Stabbing Convex Polygons with a Segment or a Polygon*

Pankaj K. Agarwal[1], Danny Z. Chen[2], Shashidhara K. Ganjugunte[1], Ewa Misiołek[3], Micha Sharir[4], and Kai Tang[5]

[1] Dept. of Comp. Sci., Duke University, Durham, NC 27708-0129.
[2] Dept. of Comp. Sci. and Engg., University of Notre Dame, Notre Dame, IN 46556.
[3] Mathematics Dept., Saint Mary's College, Notre Dame, IN 46556.
[4] School of Comp. Sci., Tel Aviv University, Tel Aviv 69978, and Courant Inst. of Math. Sci., NYC, NY 10012.
[5] Dept. of Mech. Engg., HKUST, Hong Kong, China.

Abstract. Let $\mathcal{O} = \{O_1, \ldots, O_m\}$ be a set of m convex polygons in $\mathbb{R}^2$ with a total of n vertices, and let B be another convex k-gon. A *placement* of B, any congruent copy of B (without reflection), is called *free* if B does not intersect the interior of any polygon in $\mathcal{O}$ at this placement. A placement z of B is called *critical* if B forms three "distinct" contacts with $\mathcal{O}$ at z. Let $\varphi(B, \mathcal{O})$ be the number of free critical placements. A set of placements of B is called a *stabbing set* of $\mathcal{O}$ if each polygon in $\mathcal{O}$ intersects at least one placement of B in this set.
We develop efficient Monte Carlo algorithms that compute a stabbing set of size $h = O(h^* \log m)$, with high probability, where h^* is the size of the optimal stabbing set of $\mathcal{O}$. We also improve bounds on $\varphi(B, \mathcal{O})$ for the following three cases, namely, (i) B is a line segment and the obstacles in $\mathcal{O}$ are pairwise-disjoint, (ii) B is a line segment and the obstacles in $\mathcal{O}$ may intersect (iii) B is a convex k-gon and the obstacles in $\mathcal{O}$ are disjoint, and use these improved bounds to analyze the running time of our stabbing-set algorithm.

1 Introduction

Problem statement. Let $\mathcal{O} = \{O_1, \ldots, O_m\}$ be a set of m convex polygons in $\mathbb{R}^2$ with a total of n vertices, and let B be another convex polygon. A *placement* of B is any congruent copy of B (without reflection). A set of placements of B is

* Work by P.A, S.G, and M.S, was supported by a grant from the U.S.-Israel Binational Science Foundation. Work by P.A. and S.G. was also supported by NSF under grants CNS-05-40347, CFF-06-35000, and DEB-04-25465, by ARO grants W911NF-04-1-0278 and W911NF-07-1-0376, and by an NIH grant 1P50-GM-08183-01 and by a DOE grant OEGP200A070505. Work by M.S. was partially supported by NSF Grant CCF-05-14079, by grant 155/05 from the Israel Science Fund, by a grant from the AFIRST joint French-Israeli program, and by the Hermann Minkowski–MINERVA Center for Geometry at Tel Aviv University. Work of D.C. was supported in part by the NSF under Grant CCF-0515203.

called a *stabbing set* of $\mathcal{O}$ if each polygon in $\mathcal{O}$ intersects at least one copy of B in this set. In this paper we study the problem of computing a small-size stabbing set of $\mathcal{O}$.

Terminology. A placement of B can be represented by three real parameters $(x, y, \tan(\theta/2))$ where (x, y) is the position of a reference point o in B, and θ is the counterclockwise angle by which B is rotated from some fixed orientation. The space of all placements of B, known as the *configuration space* of B, can thus be identified with $\mathbb{R}^3$ (a more precise identification would be with $\mathbb{R}^2 \times \mathbb{S}^1$; we use the simpler, albeit topologically less accurate identification with $\mathbb{R}^3$).

For a given point $z \in \mathbb{R}^3$, we use $B[z]$ to denote the corresponding placement (congruent copy) of B. Similarly, for a point $p \in B$ or a subset $X \subseteq B$, we use $p[z]$ and $X[z]$ to denote the corresponding point and subset, respectively, in $B[z]$. A placement z of B is called *free* if $B[z]$ does not intersect the interior of any polygon in $\mathcal{O}$, and *semifree* if $B[z]$ touches the boundary of some polygon(s) in $\mathcal{O}$ but does not intersect the interior of any polygon. Let $\mathbb{F}(B, \mathcal{O}) \subseteq \mathbb{R}^3$ denote the set of all free placements of B. For $1 \le i \le m$, let $K_i \subseteq \mathbb{R}^3$ denote the set of placements of B at which it intersects O_i. We refer to K_i as a *c-polygon.* Set $\mathcal{K}(B, \mathcal{O}) = \{K_1, \dots, K_m\}$. If B and the set $\mathcal{O}$ are obvious from the context, we use $\mathbb{F}$ and $\mathcal{K}$ to denote $\mathbb{F}(B, \mathcal{O})$ and $\mathcal{K}(B, \mathcal{O})$, respectively. Note that $\mathbb{F}(B, \mathcal{O}) = \text{cl}(\mathbb{R}^3 \setminus \bigcup \mathcal{K}(B, \mathcal{O}))$, where cl is the closure operator. If $\{B[z_1], \dots, B[z_h]\}$ is a stabbing set for $\mathcal{O}$, then each K_i contains at least one point in the set $\mathcal{Z} = \{z_1, \dots, z_h\}$, i.e., $\mathcal{Z}$ is a *hitting-set* for $\mathcal{K}$. Hence, the problem of computing a small-size stabbing set of $\mathcal{O}$ reduces to computing a small-size hitting set of $\mathcal{K}$.

