Computational Geometry Impact Potential: A Business and Industrial Perspective

Prabhakar Raghavan

Department K53, IBM Almaden Research Center
650 Harry Road, San Jose CA 95120
pragh@almaden.ibm.com

Abstract

The computational geometry impact task force (Chazelle et al.) has done an outstanding and comprehensive job of reviewing a number of potential application areas for geometric computing. Their report identifies a number of areas where geometric computing has potential for impact, as well as techniques in computational geometry that could be invoked for these applications. This talk will view these areas from a business and industrial perspective, pointing out where we believe the biggest impact will be outside of the research community.

References

http://www.cs.princeton.edu/~chazelle/taskforce/CGreport.ps.Z

Probabilistic algorithms for efficient grasping and fixturing

Marek Teichmann*

Abstract

Given an object with n points on its boundary where fingers can be placed, we give algorithms to select a "strong" grasp with a minimal number c of fingers (up to a logarithmic factor) for several measures of goodness. Along similar lines, given an integer c, we find the "best" $\kappa c \log c$ finger grasp for a small constant κ. Furthermore, we generalize existing measures for the case of frictionless assemblies of many objects in contact. Depending on the measure, the algorithms run in expected time $O(c^2 n^{1+\delta})$ or $O((nc)^{1+\delta} + c^4 \log n \log^3 c)$. Here δ is any positive constant. This setting generalizes to higher dimensions in the context of finding sets of fixtures. These problems translate into a collection of *convex set covering* problems. We are given a convex set L, and a set of points U with $L \subset \operatorname{conv} U$ in dimension d. There are two basic questions: (1) what is the smallest subset C of U or *cover* with $L \subset \operatorname{conv} C$, and (2) given an integer c, what is the largest λ with $\lambda L \subset \operatorname{conv} C$ among all $C \subset U$ of c points. We present an algorithmic framework which handles these problems in a uniform way and give approximation algorithms for specific instances of L including convex polytopes and balls. It generalizes an algorithm for polytope covering and approximation by Clarkson [Cla93] in several different directions: we show it can be used not only for minimizing cover size, but also maximizing the scaling factor λ (see above), and further more it is valid for smaller cover sizes than previously possible, with appropriate modifications.

1 Introduction

Consider an idealized robot hand, consisting of several independently movable force-sensing fingers; this hand is used to grasp a rigid object B. Each finger contacts the object only at one point on B and can apply a positive force. We assume that at that point the normal to B is unique, and that the contact is frictionless. We wish to find a *grasp*: a set of points on the boundary of B. The fingers will then apply forces at these points to grasp the object. In general, we want the number of fingers to be small. Another desirable characteristic of a grasp is that by varying these forces within certain limits, we can resist an arbitrary external force-torque or *wrench* of the largest possible magnitude. See for example [MNP90, MSS87, FC92] for a more detailed description.

Most existing grasp synthesis algorithms either do not attempt to optimize the grasps found with respect to grasp strength, or do so for very specific situations such as small numbers of fingers, or planar objects. The algorithmic situation for fixturing assemblies is similar. Here we address both of these issues and present an algorithm which works in general dimension for an aritrary number of fingers.

We first describe the geometric formulation of grasping and fixturing theory, and grasp *efficiency* measures. We generalize these measures for the case of fixturing, and then exhibit a collection of related geometric optimization problems which arise, along with some connections to existing results in geometry.

1.1 Grasping Theory

The finger–body contacts being frictionless, a finger can only apply force $\mathbf{f}$ on the body in the direction of the inward pointing normal $\mathbf{n}(\mathbf{p})$ at a point $\mathbf{p}$ on the boundary of B. With each point $\mathbf{p}$ we associate a six-dimensional force-torque:

$$\mathbf{\Gamma}(\mathbf{p}) = [\mathbf{n}(\mathbf{p}), \mathbf{p} \times \mathbf{n}(\mathbf{p})].$$

This represents the effect of a unit force applied at $\mathbf{p}$ in the direction of $\mathbf{n}(\mathbf{p})$. For an illustration of the wrench map for planar objects, see Figure 1. We assume that we can select the grasp points from a finite set S of n points on the boundary of B. In practice this restriction is circumvented by providing a grid of points on the polygon faces. Thus we obtain an n-point set $U = \mathbf{\Gamma}(S)$ in force-torque space.

