Theoretical Computer Science 246 (2000) 53–72

Theoretical Computer Science

www.elsevier.com/locate/tcs

Fast stabbing of boxes in high dimensions ☆

Frank Nielsen [a,b,*]

[a] *INRIA, BP93, 06902 Sophia-Antipolis cedex, France*
[b] *University of Nice, Parc Valrose, 06304 Nice cedex, France*

Received September 1996; revised March 1998
Communicated by J.D. Boissonnat

Abstract

We present in this paper a simple yet efficient algorithm for stabbing a set $\mathscr{S}$ of n axis-parallel boxes in d-dimensional space with $c(\mathscr{S})$ points in output-sensitive time $O(dn \log c(\mathscr{S}))$ and linear space. Let $c^*(\mathscr{S})$ and $b^*(\mathscr{S})$ be, respectively, the minimum number of points required to stab $\mathscr{S}$ and the maximum number of pairwise disjoint boxes of $\mathscr{S}$. We prove that $b^*(\mathscr{S}) \leqslant c^*(\mathscr{S}) \leqslant c(\mathscr{S}) \leqslant b^*(\mathscr{S})(1+\log_2 b^*(\mathscr{S}))^{d-1}$. Since finding a minimal set of $c^*(\mathscr{S})$ points is NP-complete as soon as $d>1$, we obtain a fast precision-sensitive heuristic for stabbing $\mathscr{S}$ whose quality does not depend on the input size. In the case of congruent or constrained isothetic boxes, our algorithm reports, respectively, $c(\mathscr{S}) \leqslant 2^{d-1} b^*(\mathscr{S})$ and $c(\mathscr{S}) = O_d(b^*(\mathscr{S}))$ stabbing points. Moreover, we show that the bounds we get on $c(\mathscr{S})$ are asymptotically tight and corroborate our results with some experiments. We also describe an optimal output-sensitive algorithm for finding a minimal-size optimal stabbing point-set of intervals. Finally, we conclude with insights for further research.

Keywords: Computational geometry; Geometric combinatorial optimization; Output-sensitive algorithms

1. The problem

Let $\mathscr{S}$ be a set of n d-dimensional geometric objects of constant descriptive-size. We say that $\mathscr{S}$ is stabbed by k points if there exist k points so that each object of $\mathscr{S}$ contains at least one of these points. (This stabbing point set is also called a 0-transversal.) Thus, $\mathscr{S}$ can be immobilized under translation with k points. Given

☆ Most of this work was done while the author was visiting at The University of Tokyo, Japan and being at INRIA, BP93, 06902 Sophia-Antipolis cedex, France and University of Nice Sophia-Antipolis, Valrose, France.

* Corresponding author; present address: SONY Computer Science Laboratories Inc., Takanawa Muse Bldg., 3-14-13 Higashi Gotanda, Shinagawa-Ku, Tokyo 141 0022, Japan.
E-mail address: Nielsen@csl.sony.co.jp, frank.nielsen@sophia.inria.fr (F. Nielsen).

PII: S0304-3975(98)00336-3

a set $\mathscr{S}$ as above, finding the minimum $k = c^*(\mathscr{S})$ so that $\mathscr{S}$ can be stabbed by k points has been shown [1] to be NP-complete [15] as soon as $d \geqslant 2$. Even in the one-dimensional case, this problem remains NP-complete for non-convex instances [21]. As a byproduct, this problem is intractable for small values of n (say $n \simeq 50$ and $d = 2$ – see Section 3.5). The geometric covering/piercing problem is also referred in the literature as the *covering set problem* (CSP), or dually as the *hitting set problem* (HSP), where it is transformed into an optimization problem by means of matrix formulations. Let $\mathscr{V} = \{S_i \mid i \in I\}$ be a collection of $v = |\mathscr{V}| = |I|$ subsets of $2^{\mathscr{S}}$ for a set $\mathscr{S}$ of n elements. We want to find a minimal covering collection, i.e., a subset $I' \subseteq I$ of indices such that $\mathscr{S} = \bigcup_{i \in I'} S_i$ with $|I'|$ as small as possible. In other words, we want to minimize $e_v^{\mathrm{T}} x = |I'|$ subject to $Ax \geqslant e_n$ for x a $\{0,1\}^v$-vector, e_k a k-dimensional vector of 1's and A a $(v \times n)$-binary matrix, each column of which is the incidence vector of one of the sets I_i, $1 \leqslant i \leqslant v$. The above formulation gives an *integer linear program*. Using these notations, the set cover system is said ε-*dense* (δ-*super-dense*) if $|S_i| \geqslant \varepsilon |\mathscr{S}|$ (resp. $|S_i| \geqslant |\mathscr{S}| - \mathrm{o}(m^{\delta})$), for all $i \in I$. The CSP remains Max SNP-hard even on dense cases but can be solved polynomially in super-dense cases [3, 26].

Chvátal [8, 37] gave a $\mathrm{O}(nv)$-time greedy algorithm to find a cover set of size $c(\mathscr{S}) \leqslant c^*(\mathscr{S})(\log k - \log\log k + 0.78)$, where k is the maximum column sum of A ($k \leqslant n$). Notice that CSP can be solved in polynomial time whenever $k \leqslant 2$ (using a maximum matching algorithm in a bipartite graph) and that k-CSP have slightly better heuristics using semi-local optimization [11]. An algorithm is *precision-sensitive* if its performance ratio does not depend on the input size but rather on the size of any optimal solution (greedy algorithm is not precision-sensitive). Hochbaum [19] proposed a $\mathrm{O}(n^2 v)$-time algorithm with a cover set of size at most $c^*(\mathscr{S}) f$, where f is the maximum row sum of A, using an analytical Russian method and a linear program relaxation. Interestingly, Feige [13] showed that no polynomial-time algorithm can approximate the optimal solution within a factor of $(1-\varepsilon)\log|\mathscr{S}|$, unless $\mathrm{NP} \subseteq \mathrm{DTIME}[n^{\log\log n}]$, where $1 > \varepsilon > 0$ (even when $c^*(\mathscr{S}) = \mathrm{O}(1)$). Viewing incidence matrix A as an hypergraph, we have

$$c^*(\mathscr{S}) \leqslant 11\lambda(\mathscr{S})^2(\lambda(\mathscr{S}) + b^*(\mathscr{S}) + 3)\binom{\lambda(\mathscr{S}) + b^*(\mathscr{S})}{b^*(\mathscr{S})},$$

where $b^*(\mathscr{S})$ is the maximum size of a stable, and $\lambda(\mathscr{S})$ is the maximum l so that the incidence matrix A has as a submatrix the incidence matrix of the complete graph $\mathscr{K}_l$ (clique of size l) [10].

One major drawback from the computational geometry point of view is that these methods do not consider geometrical objects nor their shapes. (Although it has been shown that the intersection graph [2] of d-dimensional convex objects can be arbitrary

[1] More precisely, Fowler et al. [15] showed that covering a set of points with fixed-size squares (the so-called BOX COVER problem) is NP-complete as soon as $d > 1$.

[2] The intersection graph of a set of objects is defined as follows: we associate to each object a node and there exists an edge between two nodes iff the corresponding objects intersect.

[38] as soon as $d\geqslant 3$.) This means that we have to supply matrix A. (Thus, the size of the dense matrix is $O(n^d)\times n$ and these algorithms require $\Omega(n^{d+1})$ time and $\Omega(n^{d+1})$ space; For sparse incidence matrix, list-like datastructures may be used to improve running times.) One way to code geometric instances is to consider from the whole arrangement of the constant-size descriptive objects all the sets induced by k-faces, $0\leqslant k\leqslant d$. We label each k-face with the set of objects fully containing it. A label is said *maximal* if it is not included in another one. We remove non-maximal labels and obtain a so-called *Sperner system*, still possibly [27] of size $O(n^d)$. Hochbaum and Maass [20] considered the case of geometrical objects and gave a *polynomial-time approximation scheme*. Their algorithm allows us to consider sets of congruent star-shaped centrally symmetric objects T (see Fig. 1, piercing thinline shapes with black-square points), or dually covering sets of points with star-shaped translates $T^*=T$ (see Fig. 1, covering black-disk points with thickline shapes). In that context, piercing families of congruent boxes is of particular interest since it corresponds to covering a set of points by a minimum number of congruent copies of the centrally symmetric box. Brönnimann and Goodrich [7] investigate these problems using the concept of the *Vapnik–Červonenkis dimension*; Let $\mathscr{S}=(\mathscr{X},\mathscr{R})$ be a set system, i.e., a set $\mathscr{X}$ of elements and a collection $\mathscr{R}$ of ranges: elements of $\mathscr{X}$. Denote by $\mathscr{S}_{|Y}$ the set system induced by Y by restricting each range $R\in\mathscr{R}$ to elements of Y. Set Y is said *shattered* if $\mathscr{R}_{|Y}=2^Y$. The VC-dimension is the maximum cardinality of a shattered set (see also the concept of VC-exponent [7]). The VC-dimension of the set system $\mathscr{S}=(\mathbb{E}^d,\mathscr{R})$ of d-dimensional isothetic boxes $\mathscr{R}$ and points of $\mathbb{E}^d$ is 2. They gave an iterative algorithm whenever the VC-dimension is bounded as it is generally the case when considering geometric scenes. Their weight-based algorithm uses subroutines such as set systems, ε-nets,[3] net-finders, etc. (see also [30]) and still relies on the fact that matrix A is (partially) computed beforehand (a selecting problem). Let $\mathscr{X}$ be our set of geometric objects and $\mathscr{R}$ be the set of labeled cells of the arrangement of objects. Then, a *hitting set* of $(\mathscr{X},\mathscr{R})$ is a geometric covering set, i.e., a subset of objects that covers all the cells. Dually, a *covering set* is a collection of ranges (cells) that pierces every geometric object.

