Probabilistic Algorithms for Efficient Grasping and Fixturing

Marek Teichmann* Bud Mishra†

Abstract

Given an object with n points on its boundary where fingers can be placed, we give algorithms to select a "strong" grasp with the least number c of fingers (up to a logarithmic factor) using 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. In addition, we generalize existing measures for the case of frictionless assemblies of many objects in contact. We also give an approximation scheme which guarantees a grasp quality close to the overall optimal value where fingers are not restricted to preselected points. These problems translate into a collection of *convex set covering* problems where we either minimize the cover size or maximize the scaling factor of an inscribed geometric object L. 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. The framework generalizes an algorithm for polytope covering and approximation by Clarkson [Cla93] in two different ways. Let $\gamma = 1/\lfloor d/2 \rfloor$, where d is the dimension of the Euclidean space containing L. For both types of problems, when L is a polytope, we get the same expected time bounds (with a minor improvement), and for a ball, the expected running time is $O(n^{1+\delta} + (nc)^{1/(1+\gamma/(1+\delta))} + c\log(n/c)(c\log c)^{\lfloor d/2 \rfloor})$ for fixed d, and arbitrary positive δ. We improve this bound if we allow in addition a different kind of approximation for the optimal radius. We also give bounds when d is not a constant.

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 with piece-wise smooth boundary ∂B. In the rest of this paper, B is always assumed to denote this object. Each finger contacts the object at only 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; the force is applied along the normal to B at the point of contact. We do not assume B to be convex. 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 a certain magnitude. See for example [MNP90, MSS87] for a more detailed description. Other recent references on this topic include [Ngu87, FC92, PSS+95, MLS94].

*MIT Lab. for Computer Science, Cambridge, MA, USA. This work was done while at the Courant Institute, New York University. Supported by an NSF Grant CDA-9018673 and by FCAR and NSERC, Canada.

†Courant Institute, New York University, NY, USA. Supported by an NSF Grant CDA-9018673, IRI-9414862 and ONR grant N00014-96-1-0400.

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 a small number 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 arbitrary 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. This paper concentrates on the algorithmic aspects of these problems.

1.1 Grasping Theory

In this section, we describe the case of a three–dimensional body B. Assuming finger–body contacts are frictionless, a finger can only apply force $\mathbf{f}$ at a point $\mathbf{p}$ on the boundary of B in the direction of the inward pointing normal $\mathbf{n}(\mathbf{p})$ at $\mathbf{p}$. This force also applies a torque $\mathbf{p} \times \mathbf{n}(\mathbf{p})$ on B, where $\times$ denotes the vector cross product. 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 $\mathbf{n}(\mathbf{p})$. We assume that we can select the grasp points from a finite set S of n points on the boundary of B. Thus we obtain a set $U = \mathbf{\Gamma}(S) = \{\mathbf{\Gamma}(\mathbf{p}) : \mathbf{p} \in S\}$ in force-torque space of n points. For an illustration of the wrench map for planar objects, see Figure 1.

For a set $A \subset \mathbb{R}^d$, let conv A denote the convex hull of A, and ∂conv A denote the boundary of the convex hull. For a set $G \subset S$ of c points, we call G a c-finger *closure grasp* if the interior of conv $\mathbf{\Gamma}(G)$ 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 $G \subset \partial B$ of size $c \leq 12$. For polyhedral objects there are linear time algorithms to find such a closure grasp. However there is no guarantee on the "quality" of the grasp. In [KMY92] such a measure of *efficiency* is proposed. It is the amount of external force and torque that can be resisted by applying at most a unit force distributed among the grasp points. This quantity corresponds in force–torque space to the radius of the largest ball, centered at the origin and contained in conv $\mathbf{\Gamma}(G)$.

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 ρ such that $\mathcal{B}(\rho) \subseteq$ conv A by $r_d(A)$ and call it the *residual* radius of A. When the dimension is clear from the context we shall simply write $r(A)$. 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 2) to replace the ball by a certain convex polytope. In practice, grasping points will not be selected from edges and vertices, and we assume that this is the case for the allowable grasp point set S. However, for technical reasons, when we write $\mathbf{\Gamma}(\partial B)$, we in fact treat ∂B as a collection of closed polygons, each having its own normal. This normal is extended to the edges and vertices of each polygonal face by continuity. The selection of a point from ∂B also implicitly requires a selection of a normal, when such a point is on an edge or a vertex. We take this approach to be able to work with closed polygonal faces in later analysis.