For a set $G \subset S$ of c points, we call G a c-finger *closure grasp* if the interior of conv $\{\mathbf{\Gamma}(\mathbf{p}) : \mathbf{p} \in G\}$ contains the origin o. It is shown in [MSS87] that for all but a certain class of objects having a boundary defined by an *exceptional* surface (a surface of revolution, for example) there is such a set of size $c \le 12$. For polyhedral objects [MSS87] give an algorithm for find such a grasp in linear time. However there is no guarantee on the "quality" of the grasp. In [KMY92] such a measure of *efficiency* is proposed. It measures the amount of external force and torque that can be resisted by applying at most a unit force distributed among the grasp

*Courant Institute, New York University, New York, NY 10012 USA. Supported by an NSF Grant CDA-9018673 and by FCAR and NSERC, Canada.

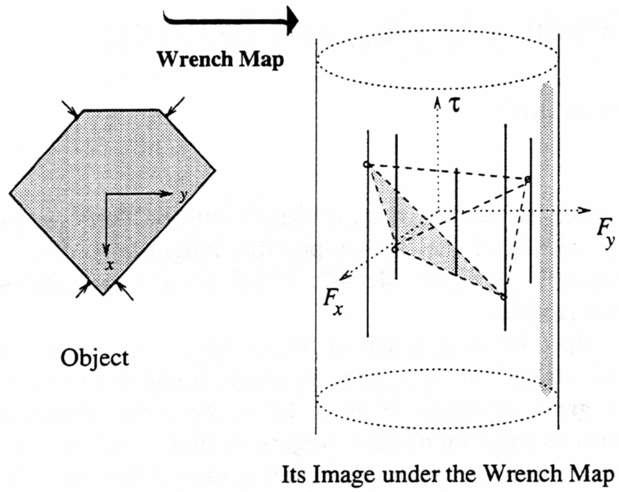

Figure 1: The wrench map.

points. This is measured by the radius of the largest ball, centered at the origin and contained in conv $\mathbf{\Gamma}(G)$. See also [FC92].

Let $\mathcal{B}(\rho)$ be the ball of radius ρ centered at the origin. For a set $A \subset \mathbb{R}^d$, denote the largest radius of such a ball in conv A by $r_d(A)$ and call it the *residual* radius of A. When the dimension is given by the context we shall simply write $r(A)$. Also denote by ∂B the boundary of the object B. The efficiency of a grasp G is then $r(\mathbf{\Gamma}(G))$. Note that it cannot be greater than $r(\mathbf{\Gamma}(\partial B))$. It is also of interest (see section 1.3) to replace the ball by a certain convex polytope.

We would like to obtain grasps of high efficiency but with few fingers. These are two conflicting goals [KMY92], so in this paper we provide algorithms for optimizing one quantity or the other.

1.2 Optimization problems.

At this point let us describe the setting for the optimization algorithms in their full generality. Let $\rho(\cdot)$ be some grasp efficiency measure, generalized to higher dimension (see below). It has an associated geometric object L containing o, which can be scaled about o. We define

$$\rho(U) = \max\{\lambda : \lambda L \subseteq U\}.$$

For $\rho(\cdot) = r(\cdot)$, $L = \mathcal{B}(1)$ in the appropriate dimension. Let a set of points C be a *cover* for L if $L \subseteq$ conv C. One class of problems is: for a desired grasp efficiency ρ_0, select a smallest possible set G out of n points on ∂B such that $\rho(\mathbf{\Gamma}(G)) \geq \rho_0$. This translates to the following purely geometric problem (valid in any dimension): let U be a set of n points in $\mathbb{R}^d$ such that the origin is contained in the interior of its convex hull.