Recently, Efrat et al. [12] studied dynamic datastructures for fat objects and obtain efficient piercing algorithms under the fatness assumptions in dimension 2 and 3. As a byproduct, we have $c^*(\mathscr{S})=O(b^*(\mathscr{S}))$ for fat objects in arbitrary fixed dimension. Solving exactly the problem (i.e., reporting an optimal solution) has also been investigated by Hwang et al. [22] for disks and recently for convex polygonal shapes by Assa and Katz [4].

In this paper, we are even more restrictive by considering the case of axis-parallel boxes in high dimensions (that are often considered in VLSI design, image processing and point location in d-dimensional Euclidean space) which has also received

[3] An ε-net of a set system $\mathscr{S}=(\mathscr{X},\mathscr{R})$ is a subset $\mathscr{X}'$ of $\mathscr{X}$ so that each range of at least $\varepsilon|\mathscr{X}|$ elements contains at least one element of $\mathscr{X}'$.

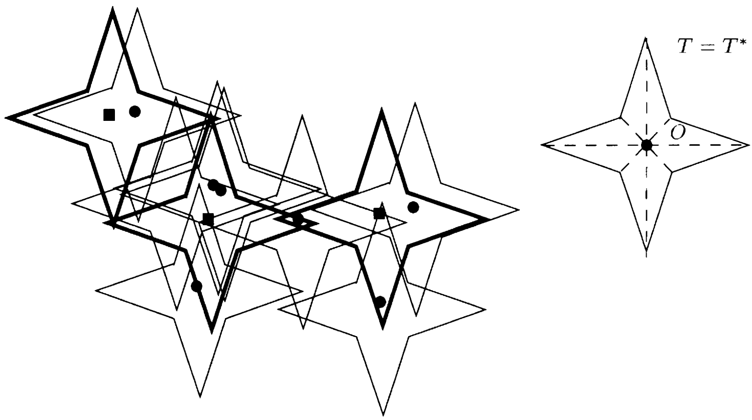

Fig. 1. Duality between piercing a set of translates of a centrally symmetric object T and covering points with copies of $T^* = T$.

combinatorial attention in [14, 24, 25, 29, 31]; As an application, suppose we are given a set of points in $\mathbb{E}^d$ and some hypercube H_d. We want to associate to each point a hypercube that contains it so that we minimize the number of hypercubes. In other words, we want to cover the point set with a minimum number of patches, i.e., translates of H_d. Throughout the paper, the boxes are considered to be closed, i.e., points on the boundary of box B stab B. Our main algorithm, described in Section 3, will not require to compute the arrangement of the isothetic boxes.[4] In the sequel, we do not consider d as a constant. Let $c^*(\mathscr{S})$, $b^*(\mathscr{S})$ and $c(\mathscr{S})$ denote, respectively, the minimum number of piercing points of $\mathscr{S}$, the maximum number of pairwise disjoint boxes of $\mathscr{S}$, and the number of points returned by our heuristic on set $\mathscr{S}$. Whenever appropriate, we will use b^*, c^* and c as shortcuts for $b^*(\mathscr{S}), c^*(\mathscr{S})$ and $c(\mathscr{S})$.

We give in this paper a simple algorithm and study its approximation factor.[5] This algorithm may be useful in many applications due to its speed (a Java applet[6] demonstrates it). More precisely, we give a truly output-sensitive $O(dn \log c)$-time algorithm that computes a set of c points stabbing the set of n d-dimensional boxes. Interestingly, we show that $c \leqslant b^*(1 + \log_2 b^*)^{d-1}$ where $b^* \leqslant c^*$ (c^* is the optimal value). Moreover, we exhibit a generic example where this bound is asymptotically matched. We can refine the complexity analysis to show that $c \leqslant 2^{d-1} b^*$ and $c = O_d(c^*)$ when dealing, respectively, with congruent isothetic boxes and *constrained* boxes.

The paper is organized as follows:

In section 2, we consider the case of a family of n intervals and give an optimal $\Theta(n \log c^*)$-time algorithm that gives an optimal stabbing set of c^* points. In

---

[4] Computing the arrangement of a set of n isothetic boxes costs $O(n^d)$ time and space [35].

[5] α is an *approximation factor* of an algorithm A if $c \leqslant \alpha c^*$ where c^* is the optimal value and c is the value delivered by heuristic A.

[6] See URL http://www.inria.fr/prisme/personel/nielsen/ or alternatively URL http://www.csl.sony.co.jp/person/nielsen/

Section 3, we present the multidimensional divide-and-conquer algorithm and study both its running time and its approximation factor. We show that the given bounds are tight. We refine the analysis for sets of congruent isothetic and constrained isothetic boxes. We corroborate our theoretical results with experiments. Finally, in Section 4, we conclude and give several guidelines for future research.

2. An optimal algorithm for stabbing intervals

In this section, we consider the case of intervals, i.e., one-dimensional boxes. Let $\mathcal{S}$ be a set of n intervals.

2.1. Principle

Finding the minimum value c^* so that $\mathcal{S}$ can be stabbed with c^* points is easy and already known in [9, 20] although no explicit efficient algorithm is reported. Consider the interval I that has the rightmost left endpoint p. I must be stabbed by a point and clearly, the best place to stab it is on the left endpoint p. We then remove all the intervals stabbed by p and loop until all the intervals of $\mathcal{S}$ are stabbed. We thus obtain a minimal-size set of $c^*(\mathcal{S})$ points that stab $\mathcal{S}$. A straightforward algorithm based on these facts has running time $O(nc^*(\mathcal{S}))$ with linear space. We show below how to get an optimal output-sensitive algorithm in time $\Theta(n \log c^*(\mathcal{S}))$.

Remark 1. In dimension 2 (and therefore in higher dimensions), the rectangle R with the rightmost left edge is not necessary the one that has the topmost bottom edge so that we cannot exhibit a rectangle R where we can easily a priori compute the best place to pierce it.

Remark 2. Let $b^*(\mathcal{S})$ be the maximum size of any subset of pairwise disjoint boxes of $\mathcal{S}$. Clearly, $c^*(\mathcal{S}) \geqslant b^*(\mathcal{S})$. In the one-dimensional case, the above algorithm shows that $c^*(\mathcal{S}) = b^*(\mathcal{S})$. This property is no longer true in higher dimensions. (For example, in dimension 2, we may have $c^*(\mathcal{S}) \geqslant \frac{3}{2} b^*(\mathcal{S})$ with $|\mathcal{S}| = 5$, and $c^*(\mathcal{S}) \geqslant \lfloor 5b^*(\mathcal{S})/3 \rfloor$ with $|\mathcal{S}| = 23$. It is an open problem to find a *worse* set $\mathcal{S}$ for which $c^*(\mathcal{S}) > \lfloor 5b^*(\mathcal{S})/3 \rfloor$.)

2.2. Getting an output-sensitive algorithm

The methodology consists in applying the divide-and-conquer strategy to the set $\mathcal{S} = \{[a_1, b_1], \ldots, [a_n, b_n]\}$ of n intervals. However, it differs from the traditional divide-and-conquer by filtering the recursive subsets. Let **Divide-and-Conquer**($\mathcal{S}$) be the algorithm which returns an optimal stabbing point set $\mathcal{P}$ of $\mathcal{S}$.

Base case: If $n = 1$ then return $\mathcal{P} = \{a_1\}$.

Partition: Computer the median m of $\{a_1, \ldots, a_n\}$. Let $\mathcal{S}_r = \{[a_i, b_i] \mid a_i \geqslant m\}$ and $\mathcal{S}_l = \mathcal{S} \backslash \mathcal{S}_r$.

Recurse: Let $\mathscr{P}_r=$ **Divide-and-Conquer**$(\mathscr{S}_r)$. Let $q=\min\mathscr{P}_r$ be the leftmost stabbing point of $\mathscr{S}_r$. Filter $\mathscr{S}_l$ by removing all intervals stabbed by q: $\mathscr{S}_l'=\{[a_i,b_i]\mid b_i<q\}$. Let $\mathscr{P}_l=$ **Divide-and-Conquer**$(\mathscr{S}_l')$. Return $\mathscr{P}=\mathscr{P}_r\cup\mathscr{P}_l$.

