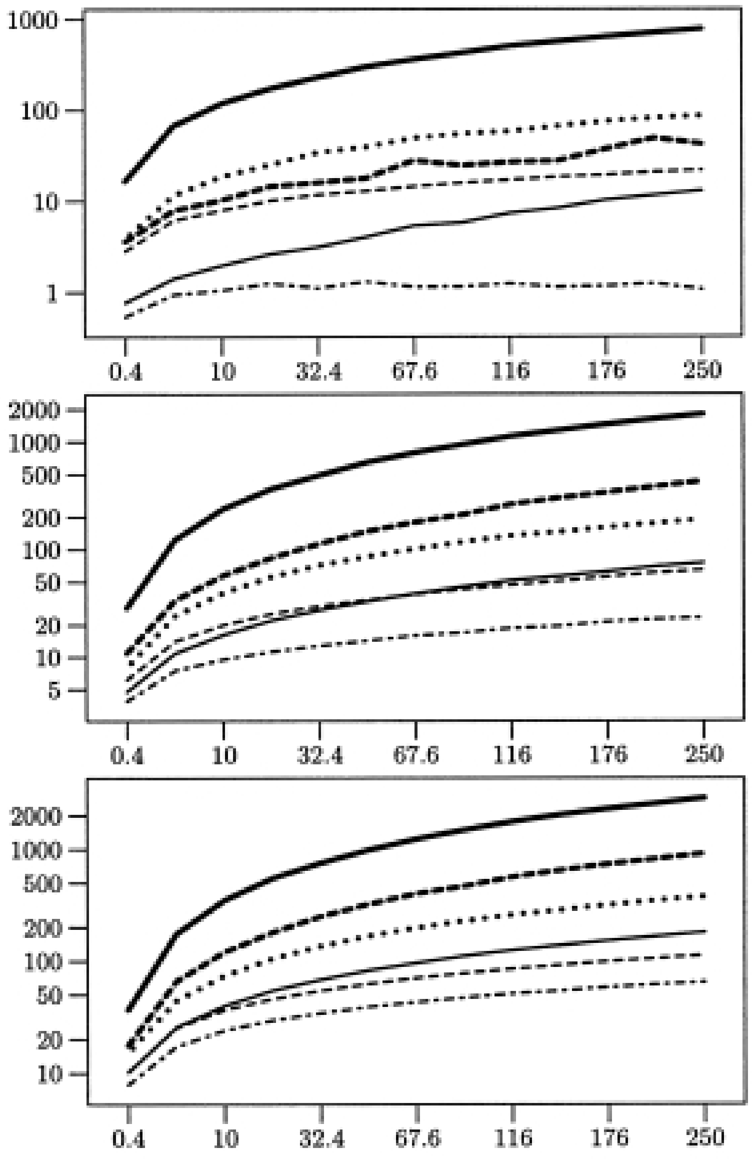

FIG. 10. *Absolute error for* 3 *(top),* 4 *(middle), and* 5 *(bottom) directions on instances of* 50% *density for GreedyA (/), ImprovementA (/), GreedyB (/), ImprovementB (/), GreedyC (/), and ImprovementC (/). The abscissa depicts the number of variables in thousands at a quadratic scale and the ordinate depicts the absolute error at a logarithmic scale.*

For GreedyC only lines with error at most 2 occur, while for ImprovementC a single instance with a line of error 5 came up. In contrast, GreedyA, GreedyB, ImprovementA, and ImprovementB always have a couple of lines with a huge error (see Figure 11). For instance, for GreedyA, ImprovementA, GreedyB, and ImprovementB instances occurred with lines of error 67, 130, 109, and 66. These huge errors do seem inappropriate in the physical application since it is more likely that many lines occur with small error rather than with very large error.