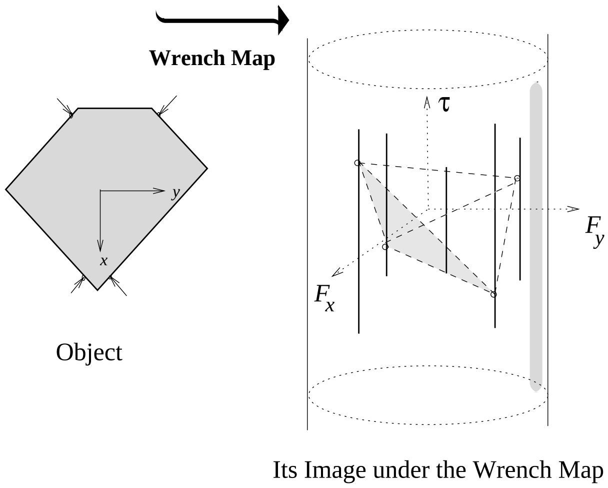

Figure 1: The wrench map.

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 in Grasping

At this point let us describe the setting for the optimization algorithms in full generality. Given a set $L \subset \mathbb{R}^d$ whose convex hull contains the origin, and a set $C \subset \mathbb{R}^d$, we define a measure associated with L to be

$$\rho(C) = \max\{\lambda : \lambda L \subseteq \operatorname{conv} C\}.$$

In the context of grasping, $\rho(\cdot)$ will be our general grasp efficiency measure, which we shall specialize. For example, we might take $\rho(\cdot) = r(\cdot)$, where $L = \mathcal{B}(1)$ in the appropriate dimension. Let a set of points C be a *cover* for L if $L \subseteq \operatorname{conv} C$. Note that by convexity, we can replace L by its convex hull without affecting this property or the value of $\rho(\cdot)$. One class of problems can be stated as follows: for a desired grasp efficiency ρ_0, select a set $G \subset \partial B$ of smallest possible size with $\rho(\mathbf{\Gamma}(G)) \geq \rho_0$. We assume in this paper that the set S of possible points on ∂B out of which points of G are selected is finite; let its size be n. This grasping problem translates to the following purely geometric problem (posed in any dimension):

> [**MinCover-L**]: Given a positive number ρ_0, a convex set L containing the origin in its interior, and a set of n points U in $\mathbb{R}^d$ with $L \subset \operatorname{conv} U$, find the smallest set $C \subset U$ of size c^* with $\rho(C) \geq \rho_0$, *i.e.* such that C is a cover for $\rho_0 L$.

In the corresponding grasping problem which optimizes the efficiency measure $r(\cdot)$, $L = \mathcal{B}(1)$ and $U = \mathbf{\Gamma}(S)$. Since the general problem is difficult (see section 1.3), 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 can be stated as follows: given a number c of fingers, which grasp $G \subset \partial B$ of size c maximizes $\rho(\mathbf{\Gamma}(G))$? The geometric version of this problem is:

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

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 hyper-plane. Such a test has been called a *strong violation* oracle [GLS88] or *half-space emptiness query* [Mat92]. When L is the unit ball centered at the origin, the corresponding optimization problems will be called **MaxScale-B** and **MinCover-B**, and when L is a polytope containing the origin, "**-L**" will be replaced by "**-P**". The dimensions encountered will be $d = 6$ for objects in three dimensions, and $d = 3$ for planar objects. Even higher dimensions will be relevant for fixturing problems (section 2.2).

Let $\gamma = 1/\lfloor d/2 \rfloor$ (where d is the dimension of the space containing U) and δ be any arbitrarily small 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 $4cd \ln c$ is returned. For a polytope P of ℓ vertices, the **MinCover-P** $4d \ln c$–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. Here we use essentially the same algorithm. We show that in the same time bound, by modifying Clarkson's algorithm, we can solve the approximation version of **MaxScale-P** (with c an input parameter).