The algorithm reports an optimal stabbing point set $|\mathscr{P}|=c^*(\mathscr{S})=c^*$ because it stabs the intervals successively from right to left as explained in Section 2.1.

Let $t(n,c^*)$ denote the time complexity of the algorithm. We prove below that $t(n,c^*)\leqslant An(1+\log_2 c^*)$ for $1\leqslant c^*\leqslant n$.

Proof. We have

$$t(n,c^*)\leqslant\begin{cases}B & \text{if } n=1,\\ Cn+t(n/2,c^*(\mathscr{S}_r))+t(n/2,c^*(\mathscr{S}_l')) & \text{otherwise,}\end{cases}$$

where C depends on the partitioning and median finding algorithms [6].

If $n=1$ then $c^*=1$ and $t(n,c^*)\leqslant A$ for $A\geqslant B$. Otherwise, we apply the inductive hypothesis to both $\mathscr{S}_l'$ and $\mathscr{S}_r$ and get

$$t(n,c^*)\leqslant Cn+A\frac{n}{2}(1+\log_2 c^*(\mathscr{S}_l')+1+\log_2 c^*(\mathscr{S}_r)).$$

Using Hölder's inequality on the concave $\log_2(\cdot)$ function, we obtain

$$t(n,c^*)\leqslant Cn+A\frac{n}{2}\left(2+2\log_2\frac{c^*}{2}\right).$$

That is $t(n,c^*)\leqslant An(1+\log_2 c^*)$ for $A\geqslant C$.

The algorithm readily uses linear storage.

Since verifying if among n numbers k are distinct requires $\Omega(n\log k)$ time on the real RAM [28], it follows that this lower bound also holds for the stabbing problem by standard reduction in linear time. Therefore, we obtain the following theorem:

Theorem 3. *Given a set $\mathscr{S}$ of n intervals, there exists an optimal output-sensitive algorithm that reports an optimal stabbing point set of size c^* in optimal $\Theta(n\log c^*)$ time with linear space.*

Remark 4. As a direct consequence, we obtain a $\Theta(n\log c^*)$-time algorithm for computing the union of an n-interval set $\mathscr{S}$, where c^* is the minimal number of points required to stab $\mathscr{S}$. Note that it is not possible to get an $O(n\log C(\mathscr{S}))$-time algorithm for computing the union of intervals, where $C(\mathscr{S})$ is the number of connected components of $\mathscr{S}$. (We may have $c^*(\mathscr{S})=\lceil n/2\rceil$ but $C(\mathscr{S})=1$.) Indeed, we prove on the algebraic decision tree model that computing $\cup\mathscr{S}$ requires $\Omega(n\log n)$ time even when $C(\mathscr{S})=1$. Let $\mathscr{X}=\{x_1,\ldots,x_n\}$ be n real numbers. The UNIFORM GAP problem asks whether the difference between consecutive numbers are uniformly equal to ε, for some $\varepsilon>0$. UNIFORM GAP has a $\Omega(n\log n)$ lower bound on the algebraic decision tree model (see [35], p. 260). Consider the interval set $\mathscr{S}=\{[x_1-\varepsilon/2,x_1+\varepsilon/2],\ldots,[x_n-\varepsilon/2,x_n+\varepsilon/2]\}$ of intervals where $\varepsilon=(\max_i x_i-\min_i x_i)/n-1$. Observe that

$C(\mathscr{S}) = 1$ if and only if $\mathscr{X}$ is equally spaced. Therefore testing whether $C(\mathscr{S}) = 1$ or not is $\Omega(n \log n)$ hard. We conclude that computing the union of intervals is $\Omega(n \log n)$ hard even when considering parameter $C(\mathscr{S})$.

Remark 5. Although the greedy algorithm [8] performs generally a H_n factor from the optimum ($H_n = \sum_{i=1}^{n} 1/i$ and $H_n \leqslant 1 + \log n$), for the case of intervals we show below that it will return at most $2c^* - 1$ points. Moreover, there is a family of intervals where the greedy algorithm attains this worst-case bound.

Proof. Let $\mathscr{I}$ be a set of intervals and $\mathscr{Q}$ an optimal set of points piercing $\mathscr{I}$. Consider two adjacent points (abscissæ) Q_1 and Q_2 of $\mathscr{Q}$. Then, the greedy algorithm will pick at most one point in the range $]Q_1, Q_2[$. Indeed, once it has picked one point Q in $]Q_1, Q_2[$ and removed all the intervals stabbed by Q, all the intervals intersecting $]Q_1, Q_2[$ intersect either Q_1 or Q_2 and therefore the maximal cells are located at the extremities. Therefore, the greedy algorithm will return at most $(c^*(\mathscr{I}) - 1) + c^*(\mathscr{I})$ points, i.e. $2c^*(\mathscr{I}) - 1$ points. However, in dimension $d > 1$, greedy algorithms may have a $\Omega(\log n)$ factor even for sets of hypercubes (see Fig. 3). □

3. The algorithm in higher dimensions

3.1. Principle

The ‘divide-and-conquer’ strategy holds in any dimension and for any kind of objects. We show in that section how we can get results on the approximation factor when dealing with axis-parallel boxes. Let $\mathscr{S}$ be a set of n d-dimensional boxes. A box B can be viewed as the ordered cartesian product $\prod_{i=1}^{d}[r_i^-(B), r_i^+(B)]$, where $[r_i^-(B), r_i^+(B)]$ is the range of B along the ith dimension. We say that box B is to the *left* (*right*) of hyperplane $H : (x_i = l)$ if $r_i^+(B) < l$ (resp. $r_i^-(B) > l$).

We describe below the 2-level recursion algorithm (see also Fig. 2):

Preprocessing($\mathscr{S}$). For dimension d, consider the set of intervals $\mathscr{S}^{(d)} = \{[r_d^-(B), r_d^+(B)],\ B \in \mathscr{S}\}$ obtained from the boxes and compute a sorted optimal stabbing set of points $P_d(\mathscr{S})$. (Apply the optimal algorithm of Section 2.2 for piercing this set of intervals.) Call **Divide-and-Conquer($\mathscr{S}$)**.

Divide-and-Conquer($\mathscr{S}$).

Intervals (*Base case*): If $\mathscr{S}$ is one-dimensional then return $P_1(\mathscr{S})$.

Partition: Select the value m of the median stabbing point of $P_d(\mathscr{S})$. Partition $\mathscr{S}$ according to the hyperplane $H_m : (x_d = m)$;

- Let $\mathscr{S}_1$ be the set of boxes that do not cross H_m and are to the left of H_m.
- Let $\mathscr{S}_2$ be the set of boxes that do not cross H_m and are to the right of H_m.
- Let $\mathscr{S}_m$ be the set of boxes intersecting H_m.

Conquest: If $P_d(\mathscr{S}_1) > 1$ then call **Divide-and-Conquer**($\mathscr{S}_1$).

If $P_d(\mathscr{S}_2) > 1$ then call **Divide-and-Conquer**($\mathscr{S}_2$).

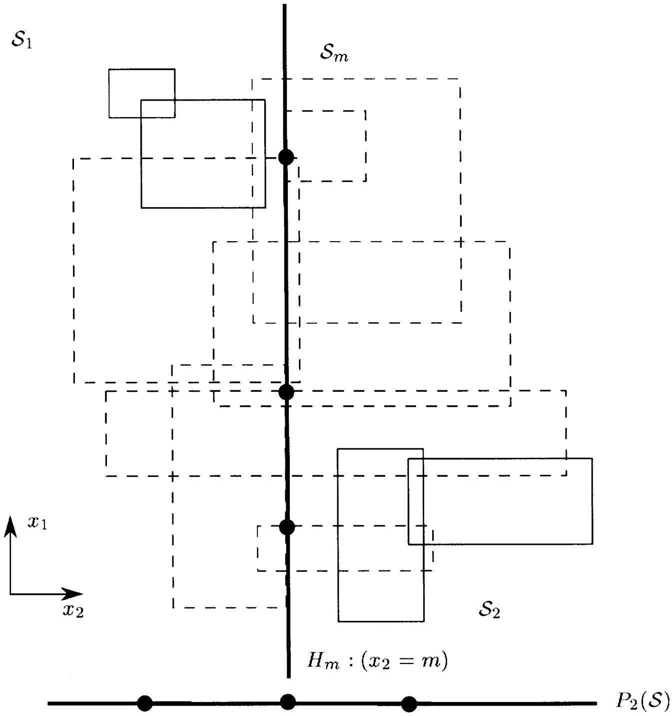

Fig. 2. Partition of $\mathcal{S}$ into three subsets depending on their location with respect to the hyperplane H_m: $(x_d = m)$. We denote by $\mathcal{S}'_m$ the set of $(d-1)$-dimensional boxes $\mathcal{S}_m \cap H_m$.

Recurse: Stab the boxes of $\mathcal{S}_m$ by piercing the set of $(d-1)$-dimensional boxes: $\mathcal{S}'_m = \{B \cap H_m \mid B \in \mathcal{S}_m\}$. Go to **Preprocessing($\mathcal{S}'_m$)**.