We use a standard greedy algorithm (see, e.g., [6]) to compute a hitting set of $\mathcal{K}$. The efficiency of our algorithm depends on the combinatorial complexity of $\mathbb{F}$, defined below. We consider the following three cases:

(C1) B is a line segment and the polygons in $\mathcal{O}$ may intersect.

(C2) B is a line segment and the polygons in $\mathcal{O}$ are pairwise disjoint.

(C3) B is a convex k-gon and the polygons in $\mathcal{O}$ are pairwise disjoint.

A *contact* C is defined to be a pair (s, w) where s is a vertex of B and w is an edge of $O \in \mathcal{O}$, or w is a vertex of O and s is an edge of B. A *double contact* is a pair of contacts, and a *triple contact* is a triple of contacts. A placement z *forms* a contact $C = (s, w)$ if $s[z]$ touches w and $B[z]$ does not intersect the interior of the polygon $O \in \mathcal{O}$ containing w. A placement z *forms* a double contact $\{C_1, C_2\}$ if it forms both the contacts C_1 and C_2, and similarly it forms a triple contact $\{C_1, C_2, C_3\}$ if it forms all three of them; we also refer to triple-contact placements as *critical.* A double (or triple) contact is *realizable* if there is a placement of B at which this contact is formed. We call a double contact $\{C_1, C_2\}$ *degenerate* if both the contacts C_1 and C_2 involve the same polygon of $\mathcal{O}$. If z forms a degenerate double contact then either a vertex of $B[z]$ touches a vertex of $\mathcal{O}$ or an edge of $B[z]$ is flush with an edge of $\mathcal{O}$. A triple contact is called *degenerate* if its three contacts involve at most two polygons of $\mathcal{O}$, i.e., if it involves a degenerate double contact. If we decompose ∂K_i into maximal connected components so that all placements within a component form the same contact(s), then the edges and vertices on ∂K_i correspond to degenerate double and triple contacts, respectively (more precisely, the vertices are those triple contacts that involve at most two polygons). A *non-degenerate* triple contact (or critical) placement is formed by the intersection of the boundaries of three distinct c-polygons. Using the fact that each O_i is a convex polygon and B is also a convex polygon, it can be shown (see, e.g., [11]) that the complexity of $\mathbb{F}$ is proportional to the number of semifree critical placements, which we denote by $\varphi(B, \mathcal{O})$. We use $\varphi^*(B, \mathcal{O})$ to denote the number of semifree non-degenerate critical placements. In many cases $\varphi(B, \mathcal{O})$ is proportional to $\varphi^*(B, \mathcal{O})$ but in some cases $\varphi^*(B, \mathcal{O})$ can be much smaller. We improve the bounds on $\varphi(B, \mathcal{O})$ for all three cases (C1)–(C3), and on $\varphi^*(B, \mathcal{O})$ for (C2).

Related work. The general hitting-set problem is NP-hard, and it is believed to be intractable to obtain an $o(\log n)$-approximation [7]. An $O(\log n)$-approximation can be achieved by a simple greedy algorithm [16]. The hitting-set problem remains NP-hard even in a geometric setting [12, 13], and in some instances also hard to approximate [4]. However, in many cases polynomial-time algorithms with approximation factors better than $O(\log n)$ are known. For example, Hochbaum and Maass [9] devise $(1 + \varepsilon)$-approximation algorithms (for any $\varepsilon > 0$), for the problem of hitting a set of unit disks by a set of points. For set systems that typically arise in geometric problems, the approximation factor can be improved to $O(\log c^*)$, where c^* is the size of the optimal solution, and in some settings a constant factor approximation is also possible; see, e.g., [5].

Motivated by motion-planning and related problems in robotics, there is a rich body of literature on analyzing the complexity of the free space of a variety of moving systems B ("robots"), and a considerable amount of the earlier work has focussed on the cases where B is a line segment or a convex polygon translating and rotating in a planar polygonal workspace. Cases (C2) and (C3) correspond to these scenarios. It is beyond the scope of this paper to review all of this work. We refer the reader to the surveys [8, 14, 15]. We briefly mention the results that are directly related to our study. Leven and Sharir [10] proved that $\varphi(B, \mathcal{O}) = O(n^2)$ if B is a line segment and $\mathcal{O}$ is a set of pairwise-disjoint polygons with a total of n vertices. They also give a near-quadratic algorithm to compute $\mathbb{F}(B, O)$. For the case where B is a convex k-gon, Leven and Sharir [11] proved that $\varphi(B, \mathcal{O}) = O(k^2 n^2 \beta_6(kn))$, where $\beta_s(t) = \lambda_s(t)/t$, and $\lambda_s(t)$ is the maximum length of an (t, s)-Davenport-Schinzel sequence [15]; $\beta_s(t)$ is an extremely slowly growing function of t.

Our results. There are two main contributions of this paper. First, we refine the earlier bounds on $\varphi(B, \mathcal{O})$ so that they also depend on the number m of polygons in $\mathcal{O}$, and not just on their total number of vertices, since $m \ll n$ in many cases. Second, we present a general approach for computing a hitting set, which leads to faster algorithms for computing stabbing sets.

Specifically, we first prove (in Section 2), for the case where B is a line segment, that the complexity of $\mathbb{F}(B, \mathcal{O})$ is $O(mn\alpha(n))$, and that $\mathbb{F}(B, \mathcal{O})$ can be computed in $O(mn\alpha(n) \log^2 n)$ randomized expected time. If the polygons in $\mathcal{O}$ are pairwise disjoint, then $\varphi(B, \mathcal{O}) = \Theta(mn)$, but $\varphi^*(B, \mathcal{O}) = O(m^2 + n)$. We then show that we can compute, in $O((m^2+n)\log m \log^2 n)$ randomized expected time, an implicit representation of $\mathbb{F}$ of size $O(m^2+n)$, which is sufficient for many applications (including ours). We then consider case (C3) (Section 3). We show that $\varphi(B, \mathcal{O}) = O(k^2 mn\beta_6(kn))$ in this case, and that $\mathbb{F}$ can be computed in expected time $O(k^2 mn\beta_6(kn)\log(kn)\log n)$.