> [**MinCover-L**]: Given ρ_0, we wish to find the smallest set $C \subset U$ or *cover* of size c^* with $\rho(\text{conv } C) \geq \rho_0$.

Since this problem is difficult (see section 2), we will solve a corresponding approximation problem: that of finding a cover of size $O(c^* d \log c^*)$. We will call this a $d \log c^*$-approximation of the optimal cover.

The companion problem is: given a maximum number of fingers c, which grasp G of size c maximizes $\rho(\mathbf{\Gamma}(G))$? The geometric version of this problem is:

> [**MaxScale-L**]: Given an integer c, find the set $C \subset U$ of size c which maximizes $\rho(C)$. Let $\rho^*(c)$ be this maximum.

In the approximation version, we ask for a $d \log c^*$-approximation of the best cover C. In the sequel, we will replace L by various convex sets.

We present an algorithmic framework derived from an algorithm by Clarkson [Cla93] for polytope covering that yields efficient randomized approximation algorithms for the above problems for various types of the set L. In fact, this approach works for any set L for which we can determine whether it lies entirely on one side of a query hyperplane. When L is a ball centered at the origin, the corresponding optimization problems will be **MaxScale-B** and **MinCover-B**, and when L is a polytope containing the origin, "**-L**" will be replaced by "**-P**".

Let $\gamma = 1/\lfloor d/2 \rfloor$ and δ be any positive constant. The approximation versions of **MaxScale-B** and **MinCover-B** can be solved in expected time $O((n^{1+\delta} + (nc)^{1/(1+\gamma/(1+\delta))}) + c\log(n/c)(c \log c)^{\lfloor d/2 \rfloor})$ when d is fixed, using sophisticated data structures. Here c represents the optimal cover size for **MinCover-B** and the desired cover size for **MaxScale-B**. In both cases a cover of size $4cd \log c$ is returned. For a polytope P of ℓ vertices, the **MinCover-P** $d \log c$–approximation problem was already solved by Clarkson's algorithm in expected time $O(n^{1+\delta} + c\ell^{1+\delta} + c(\ell c)^{1/(1+\gamma/(1+\delta))} + (nc)^{1/(1+\gamma/(1+\delta))})$ or $O(\ell c \log c + n)c \log(n/c)$ using a simpler version of the algorithm. In the same time bound, we can solve the approximation version of **MaxScale-P** (with c an input parameter), and **MinCover-P** (with c the optimal cover size.)

1.3 Further applications.

Other efficiency measures closely related to the previous one have been proposed, see for example [FC92] for a measure based on Minkowski sums instead of convex hulls. Yet another measure was proposed by W. Meyer and was described in [Mis94] – see this paper for a survey of grasp efficiency measures. Imagine an adversarial external finger capable of applying a force of an arbitrary magnitude on the grasped object. Define $r_{nasty}(G)$ to be the magnitude of the least amount of force the "nasty" finger must apply to break the grasp. Then $r_{nasty}(G) = \max\{\lambda : \lambda\mathbf{\Gamma}(\partial B) \subseteq \text{conv } -\mathbf{\Gamma}(G)\}$. It turns out that the image of any set $S \subset \partial B$ by $\mathbf{\Gamma}$ is a set of points in non-strictly-convex position, that is they all lie on the boundary of conv $\mathbf{\Gamma}(S)$. In addition, if the object B is polyhedral, then $\mathbf{\Gamma}(\partial B)$ is a set of two-dimensional polygons in $\mathbb{R}^6$, hence conv $\mathbf{\Gamma}(\partial B)$ is also polyhedral [Tei95]. The corresponding optimization problems are then simply **MinCover-P** and **MaxScale-P**.

Fixturing. We now consider the case of grasping several objects that are possibly in contact with each other, using a set of fingers, or *fixture elements*. This problem arises in manufacturing where many assembly tasks require a set of contacting objects to be held firmly. Fixture elements or *fixtures* are positioned in contact with the objects to achieve this. Often there is only a finite set of possible placements due to the construction of the workholding table [ZGW94]. In this setting, forces applied to an object are due both to contacts between objects, and between objects and fixtures. As in grasping, it is desirable to use a small number of fixtures, and/or limit the forces they must apply on the objects. The previous framework generalizes to this case by essentially concatenating the force-torque vectors for each object and working in $\mathbb{R}^{6k}$ where k is the number of objects [BMK94].