Let $t(\mathcal{S})$ and $c(\mathcal{S})$ be, respectively, the running time of the algorithm and the number of stabbing points delivered by this heuristic. Sometimes, when we want to specify the dimension d of $\mathcal{S}$, we put in subscript of these notations a d. Thus, $t_d(\mathcal{S})$ and $c_d(\mathcal{S})$ denote, respectively, the running time and the output size of our algorithm for a set of d-dimensional isothetic boxes S. We study both the approximation factor and the running time of the algorithm. Recall that d is not assumed to be a constant.

Our algorithm relies on the following three simple facts:

Monotonicity rule: For any object O, $c^*(\{O\} \cup \mathcal{S}) \geqslant c^*(\mathcal{S})$.

Additive rule: Let $\mathcal{I}_1$ and $\mathcal{I}_2$ be two subsets so that $\forall I_1 \in \mathcal{I}_1$, $\forall I_2 \in \mathcal{I}_2$, $I_1 \cap I_2 = \emptyset$ then $c^*(\mathcal{I}_1 \cup \mathcal{I}_2) = c^*(\mathcal{I}_1) + c^*(\mathcal{I}_2)$.

Cutting rule: Let $\mathcal{S}$ be a set of boxes and H a hyperplane perpendicular to the i-th axis, with $1 \leqslant i \leqslant d$. Then, $c^*(\mathcal{S}_H) = c^*(\mathcal{S}'_H)$ where $\mathcal{S}'_H = \{B \cap H \mid B \in \mathcal{S}\}$ and $\mathcal{S}_H = \{B \mid B \cap H \neq \emptyset\}$. This rule is appropriate only for isothetic-like d-boxes.

We have

$$c_d(\mathscr{S}) = \begin{cases} c_{d-1}(\mathscr{S}_m \cap H_m) & \text{if } \mathscr{S}_1 = \mathscr{S}_2 = \emptyset, \\ c_{d-1}(\mathscr{S}_m \cap H_m) + c_d(\mathscr{S}_1) + c_d(\mathscr{S}_2) & \text{otherwise.} \end{cases}$$

Let us prove by induction on the lexicographically ordered vector (d,n) that

$$b^*(\mathscr{S}) \leqslant c^*(\mathscr{S}) \leqslant c(\mathscr{S}) \leqslant b^*(\mathscr{S})(1+\log_2 b^*(\mathscr{S}))^{d-1}.$$

Proof. The inequality $b^*(\mathscr{S}) \leqslant c^*(\mathscr{S}) \leqslant c(\mathscr{S})$ is obtained straightforwardly from their respective definitions. We concentrate on proving that $c(\mathscr{S}) \leqslant b^*(\mathscr{S})$ $(1+\log_2 b^*(\mathscr{S}))^{d-1}$. We are going to prove a somewhat stronger result, namely

$$c(\mathscr{S}) \leqslant b^*(\mathscr{S})(1+\log_2 b^*(\mathscr{S}^{(d)}))(1+\log_2 b^*(\mathscr{S}))^{d-2},$$

where $\mathscr{S}^{(d)}$ is the set of intervals $\mathscr{S}^{(d)} = \{[r_d^-(B), r_d^+(B)],\ B \in \mathscr{S}\}$. Since $b^*(\mathscr{S}^{(d)}) \leqslant b^*(\mathscr{S})$, we will be able to conclude.

For $d=1$, section 2.2 describes an optimal algorithm so that $c(\mathscr{S}) = b^*(\mathscr{S}) \leqslant b^*(\mathscr{S})$ $(1+\log_2 b^*(\mathscr{S}))^0 = b^*(\mathscr{S})$. If $b^*(\mathscr{S}) = 1$ then all pairs mutually intersect and therefore $c(\mathscr{S}) = c^*(\mathscr{S}) = 1 \leqslant 1(1+0)^{d-1}$. (This may also be seen as a Helly-type number. See [9].)

Therefore, assume that $d \geqslant 2$ and $b^*(\mathscr{S}) > 1$.

If $\mathscr{S}_1 = \mathscr{S}_2 = \emptyset$ then we directly apply the recurrence inequality and get

$$c(\mathscr{S}) \leqslant b^*(\mathscr{S}'_m)(1+\log_2 b^*(\mathscr{S}'_m))^{d-2} \leqslant b^*(\mathscr{S})(1+\log_2 b^*(\mathscr{S}))^{d-1},$$

since $\mathscr{S}'_m = \mathscr{S}$.

Otherwise, either $\mathscr{S}_1$ or $\mathscr{S}_2$ is not empty. We have $b^*(\mathscr{S}_1^{(d)}) = \lfloor (b^*(\mathscr{S}^{(d)}) - 1)/2 \rfloor$ and $b^*(\mathscr{S}_2^{(d)}) = \lceil (b^*(\mathscr{S}^{(d)}) - 1)/2 \rceil$. Applying the inductive hypothesis with $b^*(\mathscr{S}^{(d)}) \leqslant b^*(\mathscr{S})$, we get

$$c(\mathscr{S}) \leqslant (b^*(\mathscr{S}) - 1)\left(1+\log_2\left\lceil \frac{b^*(\mathscr{S})-1}{2} \right\rceil\right)^{d-1} + b^*(\mathscr{S}'_m)(1+\log_2 b^*(\mathscr{S}'_m))^{d-2}.$$

Since $\lceil (b^*(\mathscr{S}) - 1)/2 \rceil \leqslant b^*(\mathscr{S})/2$ and $b^*(\mathscr{S}'_m) \leqslant b^*(\mathscr{S})$, we obtain

$$\begin{aligned} c(\mathscr{S}) &\leqslant b^*(\mathscr{S})(1+\log_2 b^*(\mathscr{S}))^{d-2}\left(1+\log_2 \frac{b^*(\mathscr{S})}{2}\right) \\ &\quad + b^*(\mathscr{S})(1+\log_2 b^*(\mathscr{S}))^{d-2}, \\ c(\mathscr{S}) &\leqslant b^*(\mathscr{S})(1+\log_2 b^*(\mathscr{S}))^{d-1}. \quad \square \end{aligned}$$

Remark 6. In fact, we proved above that $c(\mathscr{S}) \leqslant b^*(\mathscr{S}) \prod_{i=2}^{d} (1+\log_2 b^*(\mathscr{S}^{(i)}))$. Moreover, with a more careful analysis, the bound can be lowered to

$$c(\mathscr{S}) \leqslant b^*(\mathscr{S}) \log_2^{d-1} b^*(\mathscr{S}) + d - \tfrac{1}{2} b^*(\mathscr{S}) \log_2^{d-2} b^*(\mathscr{S}).$$

However, as detailed later, the asymptotic value $\Omega(b^*(\mathscr{S}) \log_2^{d-1} b^*(\mathscr{S}))$ is attained.

Let us now analyze the time $t_d(\mathscr{S})$ spent by this algorithm for reporting the $c_d(\mathscr{S})$ stabbing points.

We prove below by induction on the lexicographically ordered vector (d,n) that $t_d(\mathscr{S}) \leqslant Adn(1+\log_2 c(\mathscr{S}))$ $(\bigstar)$.

Proof. If $d=1$ then we proved in Section 2.2 an $O(n(1+\log c(\mathscr{S})))$-time algorithm. Therefore, $t_1(S) \leqslant Bn(1+\log_2 c(\mathscr{S})) \leqslant An(1+\log_2 c(\mathscr{S}))$ for $A \geqslant B$, where B depends on the implementation of the one-dimensional algorithm. If $n=1$ then $t_d(\mathscr{S}) \leqslant Ad$ and $(\bigstar)$ holds trivially.

Otherwise ($d>1$ and $n>1$), we can write $t_d(\mathscr{S})$ by decomposing it into two parts: (1) the costs of making all the $b^*(\mathscr{S}^{(d)})$ subsets partitioning $\mathscr{S}$ using $\mathscr{S}^{(d)}$, and (2) the costs of all recursive calls on the $(d-1)$-dimensional sets of boxes $\mathscr{S}'_i$. Therefore we get

$$t_d(\mathscr{S}) \leqslant Cn \log_2 c^*(\mathscr{S}^{(d)}) + \sum_{\mathscr{S}'_i} t_{d-1}(|\mathscr{S}'_i|).$$

By applying the inductive hypothesis to all sets $\mathscr{S}'_i$, we get

$$t_d(\mathscr{S}) \leqslant Cn \log_2 c^*(\mathscr{S}^{(d)}) + \sum_{\mathscr{S}'_i} A(d-1)|\mathscr{S}'_i| \log_2 c(\mathscr{S}'_i),$$

$$t_d(\mathscr{S}) \leqslant Cn \log_2 c^*(\mathscr{S}^{(d)}) + An(d-1)\log_2 c(\mathscr{S}) \leqslant Adn \log_2 c(\mathscr{S}),$$

for $A \geqslant C \geqslant B$. $\square$

Note that when $c(\mathscr{S}) = c^*(\mathscr{S}) = 1$ then our algorithm requires $O(nd)$-time (this is easily checked). Let $N = O(nd)$ denote the input size of our set $\mathscr{S}$ of n d-dimensional boxes. Then, our algorithm runs in $O(N \log c)$-time and linear space.