The subsequent results in this paper depend on the complexity of $\mathbb{F}$. Since we are mainly interested in bounds that are functions of the number of polygons and of their total size, we abuse the notation a little, and write $\varphi(m, n)$ to denote the maximum complexity of $\mathbb{F}$ for each of the three cases; the maximum is taken over all m convex polygons with a total of n vertices, and these polygons are disjoint for cases (C2) and (C3). Similarly we define $\varphi^*(m, n)$ for the maximum number of nondegenerate critical placements (in case (C3), the bounds also depend on k).

For a point $z \in \mathbb{R}^3$, we define its *depth* to be the number of c-polygons K_i that contain z. We present a randomized algorithm DEPTH_THRESHOLD , which, given an integer $l \le m$, determines whether the maximum depth of a placement (with respect to $\mathcal{O}$) is at most l. If not, it returns all critical placements (of depth at most l). The expected running time of this algorithm is $O(l^3\varphi(m/l, n/l)\log n)$. For (C2), the procedure runs in expected time $O(l^3\varphi^*(m/l, n/l)\log^2 n)$ time.

Finally, we describe algorithms for computing a hitting set of $\mathcal{K}$ of size $O(h^*\log m)$ where h^* is the size of the smallest hitting set of $\mathcal{K}$. Basically, we use the standard greedy approach, mentioned above, to compute such a hitting set, but we use more efficient implementations, which exploit the geometric structure of the problems at hand. The first implementation runs in $O(\Delta^3\varphi(m/\Delta, n/\Delta)\log n)$ time, where Δ is the maximum depth of a placement. The second implementation is a Monte Carlo algorithm, based on a technique of Aronov and Har-Peled [3] for approximating the depth in an arrangement. The expected running time of the second implementation is $O(\varphi(m, n)h\log m\log n + mn^{1+\varepsilon})$ time, where h is the size of the hitting set computed by the algorithm, which is $O(h^*\log m)$, with high probability. Finally, we combine the two approaches and obtain a Monte Carlo algorithm whose running time is $O(\varphi(m, n)\cdot n^{\varepsilon} + \eta^3\varphi(m/\eta, n/\eta)\log n\log^3 m)$, for any $\varepsilon > 0$, where $\eta = \min\{h^{1/3}, m^{1/4}\}$ and $h = O(h^*\log m)$, with high probability. For case (C2), the expected running time can be improved to $O(\varphi^*(m, n)\cdot n^{\varepsilon} + \eta^3\varphi^*(m/\eta, n/\eta)\log^c n))$, for some constant $c > 1$. We believe that one should be able to improve the expected running time to $O(\varphi(m, n)\log^{O(1)} n)$, but such a bound remains elusive for now. Because of lack of space many algorithms and proofs are omitted from this abstract, which can be found in the full version of this paper [1].

2 Complexity of $\mathbb{F}$ for a Segment

Let B be a line segment of length d, and let $\mathcal{O}$ be a set of m convex polygons in $\mathbb{R}^2$ with a total of n vertices. We first bound the number of critical placements when the polygons in $\mathcal{O}$ may intersect, and then prove a refined bound when the polygons are pairwise disjoint. We omit the algorithms for computing these placements from this abstract.

The case of intersecting polygons. There are several types of critical placements of B (see Figure 1(a)):
(i) A placement where one endpoint of B touches a vertex of one polygon and the other endpoint touches an edge of another polygon.
(ii) A placement where one endpoint of B touches a vertex of one polygon and the relative interior of B touches a vertex of another polygon.
(iii) The relative interior of B touches two vertices (of the same or of distinct polygons) and one endpoint of B touches a polygon edge.
(iv) The relative interior of B touches a vertex of a polygon, and one of its endpoints touches an intersection point of two edges (of distinct polygons).
(v) One endpoint of B touches an intersection point of two edges (of distinct polygons), and the other endpoint touches a third edge.
(vi) The relative interior of B touches a vertex of a polygon, and its two endpoints touch two respective edges (of distinct polygons).
There are $O(mn)$ placements of types (i) and (ii), and $O(m^2 + n)$ placements

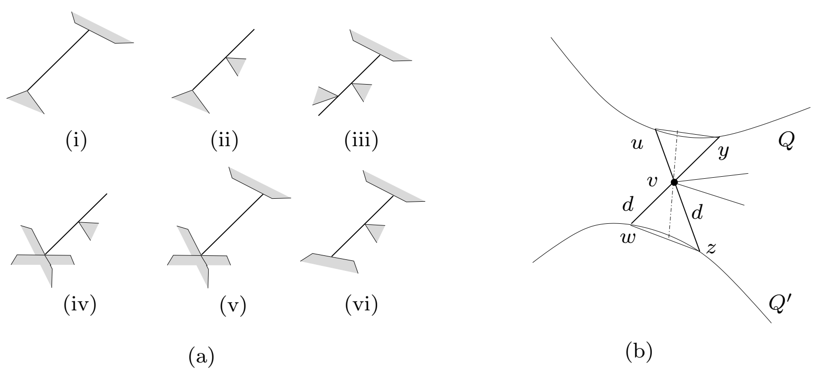

Fig. 1. (a) Critical free placements of B; (b) F_Q and $G_{Q'}$ intersect at most twice.

of type (iii).