1.3 Related Geometric Results

Exact algorithms for the problems mentioned in the previous section that run in polynomial time seem unlikely. Since problems similar to these have been shown to be NP-hard [BMK94] or NP-complete [DJ90], we suspect our problems to be equally hard. However, it remains an open problem to characterize the exact complexity of our problems.

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]. In this problem, we are given two polyhedra L' and U' with $U' \subseteq L'$, and we want to find a separating polytope S' with $U' \subseteq S' \subseteq L'$ with small number of facets. If we ask for the smallest possible number of facets, this problem is NP-complete [DJ90]. Several approximation algorithms for this problem have been presented—both deterministic [DJ90, BG94], and randomized [Cla93]. Consider U' as the intersection of a set of half-planes. In the above-mentioned algorithms, S' is found by eliminating facets (half-planes) from U', while making sure that the new polytope is still contained in L'. We assume that all polytopes involved contain the origin, and we apply the point-plane *polarity* duality transformation with respect to the unit sphere. This duality maps a hyper-plane at distance r from the origin and with unit normal n to the point n/r and vice versa. It also maps $\mathcal{B}(r)$ to $\mathcal{B}(1/r)$ with an appropriate extension of the definition. Replacing U' and L' by their duals U and L respectively, the above separation framework amounts to covering L by the convex hull of a subset of the set of points U and this is exactly what we need.

Polytope covering problems can in turn be cast as **Hitting Set** problems [MS92]. We refer the reader to this paper for the details of this transformation. The algorithm in [MS92] can be used to provide an approximation algorithm for the **MinCover-P** problem (or its dual) using a greedy heuristic, with an approximation factor of $d \log n$. It runs in time $O(n^{d+1})$. 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 $O\left((n^{\lfloor d/2 \rfloor} + c^d n \log^d(dc))c \log(n/c)\right)$.

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, since the set system defined by these problems changes as L changes (scales). It would be interesting to analyze these changes.

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 $O(\mathrm{LP}(n,d)c)$. The radius found is almost optimal for that cover size. Here $\mathrm{LP}(n,d)$ is the time required to solve a linear programming problem of size n and in 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.

In section 2, we present some convexity properties of force–torque space, and describe additional applications of the algorithmic framework, including a generalization of the grasp quality measures to fixturing multiple objects. In section 3, the framework is described in detail and specialized to various measure in section 4. Section 5 presents a method for lifting the restriction on the placement of fingers. We conclude in section 6.

2 Further applications

2.1 Other grasp efficiency measures

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 an extensive 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 at some point on the object 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 $T \subset \partial B$ by $\mathbf{\Gamma}$ is a set of points in non-strictly-convex position, that is they all lie on the boundary of conv $\mathbf{\Gamma}(T)$. 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. These facts are shown in the remainder of this section. The corresponding optimization problems are then simply **MinCover-P** and **MaxScale-P** with $L = -\mathbf{\Gamma}(\partial B)$ and $U = \mathbf{\Gamma}(S)$, in the notation of section 1.2.

Lemma 2.1 *Let $X \subset \mathbb{R}^k$ be a finite set of distinct points in convex position. With each $x \in X$, is associated a compact set of points $Y(x) \subset \mathbb{R}^l$. Let $w_{xy} = [x, y] \in \mathbb{R}^{k+l}$ and $W = \{w_{xy} : x \in X, y \in Y(x)\}$. Then*

1. *$W \subset \partial\,\text{conv}\, W$.*
2. *If $Y(x)$ is finite for each $x \in X$ then W is in convex position in $\mathbb{R}^{k+l}$ iff $Y(x)$ is in convex position for each $x \in X$ in $\mathbb{R}^l$.*