Theorem 7. *Let $\mathscr{S}$ be a set of n d-dimensional boxes. Denote by b^* the maximum number of pairwise disjoint boxes of $\mathscr{S}$. Then, there exists an output-sensitive algorithm that reports a set of c stabbing points of $\mathscr{S}$ in time $O(dn \log c)$ with linear space whose approximation c is bounded by $c \leqslant b^*(1+\log_2 b^*)^{d-1}$.*

We prove below that the greedy strategy has a $\Omega(\log_2 n)$ approximation ratio on sets of boxes.

Theorem 8. *As soon as we consider d-boxes with $d \geqslant 2$, greedy algorithm* [8] *may yield a $\Omega(\log_2 n)$ factor from the optimum.*

Proof. We build a family $\mathscr{A}$ such that $c^*(\mathscr{A})=2$ of $n=2^{k+2}-2$ squares that is decomposed into two sub-families $\mathscr{A}_1$ and $\mathscr{A}_2$ of $2^{k+1}-1$ squares each such that both $c^*(\mathscr{A}_1)=1$ and $c^*(\mathscr{A}_2)=1$. A planar box $B=[a,b]$ is defined by its leftmost bottommost corner $a=(x_a,y_a)$ and its rightmost uppermost corner $b=(x_b,y_b)$.

$\mathscr{A}_1$ is defined by 2^0 square $[(-k,0),(0,k)], \ldots, 2^i$ squares $[(-k-i,-i),(-i,k-i)]$, $\ldots$, and 2^k squares $[(-2k,-k),(-k,0)]$. Similarly, $\mathscr{A}_2$ is defined by 2^0 square $[(0,-k),$

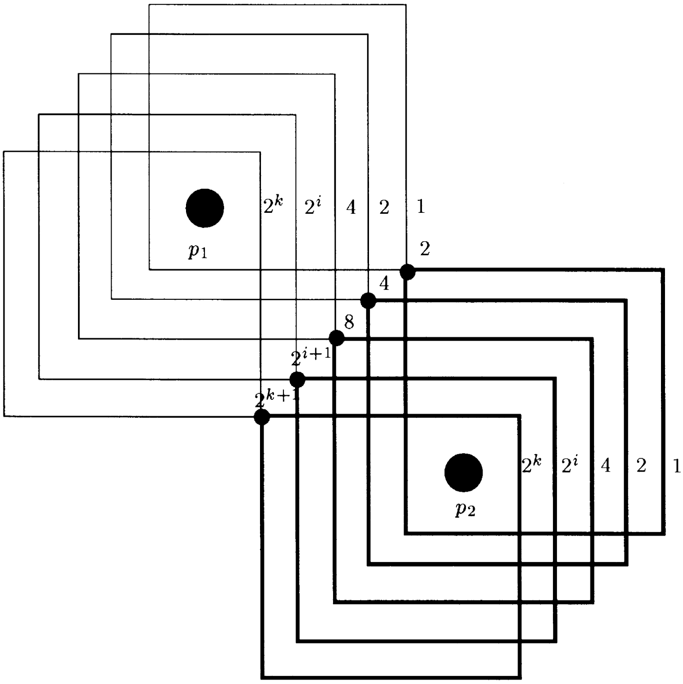

Fig. 3. Building a family $\mathscr{A}$ such that the greedy algorithm returns $\Omega(\log n)c^*(\mathscr{A})$ points.

$(k,0)],\ldots,2^i$ squares $[(-i,-k-i),(k-i,-i)],\ldots,$ and 2^k squares $[(-k,-2k),(0,-k)]$ (see Fig. 3). Clearly, the greedy algorithm will return $k+1$ points $(-k,-k),\ldots,(-k+i,-k+i),\ldots$ and $(0,0)$ stabbing $\mathscr{A}$ instead of an optimal solution of two points (e.g., p_1 and p_2 in Fig. 3). □

3.2. A bad example

Let us now analyze the tightness of the upper bound of our heuristic. In the d-dimensional Euclidean space, consider the following k families $\mathscr{F}_i, 1 \leqslant i \leqslant k$, of isothetic boxes: $\mathscr{F}_i$ is the set of all boxes with corners of integer coordinates in $[1,k]$ containing the lattice point $(i,\ldots,i)$. (We include in $\mathscr{F}_i$ the degenerate box defined by point $(i,\ldots,i)$.) Let $\mathscr{S} = \bigcup_{1 \leqslant i \leqslant k} \mathscr{F}_i$. Clearly $b^*(\mathscr{S}) = c^*(\mathscr{S}) = k$. However, our algorithm will report $\Omega(b^*(\mathscr{S}) \log^{d-1} b^*(\mathscr{S}))$ stabbing points. Thus, we proved the tightness of the approximation factor of the multidimensional heuristic.

3.3. Congruent or constrained boxes

The previous section exhibits an example where our algorithm reaches its worst-case performance. However, in order to build it, we did consider stretched boxes, i.e., non-constrained boxes. We define the aspect ratio of a d-dimensional box $B = [(a_1,\ldots,a_d),(b_1,\ldots,b_d)]$ to be $\max_{i,j=1..d}\{(b_i - a_i)/(b_j - a_j)\}$. Loosely speaking, we will call aspect ratio of $\mathscr{S}$ the maximum of the aspect ratio of the boxes in $\mathscr{S}$. Note that if the boxes have a bounded aspect ratio then they also have a bounded-size (volume) but not the converse. Thus, hypercubes have aspect ratio equal to 1. We use the non-bounded aspect ratio in order to build our bad example. In [20], the polynomial-time approximation scheme depends on the aspect ratio of the congruent isothetic boxes.

In this section, we refine the analysis of the approximation factor whenever the projections of the d-boxes onto the d-axis have the bounded aspect ratio property. Note that the boxes may have a non-bounded aspect ratio but their projections (sets of intervals) may have their aspect ratio bounded.

Lemma 9. *Let $\mathscr{S}$ be a set of n congruent isothetic d-dimensional boxes. Then, our algorithm guarantees that $c(\mathscr{S}) \leqslant 2^{d-1} b^*(\mathscr{S})$.*

Proof. For sake of simplicity, let us first consider the case of (unit) hypercubes. $\mathscr{S}$ is a collection of n congruent hypercubes. We prove below by induction on the dimension that $c(\mathscr{S}) \leqslant 2^{d-1} b^*(\mathscr{S})$. Section 2.2 shows that the algorithm ensures $c_1(\mathscr{S}) = b^*(\mathscr{S})$ for (unit) intervals.

Otherwise ($d \geqslant 2$), let $\mathscr{S}$ be a set of n d-boxes. Consider the ordered sequence (left to right) of cutting hyperplanes perpendicular to the dth-axis: $(H_m(1)\colon x_d = a_1), \ldots, (H_m(k)\colon x_d = a_k)$ with the associated partition of the hypercubes $\mathscr{S}'_m(1), \ldots, \mathscr{S}'_m(k)$. Clearly, we have $a_{i+2} - a_i > 1, 1 \leqslant i \leqslant k-1$ for the case of unit hypercubes. Therefore $\mathscr{S}'_m(i) \cap \mathscr{S}'_m(j) = \emptyset$ if $|i-j| \geqslant 2$. We have

$$c_d(\mathscr{S}) = \sum_{i=1}^{k} c_{d-1}(\mathscr{S}'_m(i)),$$

$$c_d(\mathscr{S}) \leqslant 2^{d-2} \sum_{i=1}^{k} b^*(\mathscr{S}'_m(i)).$$

We can decompose the last sum taking into account the parity of i as follows:

$$c_d(\mathscr{S}) \leqslant 2^{d-2} \left(\sum_{i=1}^{\lfloor k/2 \rfloor} b^*(\mathscr{S}'_m(2i)) + \sum_{i=0}^{\lceil k/2 \rceil - 1} b^*(\mathscr{S}'_m(2i+1)) \right).$$

But $\sum_{i=1}^{\lfloor k/2 \rfloor} b^*(\mathscr{S}'_m(2i)) = b^*(\bigcup_{i=1..\lfloor k/2 \rfloor} \mathscr{S}'_m(2i)) \leqslant b^*(\mathscr{S})$ and $\sum_{i=0}^{\lceil k/2 \rceil - 1} b^*(\mathscr{S}'_m(2i+1)) = b^*(\bigcup_{i=0..\lceil k/2 \rceil - 1} \mathscr{S}'_m(2i+1)) \leqslant b^*(\mathscr{S})$ since both $S'_m(2i) \cap S'_m(2j) = \emptyset$ and $S'_m(2i+1) \cap S'_m(2j+1) = \emptyset$ as soon as $i \neq j$.