Let $\mathbf{m}_i$ be the center of mass of object i. In the following definitions 0 is the 6 dimensional zero vector. For each object B_i and contact point $\mathbf{p}$ let $\mathbf{\Gamma}_i(\mathbf{p}) = [\mathbf{n}(\mathbf{p}), (\mathbf{p} - \mathbf{m}_i) \times \mathbf{n}(\mathbf{p})]$. A fixture applied at $\mathbf{p}$ to B_i generates the generalized force-torque

$$\mathbf{\Gamma}_*(\mathbf{p}) = [\underbrace{0, \ldots, 0}_{i-1}, \mathbf{\Gamma}_i(\mathbf{p}), 0, \ldots, 0] \in \mathbb{R}^{6k}.$$

The i-th position contains $\mathbf{\Gamma}_i(\mathbf{p})$. A contact between object B_i and object B_j generates

$$\mathbf{\Gamma}_*(\mathbf{p}) = [\underbrace{0, \ldots, 0}_{i-1}, \mathbf{\Gamma}_i(\mathbf{p}), \underbrace{0, \ldots, 0}_{j-i-1}, \mathbf{\Gamma}_j(\mathbf{p}), 0, \ldots, 0].$$

In this context force-torque closure can be defined analogously to the previous one-object case: a set G of fixtures (i.e. of these generalized force-torque vectors) is a force/torque closure fixture set if and only if $\mathbf{0}$ is in the interior of conv $\mathbf{\Gamma}_*(G)$.

Fixture quality measures. We now introduce new grasp efficiency measures for the case of several objects. Informally the goal again is to keep the boundary of conv $\mathbf{\Gamma}_*(G)$ bounded away from the origin for all objects. In fact the above grasp efficiency measures generalize quite naturally to this case, but we work in dimension $6k$ (or $3k$ for planar objects) instead of $d = 6$ (or 3). The definition of residual radius still applies. Hence the measure will be defined as $r_{6k}(\text{conv } \mathbf{\Gamma}_*(G))$. As is the case in grasping a single object, here we assume that the sum of the magnitudes of all wrenches due either to the fingers or inter-object contacts must be bounded by 1. If a generalized wrench $W \in \mathbb{R}^{6k}$ is outside of conv $\mathbf{\Gamma}_*(G)$, then W cannot be expressed as a convex combination of finger and inter-object contact wrenches, and the fixture is broken if W is applied to the assembly, i.e. the component of W corresponding to each object is applied to that object.

Other geometric objects can be used for L with slightly different geometric interpretations. For example, consider the nasty finger measure. For the inscribed set L, we simply take the direct product $\bigotimes_i$ conv $\mathbf{\Gamma}(\partial B_i)$. For the set U, we take $-L$. The corresponding optimization problems are **MaxScale-P** and **MinCover-P**.

2 Related Results

Exact algorithms for the problems mentioned in the previous section that run in polynomial time seem unlikely. In fact problems similar to these have been shown to be NP-hard [BMK94] or NP-complete [DJ90]. We therefore consider approximation algorithms for the **MinCover** and **MaxScale** problems. The **MinCover-P** problem arises in a dual form in the context of separating two nested polyhedra [MS92, Cla93]. These problems can in turn be cast as **Hitting Set** problems [MS92]. There is also a deterministic analogue of Clarkson's algorithm which solves this problem and which provides a $O(d \log c)$-approximation [BG94] in time $O\left(n^{\lfloor d/2 \rfloor} + c^d n \log^d(dc)\right) c \log(n/c)$.

It is easy to verify that these hitting set techniques also apply to our **MinCover-B** problem as well as more general versions. It is unclear however whether these techniques can be applied to yield approximation algorithms for **MaxScale-B** or **MaxScale-P**, as they depend on the fact that the set L is fixed.