Consider the placements of types (iv) and (v). Let u be an intersection point of two polygon boundaries (which lies on the boundary of their union), and let H denote the *hole* (i.e., connected component of the complement) of the union of $\mathcal{O}$ which contains u on its boundary. Again, placing an endpoint of B at u leaves B with one degree of freedom of rotation about u. However, at any such free placement, B must be fully contained in (the closure of) H. For any polygon $O \in \mathcal{O}$ whose boundary contributes to ∂H, there are at most two critical free placements of types (iv) and (v) where B swings around u and touches O, and no other polygon (namely, those which do not show up on ∂H) can generate such a placement. It follows that, for any polygon $O \in \mathcal{O}$, the intersection points u that can form with O critical free placements of type (iv) or (v) are vertices of the *zone* of ∂O in the arrangement $\mathcal{A}(\mathcal{O} \setminus \{O\})$. Since ∂O is convex, the complexity of the zone is $O(n\alpha(n))$ [2]. Hence the overall number of such placements is $O(mn\alpha(n))$.

Finally, consider critical free placements of type (vi). Let v be a fixed vertex of some polygon (not lying inside any other polygon). The placements of B at which its relative interior touches v can be parametrized in a polar coordinate system (r, θ), where r is the distance of one endpoint a of B from v, and θ is the orientation of B, oriented towards a, so that O lies to the right of (the line supporting) B. The admissible values of (r, θ) can be restricted to the rectangle $[0, d] \times I$, where I is the range of orientations of tangent lines to O at v, for which O lies to their right. For any polygon $Q \in \mathcal{O} \setminus \{O\}$, we define a *forward function* $r = F_Q(\theta)$ and a *backward function* $r = G_Q(\theta)$, where $F_Q(\theta)$ (resp., $G_Q(\theta)$) is the distance from v to $\ell_\theta \cap Q$ (resp., d minus that distance), where ℓ_θ is the line at orientation θ that passes through v. $F_Q(\theta)$ (resp., $G_Q(\theta)$) is defined only when $\ell_\theta \cap Q$ is nonempty, lies ahead (resp., behind) v along ℓ_θ, and its distance from v is at most d; in all other cases, we set $F_Q(\theta) := d$ (resp., $G_Q(\theta) := 0$). It is clear that the set $\mathbb{F}_v$ of free placements of B when its relative interior hinges over v, is given in parametric form by

$$\{(r, \theta) \mid \max_Q G_Q(\theta) \leq r \leq \min_Q F_Q(\theta)\}.$$

That is, $\mathbb{F}_v$, in parametric form, is the *sandwich region* between the lower envelope of the functions F_Q and the upper envelope of the functions G_Q. It follows that the combinatorial complexity of $\mathbb{F}_v$ is proportional to the sum of the complexities of the two individual envelopes. A placement of B, where one endpoint lies either at a vertex of some polygon (including v itself), or at the intersection point between two edges of distinct polygons, its relative interior touches v, and the portion of B between these two contacts is free, corresponds to a *breakpoint* in one of the envelopes. Arguing as in the analysis of the preceding types of critical placements, the overall number of such placements, summed over all vertices v, is $O(mn\alpha(n))$. It follows that the overall number of critical placements of type (vi) is also $O(mn\alpha(n))$. Putting everything together, we obtain:

Theorem 1. *Let B be a line segment and let $\mathcal{O}$ be a set of m (possibly intersecting) convex polygons in $\mathbb{R}^2$ with n vertices in total. The number of free critical placements of B is $O(mn\alpha(n))$.*

The case of pairwise-disjoint polygons. We now prove a refined bound on the number of free critical placements if the polygons in $\mathcal{O}$ are pairwise disjoint. A trivial construction shows that, even in this case, there can be $\Omega(mn)$ free critical placements of types (i) and (ii). However, most of these placements involve contacts with only two distinct polygons, so they are degenerate critical contacts. As we next show, the number of nondegenerate critical contacts is smaller. Specifically, we argue that there are only $O(m^2 + n)$ free nondegenerate critical placements.

We have already ruled out critical placements of types (i) and (ii) because they are degenerate, and we rule out placements of type (iv) and (v) because they involve intersecting polygons. It thus remains to bound the number of free critical placements of types (iii) and (vi). There are only $O(m^2 + n)$ critical placements of type (iii), as argued above. For placements of type (vi), we use the same scheme as above, fixing the pivot vertex v and considering the system of functions $F_Q(\theta)$, $G_Q(\theta)$ in polar coordinates about v. Let $\mathcal{L}_v(\theta) = \min_Q F_Q(\theta)$ and $\mathcal{U}_v(\theta) = \max_Q G_Q(\theta)$; Let μ_v (resp. ν_v) be the number of breakpoints in $\mathcal{L}_v$ (resp. $\mathcal{U}_v$). Using the fact that the functions F_Q (and G_Q) are pairwise disjoint, we claim the following:

Lemma 1. $\sum_v (\mu_v + \nu_v) = O(m^2 + n)$.

If we mark the θ-values at which a breakpoint of $\mathcal{L}_v$ or $\mathcal{U}_v$ occurs, we partition the θ-range into intervals so that each of $\mathcal{L}_v$ and $\mathcal{U}_v$ is attained by (a connected portion of the graph of) a single function, say F_Q and $G_{Q'}$, respectively. We claim that F_Q and $G_{Q'}$ intersect in at most two points in this interval, i.e., there are two semifree placements of B such that v lies in the interior of B and the endpoints of B lie on ∂Q and $\partial Q'$; see Figure 1(b). Hence, the number of vertices in the sandwich region between $\mathcal{L}_v$ and $\mathcal{U}_v$ is $O(\mu_v + \nu_v)$. Putting everything together, we obtain:

Theorem 2. *Let B be a line segment, and let $\mathcal{O}$ be a set of pairwise-disjoint convex polygons with n vertices in total. The number of nondegenerate free critical placements of B is $O(m^2 + n)$.*

3 Complexity of $\mathbb{F}$ for a Convex k-gon

In this section we derive an improved bound on $\varphi(B, \mathcal{O})$ for the case where B is a convex k-gon and $\mathcal{O}$ is a set of m pairwise-disjoint convex polygons in $\mathbb{R}^2$ with n vertices in total. We assume that the polygons in $\mathcal{O}$ are in general position, as in [11]. We first prove that the number of degenerate free critical placements is $O(k^2mn)$, and then show that the total number of realizable double contacts is $O(k^2mn)$. By adapting the argument of Leven and Sharir [11], we then prove that $\varphi(B, \mathcal{O}) = O(k^2mn\beta_6(kn))$. We begin by stating a lemma, which establishes an upper bound on the number of realizable double contacts when there are only two obstacles.