Therefore, we get

$$c_d(\mathscr{S}) \leqslant 2^{d-2} \times 2 b^*(\mathscr{S}),$$

$$c_d(\mathscr{S}) \leqslant 2^{d-1} b^*(\mathscr{S}).$$

We only use the fact that all boxes have identical widths along the ith dimension, $1 \leqslant i \leqslant d$. Therefore, the result applies for congruent boxes. □

Hochbaum and Maass [20] also considered this problem (in its dual form however) and gave an $O(l^d n^{2l^d+1})$-time algorithm (a polynomial-time approximation scheme) which ensures that $c(\mathcal{S}) \leqslant (1+1/l)^d c^*(\mathcal{S})$ for a given integer $l \geqslant 1$. Thus, for $l=1$ it yields an $O(n^3)$-time algorithm with performance ratio 2^d.

Since our algorithm proceeds dimension by dimension, we only need to have the bounded aspect ratio for the projected boxes (along the d-axis). Let $\mathcal{S}$ be a collection of n constrained boxes:

$$\mathcal{S} = \{[(b_{1,1},\dots,b_{1,d}),(u_{1,1},\dots,u_{1,d})],\dots,[(b_{n,1},\dots,b_{n,d}),(u_{n,1},\dots,u_{n,d})]\}$$

with $\max_{i,j=1..n}\{(u_{i,1}-b_{i,1})/(u_{j,1}-b_{j,1})\} \leqslant B_1,\dots,\max_{i,j=1..n}\{(u_{i,d}-b_{i,d})/(u_{j,d}-b_{j,d})\} \leqslant B_d$, for some constants $B_i \geqslant 1, 1 \leqslant 1 \leqslant d$.

Using the same techniques as above, we get the following lemma:

Lemma 10. *Let $\mathcal{S}$ be a collection of n d-dimensional constrained boxes with $B_1,\dots,B_d$ defined as above. Then, our algorithm will return $c(\mathcal{S})$ stabbing points so that $c(\mathcal{S}) \leqslant (\prod_{i=2}^{d} 2\lceil B_i \rceil + 1) b^*(\mathcal{S})$.*

We may assume w.l.o.g. that $B_1 = \max_{i=1..d}\{B_i\}$. Otherwise, we make a simple rotation of the orthogonal frame in linear time. This also means that we may have a direction where the projected boxes are not constrained since we are able to solve exactly the problem in one dimension (see Section 2.2).

Remark 11. In the case of congruent boxes, an alternative heuristic is to consider the dual problem of covering n points by unit hypercubes. If we allow the floor function, then in $O(dn \log c)$-time, by an 'output-sensitive' variant of Quicksort, we can compute the c cell grids that contain at least one point. Since any unit hypercube intersects at most 2^d grid cells, we conclude that $c \leqslant 2^d b^*$. (Moreover, by using the optimal 1d algorithm, we get $c \leqslant 2^{d-1} b^*$.)

Remark 12. Let $\mathcal{C}$ be a family of n planar convex objects. An object O is said α-fat if the ratio of the length of the edges of a smaller enclosing hypercube and a bigger included hypercube is bounded by α. If $\mathcal{C}$ is $O(1)$-fat and objects have more or less the same size, then we can slightly modify our heuristic in order to report at most $O_\alpha(1)^d b^*$ stabbing points. Indeed, we first pierce all enclosing hypercubes. Then for each stabbing point and its associated set of stabbed boxes, we stab the objects included in the enclosing hypercubes with only $O_\alpha(1)^d$ grid-spaced points. If objects are allowed to have a broad range of sizes, this method does not apply. Efrat et al. proposed another approach for general fat convex objects [12].

We can also mix up our algorithm with the PTAS (polynomial-time approximation scheme) of [20] in order to obtain tradeoffs both for the running time and the performance ratio. (We refer the reader to [1] for such an approach.)

Note that unless P = NP, it is not possible to achieve the absolute performance $c(\mathscr{S}) \leqslant c^*(\mathscr{S}) + k$ for some constant integer k [16]. Let $\mathscr{G}$ be the intersection graph of a set of n d-dimensional isothetic boxes $\mathscr{S}$, i.e., to each box corresponds a node and there is an edge between two nodes iff their corresponding boxes intersect. Isothetic boxes have nice combinatorial properties. For example, a set of boxes have a nonempty intersection iff they intersect pairwise. (This is a Helly-type theorem [9, 18].) Therefore, finding a minimum-size set of stabbing points can be done by first computing $\mathscr{G}$ in quadratic time and then, finding a minimum clique partition of $\mathscr{G}$, i.e., a set of cliques (complete subgraphs of $\mathscr{G}$) whose union covers all vertices of $\mathscr{G}$. There is a one-to-one correspondence between these two problems. As a direct corollary, it implies that the mimimum clique partition is NP-complete (as proved in [16]), even for intersection graphs of isothetic boxes. Note that if every 3-subset of $\mathscr{S}$ (i.e., a subset $S \in \binom{\mathscr{S}}{3}$) has an empty intersection ($\cap S = \emptyset$), then the stabbing problem can be solved in *polynomial time* by reduction to the maximum matching problem. The stabbing problem is related somehow to $\mathfrak{C}(p,q)$-numbers and $\mathfrak{N}(p,q;d)$ numbers [17, 18].

$\mathfrak{N}(p,q;d)$ numbers are defined for $2 \leqslant q \leqslant p$ for the class $\mathscr{P}$ of isothetic d-boxes as follows: it is the smallest integer so that for every set $\mathscr{S} \in \mathscr{P}$ of parallelotopes, we have: if every p-subset $S \in \binom{\mathscr{S}}{p}$ contains at least one q-subset $Q \in \binom{S}{p}$ that has a nonempty intersection ($\cap Q \neq \emptyset$), then $\mathscr{S}$ can be stabbed by $\mathfrak{N}(p,q;d)$ points. Debrunner et al. [17, 18] proved that $\mathfrak{N}(p,q;d) \leqslant \binom{p-q+d}{d}$ if $2 \leqslant q \leqslant p$.

Therefore, we have

$$c^* \leqslant \mathfrak{N}(b^* + 1, 2; d) = \binom{b^* - 1 + d}{d},$$

since every $(b^* + 1)$-subset of $\mathscr{S}$ contains at least a pair of intersecting boxes. In this paper, our heuritic shows that $c^* \leqslant b^*(1 + \log_2 b^*)^{d-1}$.

Perhaps, one of the most challenging conjectures that remains unsettled is the following:

Conjecture 13 (Wegner, 1967). *If $\mathscr{K}$ is a family of parallel rectangles in the plane, no p of which are pairwise disjoint, then $\mathscr{K}$ can be stabbed by $2p - 3$ points.*

This conjecture has been confirmed for squares and $p \leqslant 4$ but remain opened for nonconstrained boxes, i.e., nonbounded size boxes.

3.4. Computing a pairwise nonintersecting set of boxes

Given a graph $\mathscr{G} = (V, E)$, an independent set I of $\mathscr{G}$ is a set of nodes $I \subseteq V$ such that there is no edge of E between two nodes of I. Therefore, a maximal independent set of the intersection graph of the boxes corresponds to a maximal set of pairwise nonintersecting boxes. Although finding a maximal independent set on general abstract graph $\mathscr{G}$ has been shown to be $\Omega(n^{1/4})$-hard to approximate [5] in polynomial time, we describe below a $1/[\log_2^{d-1}(b^*(\mathcal{S})+1)]$ approximation heuristic based on the same partitioning scheme.

Let $b(\mathcal{S})$ be the size of an independent set found by some heuristic. Then, we have

$$b(\mathcal{S}) \leqslant b^*(\mathcal{S}) \leqslant c^*(\mathcal{S}) \leqslant c(\mathcal{S}).$$

On the other hand, we previously showed that $c(\mathcal{S}) \leqslant b^*(\mathcal{S})(1+\log_2 b^*(\mathcal{S}))^{d-1}$. More precisely, we proved

$$c(\mathcal{S}) \leqslant \max_i \{b_i\}(1+\log_2 b^*(\mathcal{S}))^{d-1},$$

where the b_i's are the maximal number of pairwise disjoint intervals of projected subsets of $\mathcal{S}$ induced by the partitioning scheme. A simple heuristic consists in giving an independent set of objects of $\mathcal{S}$ from the independent set of projected intervals which has the maximal size. By doing that, we get

$$b(\mathcal{S}) = \max_i \{b_i\} \geqslant \frac{b^*(\mathcal{S})}{(1+\log_2 b^*(\mathcal{S}))^{d-1}}.$$

Remark 14. A better heuristic, in practice, can be modeled as a binary tree where the root is set $\mathcal{S}$ and children correspond to recursive calls on subsets generated by partitioning the objects of $\mathcal{S}$. Each internal node $\mathcal{S}'$ has at most two children labeled with sets $\mathcal{S}_l$ and $\mathcal{S}_r$. We compute recursively in a bottom-to-top fashion an independent set of $\mathcal{S}$ by choosing either, at some node $\mathcal{S}$, the maximal independent set of $\mathcal{S}'$ or the union of the independent sets computed so far of sets $\mathcal{S}_l$ and $\mathcal{S}_r$.