For the **MaxScale-B** problem, Kirkpatrick *et al.* [KMY92] give an algorithm that finds a cover C of size c containing a ball of radius $r(C) = \left[1 - 3d\left(2d^2/c\right)^{2/(d-1)}\right] r(U)$, for $n \geq c \geq 13^d d^{(d+3)/2}$ in time $O(\text{LP}(n, d)c)$. The radius found is almost optimal for that cover size. Here $\text{LP}(n, d)$ is the time required to solve a linear program of size n and dimension d. Currently the best deterministic algorithm runs in $O(d^{7d+o(d)}n)$ time [CM93] and the best randomized algorithm in time $O(d^2 n + e^{O(\sqrt{d \log d})})$. See [Gol95] for this bound and a recent survey. Unfortunately this result applies only for large cover sizes.

3 The Computational Framework

Consider a set U of n points in $\mathbb{R}^d$, and a convex set L with $0 \in L \subset \operatorname{conv} U$. We will use a routine FIND COVER which, for a given cover size c outputs a cover of L of size $4cd \log c$, if a cover of size c exists, otherwise it fails. Let c^* be the size of the smallest cover $C \subset U$ for L. If we wish to find c^* for fixed, un-scaled L, we use OPTIMAL COVER. In OPTIMAL COVER, we simply call FIND COVER with cover sizes $c = (\frac{5}{4})^i$, with $i = \lg d, \lg d + 1, \ldots$. This loop finds an approximation of c^* up to a factor of 5/4, without increasing the asymptotic running time.

The algorithm FIND COVER is defined below. It takes as input the desired cover size c, and uses another routine FIND BAD FACET. Given a half space h with a positive side, FIND BAD FACET determines whether L lies entirely in the positive half-space defined by h. In fact for the **MaxScale** problems we will need more: given a direction u, we will need the supporting hyperplane for L with normal u which is the furthest in that direction.

The algorithm goes as follows. We repeatedly take a random sample R, of expected size $s = 4cd \log c$, and test whether L (or some scaled version of L) is contained in conv R using FIND BAD FACET.

```
FIND COVER
Input: Size c of desired optimal cover;
       Maximum number I of iterations.
Output: Cover of size c4d log c.

1.   s = c4d ln c, {the expected size of cover}
2.   for all p ∈ U, let w_p = 1
3.   repeat for I successful iterations:
4.        Choose R ⊂ U at random. (see text)
5.        FIND BAD FACET F of conv R.
6.        if no bad facet, return R.
7.        Let U_F = points of U seeing F
8.        if w(U_F) ≤ w(U)/(kc) then
9.             for all p ∈ U, let w_p = 2w_p
                    {reweight}
10.            else {not a successful iteration}
end{FIND COVER}.
```

Let k be a constant to be specified later. Call an iteration of the loop in FIND COVER *successful* if the weights were doubled, *i.e.* if $w(U_F) \le w(U)/(kc)$. We also require that $|R| \approx 4cd \ln c$. This can be ensured by taking new random samples if necessary. This additional requirement does not change the expected asymptotic running time.

If a cover of size c is not found in the number of iterations specified, FIND COVER fails; otherwise, it returns a cover of expected size $4cd \log c$. This number is chosen in Lemma 3.3 to guarantee success if a cover of size c exists. The random selection of R is done by picking each point p of U independently with probability $\Pr(p) = 1 - (1 - w_p/w(U))^s \le s w_p / w(U)$. The expected size of R is $\sum_{p \in U} \Pr(p) \le \sum_{p \in U} s \frac{w_p}{w(U)} = s$. Finally the heart of the work is done in FIND BAD FACET whose variations are described in the next section.

The correctness of the general algorithm follows from a series of lemmas. The following lemmas were shown in [Cla93] for L a convex polytope, but the proofs do not use the fact that L is polyhedral and actually apply to any set L. We state them here for completeness, and also to bound some constants explicitly.