Proof. (1) Consider a particular $x_0 \in X$. Since X is in convex position, there exists a hyperplane h in $\mathbb{R}^k$ which is a supporting hyperplane for X, contains x_0 and contains no other point of X. (The normal of h can be taken to be the sum of the outward pointing normals of the facets of conv X adjacent to x_0.) Say that h is defined by the equation $a \cdot x = b$ for a vector a in $\mathbb{R}^k$ and constant b, and is such that $a \cdot x_0 = b$ and $a \cdot x > b$ for $x \neq x_0$. Now let h' be the hyperplane in $\mathbb{R}^{k+l}$ defined by $[a, 0] \cdot w = b$. Then h' is a supporting hyperplane for W since the $w_{x_0 y}$ satisfy its equation for all $y \in Y(x_0)$, and $[a, 0] \cdot w_{xy} = a \cdot x > b$ for $x \neq x_0$. Note that $W \cap h' = \{w_{x_0 y} : y \in Y(x_0)\}$. We have (see [Grü67, p17]) $W \cap h' \subseteq \text{conv } (W \cap h') = (\text{conv } W) \cap h'$, hence $W \cap h' \subseteq \partial\text{conv } W$. The first part of the lemma follows.

(2) Consider a point $w = [x_0, y_0]$ of W. Since this point is in conv W, we can write $w = \sum_{x \in X, y \in Y(x)} \lambda_{xy} w_{xy}$, where $0 \leq \lambda_{xy} \leq 1$ and $\sum_{x \in X, y \in Y(x)} \lambda_{xy} = 1$. Now consider the projection $\pi: \mathbb{R}^{k+l} \to \mathbb{R}^k : w_{xy} \mapsto x$, and note that

$$x_0 = \pi(w) = \sum_{x \in X, y \in Y(x)} \lambda_{xy} \pi(w_{xy}) = \sum_{x \in X} \Big(\sum_{y \in Y(x)} \lambda_{xy} \Big) x.$$

Thus, as the points in X are in convex position, $\sum_{y \in Y(x)} \lambda_{xy} = 1$, for $x = x_0$ and is 0 for $x \neq x_0$. Since λ's are positive, we also have $\lambda_{xy} = 0$, for all $x \neq x_0$. Now let in addition $Y(x_0)$ be in convex position and consider the projection $\Pi_x: \mathbb{R}^{k+l} \to \mathbb{R}^l : w_{xy} \mapsto y$. We have

$$y_0 = \Pi_{x_0}(w) = \sum_{y \in Y(x_0)} \lambda_{x_0 y} \Pi_{x_0}(w_{x_0 y_0}) = \sum_{y \in Y(x_0)} \lambda_{x_0 y} y.$$

But this implies that $\lambda_{x_0 y} = 0$ for $y \in Y(x), y \neq y_0$ by the assumption that $Y(x_0)$ is in convex position. Thus W is in convex position.

The converse of this statement is trivial. Assume that we can express some $y_0 \in Y(x_0)$ as follows:

$$y_0 = \sum_{y \in Y(x_0)} \lambda_{x_0 y} y, \quad \lambda_{x_0 y} < 1.$$

Then, we have

$$w = [x_0, y_0] = \sum_{x \in X, y \in Y(x)} \lambda_{xy}[x, y] = \sum_{x \in X, y \in Y(x)} \lambda_{xy} w_{xy}, \quad \lambda_{xy} < 1.$$

□

Applying Lemma 2.1 to force-torque spaces gives us

Corollary 2.2 *A grasp G satisfies the condition*

$$\mathbf{\Gamma}(G) \subset \partial \operatorname{conv} \mathbf{\Gamma}(G).$$

Proof. Let $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^3$ denote the set of distinct unit normals defined by the points of G, i.e., $\{\mathbf{n}(\mathbf{p}_1), \ldots, \mathbf{n}(\mathbf{p}_m)\}$. Since $x_1, \ldots, x_n$ are on the unit sphere (and distinct), they are in convex position. Let $Y(x_i) = \{\mathbf{p} \times \mathbf{n}(\mathbf{p}) : \mathbf{p} \in G, \mathbf{n}(\mathbf{p}) = x_i\} \subset \mathbb{R}^l$, and let n_i be the cardinality of $Y(x_i)$. Thus $W = \{w_{x_i y_{ij}} = [x_i, y_{ij}] : i = 1, \ldots, n; j = 1, \ldots, n_i\} = \mathbf{\Gamma}(G)$. The lemma now follows from Lemma 2.1. □