3.5. Experimental results

We did the implementation in C++ using the LEDA [7] and GALIA [8] librairies. The code length is about 1000 lines. It should be noted that the algorithm and therefore its implementation are robust. Indeed, we only compare our standard input values without creating intermediate values (even when our algorithm reports intersection points). The algebraic degree of all test predicates is one. Since the problem of computing the minimum value c^* so that our set is c^*-pierceable is NP-complete, we could not compare in the experiments the precise relationships between c and c^*.

However, loosely speaking, if we admit that in average $c^*(\mathcal{S}'_m) = c^*(\mathcal{S})^{(d-1)/d}$ and $c^*(\mathcal{S}_1) = c^*(\mathcal{S}_2)$ (which might be the case, for example, when considering the uniform distribution of congruent boxes) then we expect $c(\mathcal{S})$ to be an O(1)-approximate

[7] Library for Efficient Datastructure and Algorithms. Max-Planck Institut für Informatik, Im Stadtwald 66123 Saarbrücken, Germany.

[8] GALIA is the european successor of CGAL – see "The CGAL Kernel User Manual" – INRIA Sophia-Antipolis, France.

Table 1
Considering uniformly distributed rectangles in the unit square

$\vert\mathcal{S}\vert =$	16	32	64	128	256	512	1024	2048	4096	8192	16364	32728	65536	130000	260000	520000
$c(\mathcal{S}) =$	9	14	23	30	56	78	123	188	272	417	648	928	1413	2093	3122	4486

Table 2
Number of different configurations conf($\mathcal{S}$) for some input set $\mathcal{S}$, i.e. number of distinct piercing point sets of size ranging over $[c^*, n]$ for small values of n. Note that for d-dimensional set $\mathcal{S}$ of boxes, the complexity of the arrangement of $\mathcal{S}$ is $O(|\mathcal{S}|^d)$. Thus, we have $\mathrm{conf}(\mathcal{S}) \leqslant \sum_{i=c^*(\mathcal{S})}^{|\mathcal{S}|} \binom{O(|\mathcal{S}|^d)}{i} \leqslant 2^{O(|\mathcal{S}|^d)}$

$\vert\mathcal{S}\vert$	=	14	15	16	17	18
conf($\mathcal{S}$)	=	15120	29400	99120	241272	672964

of $c^*(\mathcal{S})$ for fixed dimension d. Indeed, let K_d (with $K_1 = 1$) be the multiplicative constant. We have

$$c(\mathcal{S}) \leqslant K_d(c^*(\mathcal{S})) \leqslant c^* + \sum_{i=0}^{\lceil \log c^*(\mathcal{S}) \rceil} 2^i K_{d-1}\left(\left(\frac{c^*(\mathcal{S})}{2^i}\right)^{(d-1)/d}\right).$$

Therefore, we find $K_d \leqslant 1 + (1/(2^{1/d-1}))K_{d-1}$ with $K_1 = 1$. Thus, in the planar case we get under these hypothesis that $c(\mathcal{S}) \leqslant 3.42c^*(\mathcal{S})$. (This result is corroborated in the experiments using a good lower bound for $c^*(\mathcal{S})$, e.g., $b(\mathcal{S})$.)

We report the value of $c(\mathcal{S})$ in Table 1 for a set $\mathcal{S}$ of uniformly distributed isothetic planar boxes. We have also implemented an exhaustive search procedure for finding a minimal set of piercing points. This algorithm could not handle input size greater than 25 (although some tricks have been plugged [9] in). Table 2 shows the number of configurations explored by the exhaustive algorithm.

Fig. 5 shows some experiments for sets $\mathcal{S}$ that are 20-pierceable, i.e., $c^*(\mathcal{S}) = 20$. The left chart shows the number of stabbing points reported by our algorithm in case of congruent/nonconstrained boxes. The right chart depicts the running time of our algorithm. (We took the average over 20 trials.) Fig. 4 shows the result of our algorithm on 200 boxes.

It may be nice to study the average stabbing number $a_n(d)$ of a fixed-size randomly chosen set of n d-boxes as a function of d.

[9] Finding good cuts for an exhaustive algorithm is interesting in itself since it allows to handle large input sizes for 'special' down-to-earth tailored instances (see for example the well-known Traveling Salesman Problem [36]).

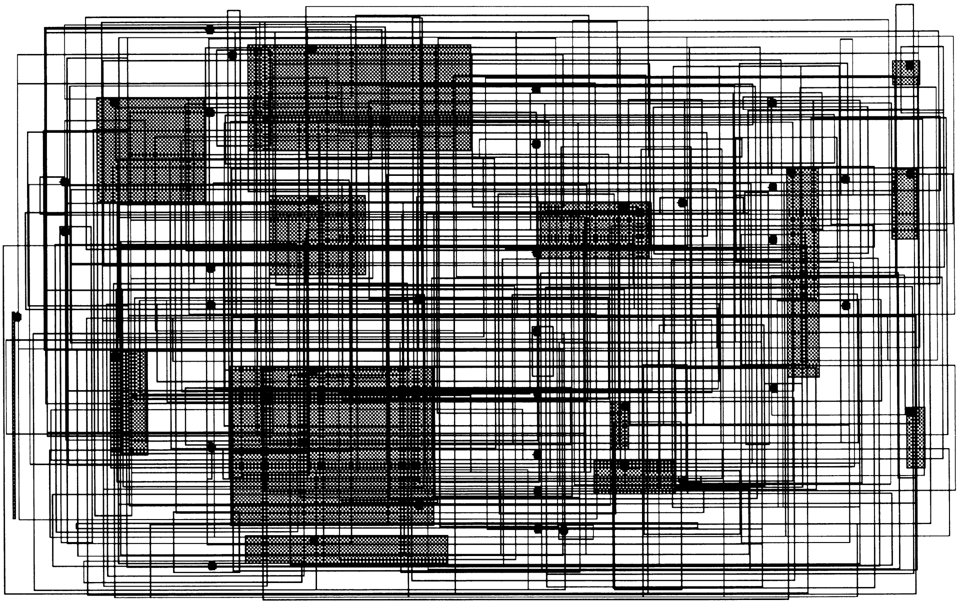

Fig. 4. A stabbing set of 42 points (black points) of a set of 200 boxes on the plane and a pairwise nonintersecting set of 14 boxes (grey boxes).

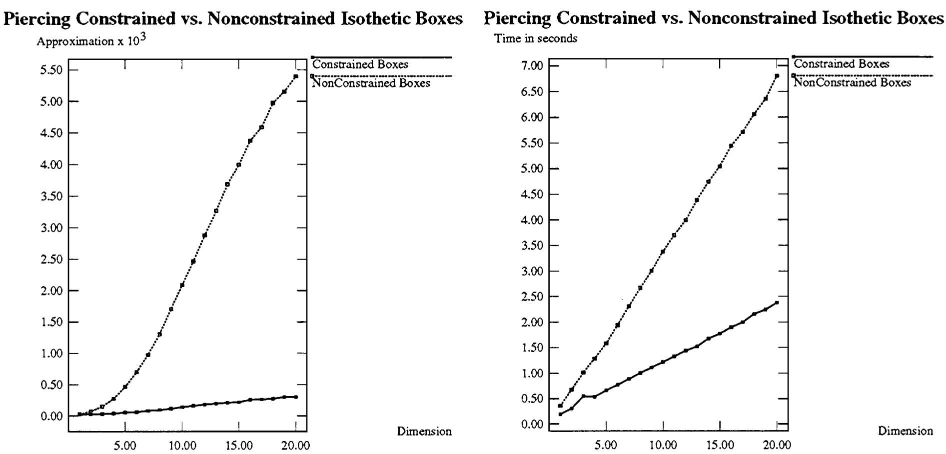

Fig. 5. Impact of the dimension over a set $\mathscr{S}$ ($c^*(\mathscr{S}) = 20$ and $|\mathscr{S}| = 20\,000$) of constrained/nonconstrained d-boxes for $1 \leqslant d \leqslant 20$. The right chart exhibits the running time of our implementation.

4. Concluding remarks

We have investigated in this paper the *stabbing problem* for a set of d-dimensional isothetic boxes which consists in finding a set of points so that each box contains at least one of these points. Finding a minimum-size set of stabbing points has been

Table 3

Class	Approximation factor $c(\mathscr{S})/c^*(\mathscr{S})$	Reference
Boxes	$(1+\log_2 b^*)^{d-1}$	This paper
Fat objects	$O_{d,\alpha}(1)$	[12]
k-oriented objects	$\mathfrak{G}(2,k)\log^{O_d(k^{d-1})}(b^*+1)$	$\mathfrak{G}$ is a Gallai number [32, 33]
Bounded VC-dim.	$O(\log c^*)$, $O(1)$	[7]
Abstract sets	$O(\log n)$	[8]

shown to be NP-complete as soon as $d>1$, even when considering congruent isothetic boxes. Therefore that problem is intractable in practice. We gave in this paper a heuristic that computes c points, stabbing a set of n d-dimensional axis-parallel boxes, in output-sensitive time $O(dn\log c)$ using linear space. Moreover, we proved that $c \leqslant b^*(1+\log_2 b^*)^{d-1}$, where b^* is the maximum number of pairwise disjoint boxes. We showed the tightness of the bounds by building a generic family $\mathscr{S}$ of d-dimensional boxes so that $c=\Omega(b^*\log_2^{d-1} b^*)$. We proved in the case of congruent boxes and 'constrained' boxes that $c \leqslant 2^{d-1}b^*$ and $c=O_d(b^*)$, respectively. Our algorithm can be easily parallelized onto PRAM computers in order to gain efficiency (see [2]). We plan to investigate the tradeoffs between the running time of any stabbing algorithm for sets of boxes and its relative performance. We also focus on the case of c-oriented objects and general convex/nonconvex objects [32].