Lemma 3.1 (Clarkson) *Let L be any convex set not contained in* conv R, *with a point $p \in L$ on the negative side of some facet F of* conv R. *Let U_F be the set of points of U that see F. Then there is a point of the optimal cover C of L among the points of U_F.*

This lemma is the basis of this algorithm, and its derivatives. It essentially says that by finding a set U_F, we have gained some information about C since one of its members must be in the relatively small set U_F. We restrict the size of this set to be bounded by the condition $w(U_F) \le w(U)/(kc)$.

We say that a facet F of a polytope P is visible from a point q if for every $p \in F$, the segment $\overline{pq}$ does not meet P. Here P will be conv R. Following [Cla93], we define an *L-facet* to be a facet visible from of point of L. Then we have the following lemma, which is a slightly modified version of Clarkson's lemma 2.2.

Lemma 3.2 *Given that an L-facet F is found, the probability that an iteration of* FIND COVER *will be successful (i.e. the set U_F satisfies $w(U_F) \le w(U)/(kc)$), is at least 1/2.*

Proof. The proof follows Clarkson's closely, but is addition we note that the probability that an iteration of FIND COVER will be not be successful is bounded above by 1/2 for $k = 2, d > 2$ and $c > 43$; or for $k = 1.501$, $d > 2$ and $c > 6$. See [Tei95] for details. □

Finally, to obtain a bound on the running time, we need to bound the number of iterations of the loop in FIND COVER. Again this lemma is a slight variation of Clarkson's and a similar lemma also appears in [BG94].

Lemma 3.3 *The number of successful iterations of the loop in* FIND COVER *before a cover is found is bounded above by $\frac{2k}{2k-3} c \lg(n/c)$ which is $4c \lg(n/c)$ for $k = 2$, $1501 c \lg(n/c)$ for $k = 1.501$.*

Proof. See [Tei95] for details. □

This implies together with Lemma 3.2 that the expected number of iterations of the loop in FIND COVER is $O(c \lg(n/c))$. It is interesting to note the tradeoff in the constants between the constant in the running time, and the constant k, which in turn determines the minimum cover size for which these lemmas are valid.

Note also that these lemmas hold for small values of c, which makes the algorithm useful for small size covers. Of course $c > d$, since we cannot have a cover of any smaller size unless the input is highly degenerate.

4 Particular Measures

In this section we describe several versions of FIND BAD FACET and give the corresponding running times of the entire algorithms.

MinCover-B:
We are given a ball $\mathcal{B}(r)$ of radius r centered at the origin, and we would like to find an approximation of its minimal cover among the points of U. To do this, we use OPTIMAL COVER, but we define FIND BAD FACET to test whether the ball is contained in the convex hull of the random sample R. This can be done by computing the convex hull of R and finding a facet that is at a distance of less than r to the origin. This takes time $O(|R|^{\lfloor d/2 \rfloor})$ for fixed $d \geq 4$ (or $O(|R| \log |R|)$ if $d \leq 3$) [Cha93]. For variable d, we can use the algorithm in [AF92], which finds f facets of the convex hull of n points in dimension d in time $O(ndf)$.

Let $\gamma = 1/\lfloor d/2 \rfloor$ and δ be any positive constant. Since the expected number of iterations is $O(c \log(n/c))$ and the expected size of R is $4cd \ln c$, the expected running time of OPTIMAL COVER for $\mathcal{B}(r)$ is:

$$O\Big(n^{1+\delta} + (nc)^{1/(1+\gamma/(1+\delta))} + c\log(n/c)(c\log c)^{\lfloor d/2 \rfloor}\Big)$$

for fixed d, or

$$O\Big(ncd\log(n/c) + cd^2 2^d \log(n/c)(4cd\log c)^{\lfloor d/2 \rfloor + 1}\Big)$$

for any d (and non-degenerate convex hull of the covers) using the result in [AF92] and the crude bound on the complexity of the convex hull of $d2^d v^{\lfloor \frac{d}{2} \rfloor}$ (derived from the Upper Bound Theorem [MS71].)