When S is the set of boundary points of a polyhedral object, we have

Theorem 2.3 *When B is polyhedral with v vertices,* conv $\mathbf{\Gamma}(\partial B)$ *is polyhedral and has $O(v)$ vertices, and $\mathbf{\Gamma}(\partial B) \subset \partial \operatorname{conv} \mathbf{\Gamma}(\partial B)$.*

Proof. Let $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^3$ denote the set of distinct unit normals defined by the faces of B. That is, $x_i = \mathbf{n}(\mathbf{p})$ for some $\mathbf{p} \in F$, a face F of B. Since x_i's are on the boundary of a unit sphere they are in convex position. Let

$$Y(x_i) = \{\mathbf{p} \times \mathbf{n}(\mathbf{p}) : \mathbf{p} \in \partial B \text{ and } \mathbf{n}(\mathbf{p}) = x_i\}.$$

Define $W_i = \{[x_i, y] : y \in Y(x_i)\}$ and $W = \bigcup W_i$. Clearly, $W = \mathbf{\Gamma}(\partial B)$ and by Lemma 2.1 $\mathbf{\Gamma}(\partial B) \subset \partial \operatorname{conv} \mathbf{\Gamma}(\partial B)$, since x_i's are in convex position. Furthermore by using the linearity of $\mathbf{\Gamma}$ and the fact that W_i is the image of polyhedral faces of B under $\mathbf{\Gamma}$, we note that conv W_i themselves are polyhedral. Thus conv $W =$ conv $\bigcup W_i$ is also polyhedral.

Note further that every vertex of conv W can be identified with a vertex of conv W_i and hence to a face of B. Thus if a face F of B has v_F vertices of B, then it can contribute at most v_F vertices to conv W. Thus the total number of vertices of conv W is bounded by $\sum_{F \in F(B)} v_F$, where the sum is taken over the faces of B. This sum is easily seen to be $O(v)$. □

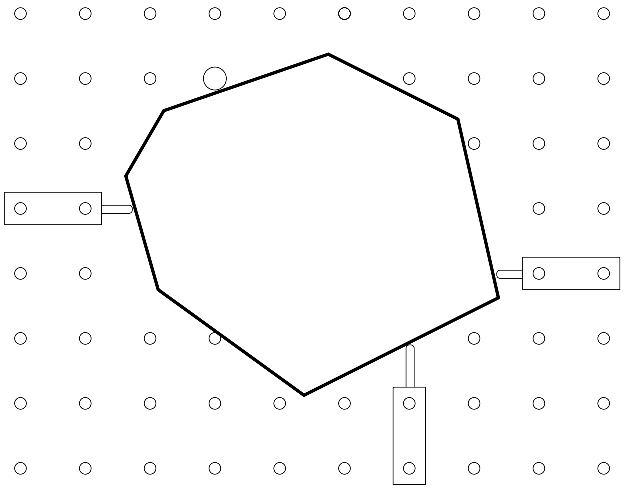

Figure 2: Fixturing a single object with 3 clamps and 1 locator.

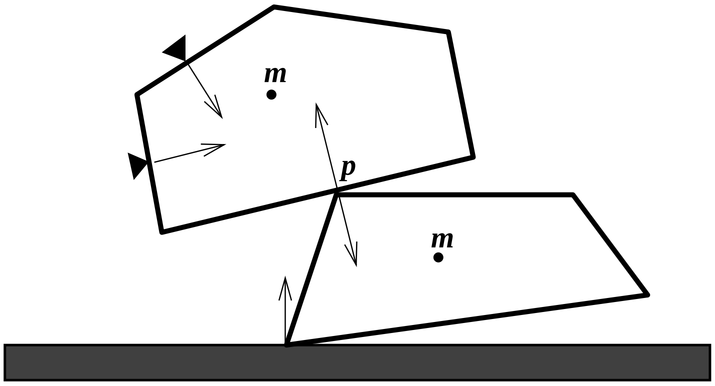

Figure 3: An assembly with fixtures.