Lemma 2. *Let B be a convex k-gon, and let O_1 and O_2 be two disjoint convex polygons with n_1 and n_2 vertices, respectively, then the number of semifree degenerate critical placements in $\mathbb{F}(B, \{O_1, O_2\})$ is $O(k^2(n_1 + n_2))$.*

The following corollary follows immediately from Lemma 2.

Corollary 1. *Let B be a convex k-gon and let $\mathcal{O}$ be a set of m pairwise-disjoint convex polygons with n vertices in total. The number of degenerate critical placements in $\mathbb{F}(B, \mathcal{O})$ is $O(k^2mn)$.*

Next, we bound the number of realizable double contacts. It is tempting to prove that a fixed contact C can realize only $O(km)$ double contacts, but, as shown in the full version, a contact may be involved in $\Omega(kn)$ realizable double contacts, so we have to rely on a more global counting argument. Note first that the preceding argument shows that the number of degenerate double contacts is $O(k^2mn)$, so it suffices to consider only nondegenerate double contacts. Since we assume that the polygons are in general position, the locus of placements forming a fixed non-degenerate double contact $\{C_1, C_2\}$ is a curve in $\mathbb{R}^3$. Let O_1 and O_2 be the two (distinct) polygons involved in $\{C_1, C_2\}$. Adapting the argument in [15, Lemma 8.55], one can show that at least one endpoint of this curve is a degenerate triple contact, which we denote by $z(C_1, C_2)$, which is semifree with respect to O_1 and O_2. We thus charge $\{C_1, C_2\}$ to $z(C_1, C_2)$, and argue that each nondegenerate triple contact in $\mathbb{F}(B, \{O_1, O_2\})$ is charged at most $O(1)$ times. Omitting all further details, we obtain:

Lemma 3. *Let B be a convex k-gon and let $\mathcal{O}$ be a set of m pairwise-disjoint convex polygons with n vertices in total. The number of realizable double contacts is $O(k^2mn)$.*

Plugging Corollary 1 and Lemma 3 into the proof of Leven and Sharir [11], we obtain the main result of this section.

Theorem 3. *Let B be a convex k-gon, and let $\mathcal{O}$ be a set of m pairwise-disjoint convex polygons with n vertices in total. Then $\varphi(B, \mathcal{O}) = O(k^2mn\beta_6(kn))$.*

4 Computing Critical Placements

So far, we have only considered *semifree* critical placements, but, since we want to construct a set of stabbing placements of B, we need to consider (and compute) the set of all (nonfree) critical placements.

Bounding the number of critical placements. Let $\mathcal{K} = \{K_1, \ldots, K_m\}$ be the set of c-polygons yielded by B and $\mathcal{O}$, as defined in the Introduction, and let $\mathcal{A}(\mathcal{K})$ denote the 3-dimensional arrangement of $\mathcal{K}$. For a point $z \in \mathbb{R}^3$ and a subset $\mathcal{G} \subseteq \mathcal{K}$, let $\Delta(z, \mathcal{G})$ denote the *depth* of z with respect to $\mathcal{G}$, i.e., the number of c-polygons in $\mathcal{G}$ containing z in their interior; we use $\Delta(z)$ to denote $\Delta(z, \mathcal{K})$. Let $\Phi_l(\mathcal{K})$ denote the set of vertices of $\mathcal{A}(\mathcal{K})$, whose depth is l, and put $\Phi_{\le l}(\mathcal{K}) = \bigcup_{h \le l} \Phi_h(\mathcal{K})$. Set $\varphi_l(\mathcal{K}) = |\Phi_l(\mathcal{K})|$ and $\varphi_{\le l}(\mathcal{K}) = |\Phi_{\le l}(\mathcal{K})|$. We now state a theorem, whose proof is deferred to the full version of this paper.

Theorem 4. *(i) Let B be a line segment, let $\mathcal{O}$ be a set of m convex polygons in $\mathbb{R}^2$ with a total of n vertices, and let $\mathcal{K} = \mathcal{K}(B, \mathcal{O})$. Then, for any $1 \le l \le m$, we have $\varphi_{\le l}(\mathcal{K}) = O(mnl\alpha(n))$. If the polygons in $\mathcal{O}$ are pairwise disjoint, then the number of non-degenerate critical placements in $\Phi_{\le l}(\mathcal{K})$ is $O(m^2l + nl^2)$.*

(ii) Let B be a convex k-gon, let $\mathcal{O}$ be a set of m pairwise-disjoint polygons in $\mathbb{R}^2$ with a total of n vertices, and let $\mathcal{K} = \mathcal{K}(B, \mathcal{O})$. Then, for any $1 \le l \le m$, we have $\varphi_{\le l}(\mathcal{K}) = O(k^2mnl\beta_6(kn))$.