MinCover-P:
The geometric object to be covered is a fixed polytope P with ℓ vertices containing the origin. Clarkson's original algorithm solves this approximation problem for fixed d in expected time $O(n^{1+\delta} + c\ell^{1+\delta} + c(\ell c)^{1/(1+\gamma/(1+\delta))} + (nc)^{1/(1+\gamma/(1+\delta))})$ or $O(\ell c \log c + n)c\log(n/c)$ using a simpler version of the algorithm. Let the *positive* side of a facet of a polytope be the side containing the polytope. Here FIND BAD FACET finds a facet F of conv R such that there is a vertex of P on the side of F not containing the origin, *i.e.* on its negative side, if P is not contained in conv R. This is done using linear programming. Each such step can be done in $O(LP(4cd\log c, d))$ time. For variable d, the entire algorithm runs in expected time

$$O\Big(ncd\log(n/c) + \ell c^2 d^3 \log c \log(n/c) + \ell c\log(n/c) e^{O(\sqrt{d\log d})}\Big).$$

For fixed d, we can also use linear programming queries using the recent batched version of Chan [Cha95] to improve slightly the running time for large l. If $\ell = O(n)$, a much simplified expression of the running time is $O(n^{1+\delta} + c^2 n \log n \log c)$.

MaxScale-B:
Let $r^*(s)$ be the radius of the largest ball centered at the origin and contained in a cover of size s. In this problem we are given a desired cover size c, and we would like to find $r^*(c)$. We will not use OPTIMAL COVER but skip directly to FIND COVER. Here again we get only an $O(d\log c)$ approximation to this cover. The corresponding version of FIND BAD FACET will be to use FIND BAD FACET as for the **MinCover-B** problem, but always return the facet *closest* to the origin, *i.e.* never say that a ball of some radius has been covered. We also remember the maximum distance to the origin of the closest facet at each iteration, and the corresponding cover. Here again we do only $4c\lg(n/c)$ successful iterations, *i.e.* $O(c\log(n/c))$ calls to FIND BAD FACET. This will guarantee that the version of FIND COVER for **MaxScale-B** finds a cover of size at most $4cd\ln c$ containing a ball of radius $r^*(c)$. This can be seen as follows. By Lemma 3.3 we are certain to find a cover if it exists. Hence if $\mathcal{B}(r^*(c))$ is the largest ball in covers of size c, a cover of size $4cd\ln c$ containing $\mathcal{B}(r^*(c))$ must be found. The running time is the same as for the **MinCover-B** approximation.

MaxScale-P:
The solution to this problem is very similar to the previous case. Here FIND BAD FACET finds the largest possible scaling factor λ such that $\lambda P \subseteq$ conv R using linear programming queries as for **MinCover-P**. We summarize both results in

Theorem 4.1 *The versions of* FIND COVER *for* **MaxScale-B** *(resp.* **MaxScale-P***) find a cover of size at most* $4cd\ln c$ *containing a ball of radius* $r^*(c)$ *(resp.* P *scaled by* $\lambda^*(c)$*) in time identical to their* **MinCover-B** *and* **-P** *counterparts.*

Similar techniques can be used for other sets L.

4.1 Speeding up MinCover-B and MaxScale-B

To avoid taking the expensive convex hull of R to find a facet closest to the origin, we can *approximate*

$r^*(c)$. We sample conv R by performing the ray shooting queries described above for the case of polytopes, along equally distributed rays on the set of directions (or points on a sphere.) We can use the upper bound contruction of [KMY92] to find such a set of points or rays. We get

Theorem 4.2 *The optimal radius (for a cover of size c) for both* **MaxScale-B** *and* **MinCover-B** *can be* $1-\epsilon$ *approximated in expected time* $O(n^{1+\delta} + (nc)^{1/(1+\gamma/(1+\delta))} + \left[\frac{2d^2}{17^{\frac{d-2}{2}}}\left(\frac{1}{\epsilon}\right)^{\frac{d-2}{2}}\right] c^2 \log\frac{n}{c}\log c)$ *using a simplified version of expression for the running time.*

5 Concluding remarks

It is possible to extend the framework to the following case: instead of a set of points U, we have a set of planar polygons. This corresponds to the situation in grasping where we allow fingers to be placed anywhere on ∂B. However we no not know how to analyze the performance of this modification in terms of optimality of cover size.