This paper raises some open problems:

- The exact relationships between c^* and b^* (related to Wegner's conjecture that $c^* \leqslant 2b^*-1$ in the planar case).
- The hardness of approximation of constrained boxes compared with general boxes inside the polynomial-time solvable hierarchy of problems.
- Can we obtain better approximation algorithms by applying to our set of c stabbing points other (time-costly) algorithms?
- In [23], Imai and Asano gave an algorithm to compute a deepest cell in an arrangement of n isothetic boxes in $O(n^{d-1}\log n)$. Using similar ideas of Overmars and Yap [34], computing a deepest cell can be improved to $O(n^{d/2}\log n)$. Can we compute a deepest cell in a better running time? This would improve the running time of the greedy algorithm.

Another aspect of this problem that is currently being investigated is to give efficient algorithms to detect whether a set of objects is k-pierceable or not for small values of k [27].

To conclude, we give in Table 3 stating the main results obtained on the geometric set stabbing problem:

Finally, we would like to mention a related geometric covering/hitting set problem which has applications in classifiers and neural networks:

Problem 15. *Given a set $\mathscr{S}$ of n points in the plane, in degenerate position, find a minimal number of lines covering $\mathscr{S}$.*

Currently, the best approach which guarantees a performance ratio of O(log c^*), where c^* is the size of any optimal solution, is to apply the Brönnimann and Goodrich techniques [7].

Acknowledgement

I would like to thank Pr. Jean-Daniel Boissonnat, Dr. Hervé Brönnimann and Pr. Mariette Yvinec for helpful discussions. Many thanks to all members of Hiroshi Imai's laboratory who welcomed me during my first stay in Japan. I am also indebted to referees for their wise remarks and comments which enhanced the contents of this paper. Finally, I wish to thank Pr. Mario Tokoro for his support.

References

[1] P.K. Agarwal, M. van Kreveld, S. Suri, Label placement by maximum independent set in rectangles, Proc. 9th Canad. Conf. Comput. Geom., 1997, pp. 233–238.
[2] S.G. Akl, K.A. Lyons, Parallel Computational Geometry, Prentice-Hall, Englewood Cliffs, NJ, 1993.
[3] S. Arora, D. Karger, M. Karpinski, Polynomial-time approximation schemes for dense instances of NP-hard problems, Symp. on Theory of Comp., 1995, pp. 284–293.
[4] E. Assa, M.J. Katz, p-piercing sets of boxes and polytopes, Personal communication, 1998.
[5] M. Bellare, M. Sudan, Improved non-approximability results, Proc. 26th Annu. ACM Symp. Theory Comput., 1994, pp. 184–193.
[6] M. Blum, R.W. Floyd, V.R. Pratt, R.L. Rivest, R. Tarjan, Time bounds for selection, J. Comput. Systems Sci. 7 (1972).
[7] H. Brönnimann, M.T. Goodrich, Almost optimal set covers in finite VC-dimension, Discrete Comput. Geom. 14 (1995) 263–279.
[8] V. Chvátal, A greedy heuristic for the set-covering problem, Math. Oper. Res. 4 (1979) 233–235.
[9] L. Danzer, B. Grünbaum, Intersection properties of boxes in R^d, Combinatorica 2(3) (1982) 237–246.
[10] G.L. Ding, P. Seymour, P. Winkler, Bounding the vertex cover number of a hypergraph, Combinatorica 14(1) (1994) 23–34.
[11] R. Chii Duh, M. Fürer, Approximation of k-set cover by semi-local optimization, Proc. 29th Annu. ACM Symp. Theory Comput., 1997, pp. 256–264.
[12] A. Efrat, M.J. Katz, F. Nielsen, M. Sharir, Dynamic data structures for fat objects and their applications, Proc. Work. on Alg. and Data Struct. 97, LNCS 1272, Halifax, Nova Scotia, Canada, 1997, pp. 297–306.
[13] U. Feige, A threshold of $\ln n$ for approximating set cover (preliminary version), Proc. 28th Annual ACM Symp. on the Theory of Computing, Philadelphia, Pennsylvania, 22–24 May 1996, pp. 314–318.
[14] D.G. Fon der Flass, A.V. Kostochka, Covering boxes by points, Discrete Math. 120 (1993) 269–275.
[15] R.J. Fowler, M.S. Paterson, S.L. Tanimoto, Optimal packing and covering in the plane are NP-complete, Inform. Process. Lett. 12(3) (1981) 133–137.
[16] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, New York, NY, 1979.
[17] P.M. Gruber, J.M. Wills (Eds.), Handbook of Convex Geometry, vol. A, North-Holland, Amsterdam, Netherlands, 1993.
[18] H. Hadwiger, H. Debrunner, Kombinatorische Geometrie in der Ebene. Monographies de l'Enseignement Mathématique, N2, Geneva, 1960.
[19] D.S. Hochbaum, Approximation algorithms of the set covering and vertex cover problems, SIAM J. Comput. 11(3) (1982) 555–556.
[20] D.S. Hochbaum, W. Maass, Approximations schemes for covering and packing problems in image processing and VLSI, J. ACM 31 (1984) 130–136.

[21] D.S. Hochbaum, W. Maass, Fast approximation algorithms for a nonconvex covering problem, J. Algorithms 8 (1987) 305–323.
[22] R.Z. Hwang, R.C.T. Lee, R.C. Chang, The slab dividing approach to solve the Euclidean p-center problem, Algorithmica 9 (1993) 1–22.
[23] H. Imai, Ta. Asano, Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane, J. Algorithms 4 (1983) 310–323.
[24] G.Y. Károlyi, On point covers of parallel rectangles, Periodica Mathematica Hungarica 23(2) (1991) 105–107.
[25] G. Károlyi, G. Tardos, On point covers of multiples intervals and axis-parallel rectangles, Technical Report, DIMACS TR 95-45, 1995.
[26] M. Karpinski, A. Zelikovsky, Approximating dense cases of covering problems, Technical report, DIMACS TR 96-59, 1996.
[27] M.J. Katz, F. Nielsen, On piercing sets of objects, Proc. 12th Annu. ACM Symp. Comput. Geom., 1996, pp. 113–121.
[28] D.G. Kirkpatrick, R. Seidel, The ultimate planar convex hull algorithm? SIAM J. Comput. 15 (1986) 287–299.
[29] C. Makris, A. Tsakalidis, Fast piercing of iso-oriented rectangles, Proc. 9th Canad. Conf. Comput. Geom., 1997, pp. 217–222.
[30] J. Matoušek, Approximations and optimal geometric divide-and-conquer, Proc. 23rd Annu. ACM Symp. Theory Comput., 1991, pp. 505–511. Also in J. Comput. Systems Sci. (50)2 (1995) 203–208.
[31] F. Nielsen, Fast stabbing of boxes in high dimensions, Proc. 8th Canad. Conf. Comput. Geom., 1996, pp. 87–92.
[32] F. Nielsen, On the geometric piercing problem, Proc. 10th Franco-Japanese Franco-Chinese Conf. Combin. and Comp. Sci., Palaiseau, France, 1997.
[33] F. Nielsen, On Point Covers of c-Oriented polygons, Proc. 10th Canad. Conf. Comput. Geom. p. 6 and `http://cgm.cs.mcgill.ca/cccg98/`, 1998.
[34] M.H. Overmars, C.-K. Yap, New upper bounds in Klee's measure problem, SIAM J. Comput. 20 (1991) 1034–1045.
[35] F.P. Preparata, M.I. Shamos, Computational Geometry: An Introduction, Springer, New York, NY, 1985.
[36] D.J. Rosenkrantz, R.E. Stearns, P.M. Lewis, Approximate algorithms for the traveling salesperson problem, Proc. 15th Annu. IEEE Symp. Switching Automata Theory, 1974, pp. 33–42.
[37] P. Slavík, A tight analysis of the greedy algorithm for set cover, Proc. 28th Annual ACM Symp. on the Theory of Computing, Philadelphia, Pennsylvania, May 1996, pp. 435–441.
[38] G. Wegner, Eigenschaften der Nerven homologish-einfacher Familien im $\mathbb{R}^n$, Ph.D. Thesis, Göttingen, 1967.