The DEPTH_THRESHOLD *procedure.* One of the strategies that we will use for computing a stabbing set is based on determining whether the maximal depth in $\mathcal{A}(\mathcal{K})$ exceeds a given threshold l. For this we use the DEPTH_THRESHOLD procedure, which, given an integer $l \geq 1$, determines whether DEPTH $(\mathcal{K}) \leq l$. If not, it returns a critical placement whose depth is greater than l. Otherwise, it returns all critical placements of B (which are all the vertices of $\mathcal{A}(\mathcal{K})$). Without describing the details of this procedure, we claim the following.

Theorem 5. *(i) Let B be a line segment, and let $\mathcal{O}$ be a set of m convex polygons in $\mathbb{R}^2$ with a total of n vertices. For a given integer $1 \leq l \leq m$, the* DEPTH_THRESHOLD (l) *procedure takes $O(mn(\log n + l\alpha(n)))$ expected time. If the polygons in $\mathcal{O}$ are pairwise disjoint, the expected running time is $O((m^2l + nl^2)\log^2 n)$.*

(ii) Let B be a convex k-gon and $\mathcal{O}$ be a set of m pairwise-disjoint convex polygons in $\mathbb{R}^2$ with a total of n vertices. For a given integer $1 \leq l \leq m$, the DEPTH_THRESHOLD (l) *procedure takes $O(k^2mn(\log n + l\beta_6(kn)))$ expected time.*

5 Computing a Hitting Set

Let $\mathcal{K} = \{K_1, \ldots, K_m\}$ be the set of c-polygons, for an input collection $\mathcal{O}$ of convex polygons and a line segment or convex polygon B, as above. Our goal is to compute a small-size hitting set for $\mathcal{K}$, and we do it by applying a standard greedy technique which proceeds as follows. In the beginning of the ith step we have a subset $\mathcal{K}_i \subseteq \mathcal{K}$; initially $\mathcal{K}_1 = \mathcal{K}$. We compute a placement $z_i \in \mathbb{R}^3$ such that $\Delta(z_i, \mathcal{K}_i) =$ DEPTH $(\mathcal{K}_i)$, and we also compute the set $\mathcal{K}_{z_i} \subseteq \mathcal{K}_i$ of the c-polygons that contain z_i. We add z_i to H, and set $\mathcal{K}_{i+1} = \mathcal{K}_i \setminus \mathcal{K}_{z_i}$. The algorithm stops when $\mathcal{K}_i$ becomes empty. The standard analysis of the greedy algorithm [6] shows that $|H| = O(h^* \log m)$, where h^* is the size of the smallest hitting set for $\mathcal{K}$. In fact, the size of H remains $O(h^* \log m)$, even if at each step we choose a point z_i such that $\Delta(z_i, \mathcal{K}_i) \geq$ DEPTH $(\mathcal{K}_i)/2$. We describe three different procedures to implement this greedy algorithm. The first one, a Las Vegas algorithm, works well when DEPTH $(\mathcal{K})$ is small. The second one, a Monte Carlo algorithm, works well when h^* is small. Finally, we combine the two approaches to obtain an improved Monte Carlo algorithm. For simplicity, and due to lack of space, we focus on case (C1): B is a segment and the polygons in $\mathcal{O}$ may intersect.

The Las Vegas algorithm. It suffices to find a deepest point in $\mathcal{A}(\mathcal{K})$ that lies on ∂K_i for some i, and that (assuming general position), we may assume it to lie in the relative interior of some 2-face (the depth of all the points within the same 2-face is the same). Thus, for each 2-face f of $\mathcal{A}(\mathcal{K})$ we choose a sample point z_f. Let $\mathcal{Z} \subseteq \mathbb{R}^3$ be the set of these points. We maintain $\Delta(z, \mathcal{K}_i)$ for each $z \in \mathcal{Z}$, as we run the greedy algorithm, and return $z_i = \arg\max_{z \in \mathcal{Z}} \Delta(z, \mathcal{K}_i)$ at each step, and delete the c-polygons containing z_i from $\mathcal{K}$. It will be expensive to maintain the depth of each point in $\mathcal{Z}$ explicitly. We describe a data structure that maintains the depth of each placement z_i in $\mathcal{Z}$ implicitly, supports deletion of c-polygons and returns a placement of maximum depth. For each c-polygon K_j, let $\Gamma_j = \{\gamma_{ji} = \partial K_j \cap K_i \mid i \neq j\}$ be a set of regions on ∂K_j. We compute $\mathcal{A}(\Gamma_j)$ using Theorem 5. Let $\mathcal{D}(\Gamma_j)$ be the planar graph that is dual to $\mathcal{A}(\Gamma_j)$. We choose a representative point z_f from each face f of $\mathcal{A}(\Gamma_j)$, and use z_f to denote the node of $\mathcal{D}(\Gamma_j)$ dual to f. If an edge e of $\mathcal{A}(\Gamma_j)$ lies on ∂K_a, for some $K_a \in \mathcal{K}$, we label the edge e of $\mathcal{D}(\Gamma_j)$ with K_a and denote this label by $\chi(e)$. We compute a spanning tree T of $\mathcal{D}(\Gamma_j)$, and then convert T into a path Π by performing a traversal of T, starting from some leaf v; each edge of T appears twice in Π. The sequence of vertices in Π can be decomposed into intervals, such that all vertices in each interval either lie in a c-polygon K_a or none of them lie inside K_a. Let J_a be the subset of those intervals whose vertices lie inside K_a. We represent an interval $v_x, \ldots, v_y$ by the pair $[x, y]$. Set $J = \bigcup_{a \neq j} J_a$. For any vertex $v_s \in \Pi$, we define the weight $w(v_s)$ to be the number of intervals $[x, y]$ in J that contain v_s, i.e., intervals satisfying $x \leq s \leq y$. For a subset $\mathcal{G} \subseteq \mathcal{K}$, $\Delta(v_s, \mathcal{G})$ is the number of intervals in $\bigcup_{K_a \in \mathcal{G}} J_a$ that contain v_s. We store J in a segment tree, Σ, built on the sequence of edges in Π. Each node σ of Σ corresponds to a subpath Π_σ of Π. For each σ, we maintain the vertex of Π_σ of the maximum weight. The root of Σ stores a vertex of Π of the maximum weight. Once we have computed $\mathcal{A}(\Gamma_j)$, J and Σ can be constructed in $O(\kappa_j \log \kappa_j)$ time, where κ_j is the complexity of $\mathcal{A}(\Gamma_j)$. We have $\sum_j \kappa_j = O(mn\Delta\alpha(n))$, where $\Delta = \textsc{Depth}\ (\mathcal{K})$. The information in Σ can be updated in $O(\log n)$ time when an interval is deleted from J. When the greedy algorithm deletes a c-polygon K_a, we delete all intervals in J_a from J and update Σ. The total time spent in updating Σ is $O(\kappa_j \log n)$. Maintaining this structure for each c-polygon K_j, the greedy algorithm can be implemented in $O(mn\Delta\alpha(n) \log n)$ expected time.