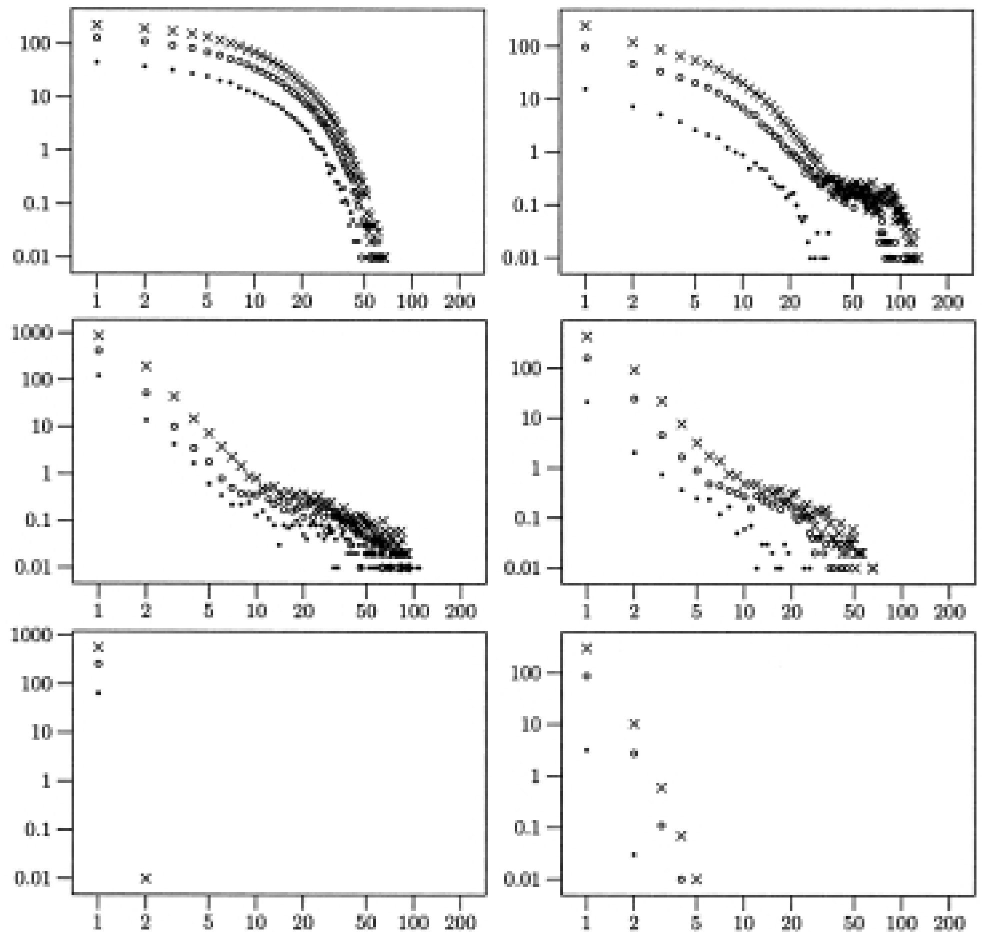

FIG. 11. *Distribution of error for* 100 *instances with* 500^2 *variables, density* 50%, *and* 3 (.), 4 (∘), *and* 5 (×) *directions. Depicted are: GreedyA (top left), ImprovementA (top right), GreedyB (middle left), ImprovementB (middle right), GreedyC (bottom left), and ImprovementC (bottom right). The abscissa depicts the absolute error on a line at a logarithmic scale and the ordinate depicts the average number of lines with this error at a logarithmic scale.*

Acknowledgments. We thank Jens Zimmermann for helping with coding the data structures and algorithms. Furthermore, we thank the referees for some valuable comments.

REFERENCES

[1] S. Chang, *The reconstruction of binary patterns from their projections*, Comm. ACM, 14 (1971), pp. 21–25.

[2] R. Duh and M. Fürer, *Approximation of k-set cover by semi-local optimization*, in Proceedings, STOC '97, ACM, New York, pp. 256–264.

[3] P. Fishburn, P. Schwander, L. Shepp, and R. Vanderbei, *The discrete Radon transform and its approximate inversion via linear programming*, Discrete Appl. Math., 75 (1997), pp. 39–61.

[4] H. Gabow and R. Tarjan, *Faster scaling algorithms for general graph-matching problems*, J. ACM, 38 (1991), pp. 815–853.

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

[6] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP–Completeness*, W.H. Freeman, San Francisco, CA, 1979.

[7] J. Gerbrands and C. Slump, *A network flow approach to reconstruction of the left ventricle from two projections*, Comput. Graphics Image Process., 18 (1982), pp. 18–36.

[8] O. Goldschmidt, D. Hochbaum, and G. Yu, *A modified greedy heuristic for the set covering problem with improved worst case bound*, Inform. Process. Lett., 48 (1993), pp. 305–310.

[9] P. Gritzmann, *On the reconstruction of finite lattice sets from their X-rays*, in Proceedings 7th International Workshop, DGCI '97, Montpellier, France, E. Ahronovitz and C. Fiorio, eds., Lecture Notes in Comput. Sci. 1347, Springer, Berlin, 1997, pp. 19–32.

[10] P. Gritzmann, D. Prangenberg, S. de Vries, and M. Wiegelmann, *Success and failure of certain reconstruction and uniqueness algorithms in discrete tomography*, Int. J. Imaging Syst. Technol., 9 (1998), pp. 101–109.

[11] M. Halldórsson, *Approximating k–set cover and complementary graph coloring*, in Proceedings 5th International Integer Programming and Combinatorial Optimization Conference, Lecture Notes in Comput. Sci., Springer-Verlag, Berlin, 1996, pp. 118–131.

[12] C. A. J. Hurkens and A. Schrijver, *On the size of systems of sets every t of which have an SDR, with an application to the worst-case ratio of heuristics for packing problems*, SIAM J. Discrete Math., 2 (1989), pp. 68–72.

[13] C. Kisielowski, P. Schwander, F. Baumann, M. Seibt, Y. Kim, and A. Ourmazd, *An approach to quantitative high-resolution transmission electron microscopy of crystalline materials*, Ultramicroscopy, 58 (1995), pp. 131–155.

[14] G. Lorentz, *A problem of plane measure*, Amer. J. Math., 71 (1949), pp. 417–426.

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

[16] H. Ryser, *Combinatorial Mathematics*, Carus Math. Monogr. 14, Mathematical Association of America and John Wiley, New York, 1963, ch. 6, Matrices of zeros and ones, pp. 61–78.

[17] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, New York, 1986.

[18] P. Schwander, C. Kisielowski, F. Baumann, Y. Kim, and A. Ourmazd, *Mapping projected potential, interfacial roughness, and composition in general crystalline solids by quantitative transmission electron microscopy*, Phys. Rev. Lett., 71 (1993), pp. 4150–4153.

[19] S. de Vries, *Discrete Tomography, Packing and Covering, and Stable Set Problems: Polytopes and Algorithms*, Ph.D. thesis, Technische Universität München, München, Germany, 1999.

[20] M. Wiegelmann, *Gröbner Bases and Primal Algorithms in Discrete Tomography*, Ph.D. thesis, Technische Universität München, München, Germany, 1998.