On a different note, the physical interpretation of the fixturing metrics do not seem to be as natural as the corresponding ones for a single object. Finally, we are in the process of implementing this algorithm. This might shed some light on whether the special structure of the point sets in our application has any impact on the efficiency.

We wish to thank Bud Mishra, Ken Clarkson and Chee Yap for helpful discussions on this subject.

References

[AF92] D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete Comput. Geom.*, 8:295–313, 1992.

[BG94] H. Brönnimann and M. T. Goodrich. Almost optimal set covers in finite vc-dimension. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 293–302, 1994.

[BMK94] D. Baraff, R. Mattikalli, and P. Khosla. Minimal fixturing of frictionless assemblies: Complexity and algorithms. Report CMU-RI-TR-94-08, The Robotics Inst., Carnegie Mellon University, Pittsburgh, PA, 1994.

[Cha93] B. Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete Comput. Geom.*, 10:377–409, 1993.

[Cha95] Timothy M. Y. Chan. Output-sensitive results on convex hulls, extreme points, and related problems. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 10–19, 1995.

[Cla93] Kenneth L. Clarkson. Algorithms for polytope covering and approximation. In *Proc. 3rd Workshop Algorithms Data Struct.*, volume 709 of *Lecture Notes in Computer Science*, pages 246–252, 1993.

[CM93] B. Chazelle and J. Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension. In *Proc. 4th ACM-SIAM Sympos. Discrete Algorithms*, pages 281–290, 1993.

[DJ90] G. Das and D. Joseph. The complexity of minimum convex nested polyhedra. In *Proc. 2nd Canad. Conf. Comput. Geom.*, pages 296–301, 1990.

[FC92] C. Ferrari and J. Canny. Planning optimal grasps. In *1992 IEEE International Conference on Robotics and Automation*, pages 2290–2295, Nice, France, 1992.

[Gol95] M. Goldwasser. A survey of linear programming in randomized subexponential time. *SIGACT News*, 26(2):96–104, 1995.

[KMY92] D. Kirkpatrick, B. Mishra, and C.-K. Yap. Quantitative Steinitz's theorems with applications to multifingered grasping. *Discrete Comput. Geom.*, 7:295–318, 1992.

[Mis94] B. Mishra. Grasp metrics: Optimality and complexity. In K. Goldberg, D. Halperin, J.C. Latombe, and R. Wilson, editors, *Algorithmic Foundations of Robotics*, pages 137–166. A.K. Peters, 1994.

[MNP90] Xanthippi Markenscoff, Luqun Ni, and Christoph H. Papadimitriou. The Geometry of Grasping. *The International Journal of Robotics Research*, 9(1), February 1990.

[MS71] P. McMullen and G. C. Shephard. *Convex Polytopes and the Upper Bound Conjecture.* Cambridge University Press, Cambridge, England, 1971.

[MS92] J. S. B. Mitchell and S. Suri. Separation and approximation of polyhedral surfaces. In *Proc. 3rd ACM-SIAM Sympos. Discrete Algorithms*, pages 296–306, 1992.

[MSS87] B. Mishra, J.T. Schwartz, and M. Sharir. On the Existence and Synthesis of Multifinger Positive Grips. *Algorithmica*, 2:541–558, 1987.

[Tei95] M. Teichmann. *Grasping and Fixturing: A Geometric Analysis and an Implementation.* Ph.D. thesis, New York Univ., New York, NY, 1995.

[ZGW94] Y. Zhuang, K.Y. Goldberg, and Y.-C. Wong. On the existence of modular fixtures. In *IEEE International Conference on Robotics and Automation*, pages 543–549, San Diego, CA, 1994.