Lemma 4. *A hitting set of $\mathcal{K}$ of size $O(h^* \log m)$ can be computed in expected time $O(mn\Delta\alpha(n) \log n)$, where $\Delta = \textsc{Depth}\ (\mathcal{K})$ and where h^* is the size of a smallest hitting set of $\mathcal{K}$.*

A simple Monte Carlo algorithm. Let $\Delta = \textsc{Depth}\ (\mathcal{K})$. If $\Delta = O(\log m)$, we use the above algorithm and compute a hitting set in time $O(mn\alpha(n) \log m \log n)$. So assume that $\Delta \geq c \log m$ for some constant $c \geq 1$. We use a procedure by Aronov and Har-Peled [3], which computes a placement whose depth is at least $\Delta/2$. Their main algorithm is based on the following observation. Fix an integer $l \geq \Delta/4$. Let $\mathcal{G} \subseteq \mathcal{K}$ be a random subset obtained by choosing each c-polygon of $\mathcal{K}$ with probability $\rho = (c_1 \ln m)/l$, where c_1 is an appropriate constant. Then the following two conditions hold with high probability, (i) if $\Delta \geq l$ then $\textsc{Depth}\ (\mathcal{G}) \geq 3l\rho/2 = (3c_1/2) \ln m$, and, ii) if $\Delta \leq l$ then $\textsc{Depth}\ (\mathcal{G}) \leq 5l\rho/4 = (5c_1/4) \ln m$.

This observation immediately leads to a binary-search procedure for approximating $\textsc{Depth}\ (\mathcal{K})$. Let $\tau = (5c_1/4) \ln m$. In the ith step, for $i \leq \lceil \log_2(m/\log_2 m) \rceil$, we set $l_i = m/2^i$. We choose a random subset $\mathcal{G}_i \subseteq \mathcal{K}$ using the parameter $l = l_i$, and then run the procedure $\textsc{Depth\_Threshold}$ on $\mathcal{G}_i$ with parameter τ. If the procedure determines that $\textsc{Depth}\ (\mathcal{G}_i) \leq \tau$, then we conclude that $\textsc{Depth}\ (\mathcal{K}) \leq l_i$, and we continue with the next iteration. Otherwise, the algorithm returns a point $z \in \mathbb{R}^3$ such that $\Delta(z, \mathcal{G}_i) \geq \tau$. We need a data structure for reporting the set of polygons in $\mathcal{O}$ intersected by $B[z]$ for a placement $z \in \mathbb{R}^3$. As we show in the full version, we can preprocess $\mathcal{O}$ into a data structure of size $O(mn^{1+\varepsilon})$, for any $\varepsilon > 0$, so that a convex polygon O_i of $\mathcal{O}$ can be deleted in time $O(|O_i| \cdot n^{\varepsilon})$, where $|O_i|$ is the number of vertices of O_i, and so that the set of all κ polygons intersecting a query placement $B[z]$ of B can be reported in time $O((1+\kappa)\log n)$.

Set $m_i = |\mathcal{G}_i|$, and let n_i be the number of vertices in the original polygons corresponding to the c-polygons in $\mathcal{G}_i$. Then the expected running time of the ith iteration is $O(m_i n_i \tau \alpha(n) \log n)$. Since $E[m_i n_i] = O(mn\rho^2 + n\rho)$, the expected running time of the ith iteration is $O((mn/l_i^2)\alpha(n)\log^3 m \log n)$.

Since the algorithm always stops after at most $\lceil \log_2(m/\log_2 m)\rceil$ iterations, the overall expected running time is $O(mn\alpha(n)\log m \log n)$. Note that if the algorithm stops after i steps, then, with high probability, $\Delta \in [l_i, 2l_i]$. Hence, the expected running time of the algorithm is $O((mn/\Delta^2)\alpha(n)\log^3 m \log n)$. Plugging this procedure into the greedy algorithm described above, and accounting for $O(mn^{1+\varepsilon})$ time for preprocessing and reporting the polygons intersecting a placement z, we get the following lemma.

Lemma 5. *There is a Monte Carlo algorithm for computing a hitting set of $\mathcal{K}$ whose size is $h = O(h^* \log m)$ with probability at least $1 - 1/m^{O(1)}$, and whose expected running time is $O(mnh\alpha(n)\log m \log n + mn^{1+\varepsilon})$.*

An improved Monte Carlo algorithm. We now combine the two algorithms given above, to obtain a faster algorithm for computing a small-size hitting set of $\mathcal{K}$. For this we use the data structure mentioned above, which preprocesses $\mathcal{O}$ in $O(mn^{1+\varepsilon})$ time to support deletion.

We now run the greedy algorithm as follows. We begin by running the Monte Carlo algorithm described above. In the ith iteration, it returns a point z_i such that $\Delta(z_i, \mathcal{K}_i) \geq \text{Depth}\ (\mathcal{K}_i)/2$, with high probability. We use the above data structure to report the set $\mathcal{O}_{z_i}$ of all polygons that intersect the query placement $B[z_i]$, or, equivalently, the set $\mathcal{K}_{z_i}$ of the c-polygons that contain z_i. We delete these polygons from the data structure. If $|\mathcal{O}_{z_i}| < i^{1/3}$ then we switch to the Las Vegas algorithm described earlier, to compute a hitting set of $\mathcal{K}_{i+1}$.

We now analyze the expected running time of the algorithm. The total time spent in reporting the polygons intersected by the placements $B[z_1], \ldots, B[z_h]$, is $O(mn^{1+\varepsilon})$, so it suffices to bound the time spent in computing $z_1, \ldots, z_h$. Suppose that the algorithm switches to the second stage after $\xi + 1$ steps. Then $\text{Depth}\ (\mathcal{K}_i) \geq \xi^{1/3}$, for $1 \leq i \leq \xi$, and, the expected running time of each of the iterations of the first stage is $O((mn/\xi^{2/3})\alpha(n)\log n \log^3 m)$. Hence, the expected running time of the first stage is $O(mn\xi^{1/3}\alpha(n)\log n \log^3 m)$. The expected running time of the second stage is $O(mn\xi^{1/3}\alpha(n)\log n)$ because $\text{Depth}\ (\mathcal{K}_{\xi+2}) \leq 2\xi^{1/3}$. Suppose h is the size of the hitting set computed by the algorithm. Then $\xi \leq h$. Moreover, for $1 \leq i \leq \xi$, each z_i lies inside at least $(\xi^{1/3})/2$ c-polygons of $\mathcal{K}_i$, and all these polygons are distinct. Therefore, $\xi^{4/3} \leq 2m$. The expected running time of the overall algorithm is $O(mn\eta\alpha(n)\log n \log^3 m + mn^{1+\varepsilon})$, where $\eta = \min\{m^{1/4}, h^{1/3}\}$. We thus obtain the following.

Theorem 6. *Let B be a line segment, and let $\mathcal{O}$ be a set of m (possibly intersecting) convex polygons in $\mathbb{R}^2$, with a total of n vertices. A stabbing set of $\mathcal{O}$ of $h = O(h^* \log m)$ placements of B can be computed, with probability at least $1 - 1/m^{O(1)}$, in expected time $O(mn(n^\varepsilon + \eta\alpha(n) \log n \log^3 m))$, where $\eta = \min\{m^{1/4}, h^{1/3}\}$, h^* is the smallest size of a hitting set, and $\varepsilon > 0$ is an arbitrarily small constant.*

Remark: The expected running time of the above approach is $O((m^2 + n)n^\varepsilon + (m^2\eta + n\eta^2) \log^c(n))$ for case (C2) and $O(k^2mn(n^\varepsilon + \eta\beta_6(kn) \log n \log^3 m))$ for case (C3).

References

1. P. K. Agarwal, D. Z. Chen, S. K. Ganjugunte, E. Misołek, M. Sharir, and K. Tang. Stabbing convex polygons with a segment or a polygon. `http://www.cs.duke.edu/~shashigk/sstab/shortstab.pdf`, 2008.
2. P. K. Agarwal and M. Sharir, Arrangements and their applications, in: *Handbook of Computational Geometry* (J.-R. Sack and J. Urrutia, eds.), Elsevier, 2000, pp. 49–119.
3. B. Aronov and S. Har-Peled, On approximating the depth and related problems, *Proc. of the 16th Annu. ACM-SIAM Sympos. Discrete Algorithms*, 2005, pp. 886–894.
4. P. Berman and B. DasGupta, Complexities of efficient solutions of rectilinear polygon cover problems, *Algorithmica*, 17 (1997), 331–356.
5. K. L. Clarkson and K. Varadarajan, Improved approximation algorithms for geometric set cover, *Discrete Comput. Geom.*, 37 (2007), 43–58.
6. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 2001.
7. U. Feige, A threshold of $\ln n$ for approximating set cover, *J. ACM*, 45 (1998), 634–652.
8. D. Halperin, L. Kavraki, and J.-C. Latombe, Robotics, in: *Handbook of Discrete and Computational Geometry* (J. E. Goodman and J. O'Rourke, eds.), CRC Press, Inc., Boca Raton, FL, USA, 1997, pp. 755–778.
9. D. S. Hochbaum and W. Maass, Approximation schemes for covering and packing problems in image processing and VLSI, *J. ACM*, 32 (1985), 130–136.
10. D. Leven and M. Sharir, An efficient and simple motion planning algorithm for a ladder moving in two-dimensional space amidst polygonal barriers, *Proc. 1st Annu. Sympos. on Comput. Geom.*, ACM, 1985, pp. 221–227.
11. D. Leven and M. Sharir, On the number of critical free contacts of a convex polygonal object moving in two-dimensional polygonal space., *Discrete Comput. Geom.*, 2 (1987), 255–270.
12. N. Megiddo and K. J. Supowit, On the complexity of some common geometric location problems, *SIAM J. Comput.*, 13 (1984), 182–196.
13. N. Megiddo and A. Tamir, On the complexity of locating linear facilities in the plane., *Operations Research Letters*, 1 (1982), 194–197.
14. M. Sharir, Algorithmic motion planning in robotics, *IEEE Computer*, 22 (1989), 9–20.
15. M. Sharir and P. K. Agarwal, *Davenport-Schinzel Sequences and their Geometric Applications*, Cambridge University Press, New York, 1995.
16. V. Vazirani, *Approximation Algorithms*, Springer, Heidelberg, 2004.