2.2 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 *fixels* 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 [Mis91, ZGW94] (see Figure 2). In this setting, forces applied to an object are due both to contacts between objects, and between objects and fixtures (see Figure 3). 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 $\mathbf{p} \in B_i$ 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 $(j > i)$ generates

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

Selecting a fixture set G of fixels on the surface of the objects gives us a set of points $\mathbf{\Gamma}_*(G)$ in $\mathbb{R}^{6k}$. It is worth noting that as before this framework specializes to the case where the objects are planar; in this case, the force-torque space for each object is three-dimensional.

In this context force-torque closure can be defined analogously to the previous one-object case: a set $G \subset \bigcup_i \partial B_i \subset \mathbb{R}^3$ of fixtures is a force/torque closure fixture set if and only if the origin of $\mathbb{R}^{6k}$ is in the interior of conv $\mathbf{\Gamma}_*(G)$.

2.2.1 Fixture efficiency 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 $d = 6k$ (or $d = 3k$ for planar objects) instead of $d = 6$ (or $d = 3$). The definition of residual radius still applies. The corresponding efficiency measure will be defined as $r_{6k}(\text{conv}\, \mathbf{\Gamma}_*(G))$. Here, in the notation of section 1.2, L is the unit ball in $\mathbb{R}^{6k}$, and U is the image by $\mathbf{\Gamma}_*$ of points on the objects where fixels can be placed. 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 is bounded by 1 (see [Mis94]). 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.* if the component of W corresponding to each object is applied to that object.

We can also replace the ball by the direct product of (6 dimensional) balls of identical radius r (which we call $\mathcal{B}(r)$), one for each object, *i.e.* $\bigotimes_{i=1}^{k} \mathcal{B}(r) = \mathcal{B}^k(r)$. ($\otimes$ is defined for two sets $X \subseteq \mathbb{R}^s$ and $Y \subseteq \mathbb{R}^t$ by $X \otimes Y = \{(x, y) \in \mathbb{R}^{s+t} : x \in X, y \in Y\}$.) This measure applied to a fixture set G is then the largest scaling factor λ such that $\lambda \mathcal{B}^k(1) \subset \bigotimes_i \text{conv}\, \mathbf{\Gamma}_i(G \cap B_i)$ and the corresponding optimization problems are **MaxScale-B**^{k} and **MinCover-B**^{k}. The notation $G \cap B_i$ represents the fixels which apply to B_i. We note that this amounts to considering each object separately and taking the minimum of λ over all objects B_i of $r_6(G \cap B_i)$. Also, the assumption is now only that the sum of the magnitudes of all wrenches acting on *each* object is bounded by 1 ([Mis94]).

We can also extend the nasty finger measure to fixturing. Here we let the set L to be a scaled version of the direct product $-\bigotimes_i \text{conv}\, \mathbf{\Gamma}(\partial B_i)$. This construction is a special case of taking Minkowski sums of polytopes, and the result is still convex [Grü67]. The set U becomes $-L$. Alternately, if fixtures can be placed at only a finite number of points on B_i, one can replace B_i by this set of points. The corresponding optimization problems are **MaxScale-P** and **MinCover-P**. This measure is particularly useful when the fixturing is done for machining purposes. Here a "tool" is applied to the boundary and can apply potentially high forces on the object. Other force-torques due to acceleration and gravity can be neglected.

Other variations, using for example Minkowski sums instead of convex hulls, are possible and have slightly different physical interpretation, but do not change the situation qualitatively; see [Mis94] for a discussion. The application at hand will determine which measure will be used.

3 The Computational Framework

In this section, we describe the optimal cover-finding algorithms referred to previously[1]. Consider a set U of n points in $\mathbb{R}^d$, and a convex set L containing the origin and such that $L \subset \operatorname{conv} U$. The algorithm FIND COVER solves one class of problems: for a given cover size c, FIND COVER outputs a cover $C \subset U$ for L (or for a scaled version of L) of expected size $4cd \ln c$, if a cover of size c exists or fail otherwise. We will use versions of FIND COVER to solve the **MaxScale** problems.

Before describing the algorithm FIND COVER, we first need to introduce several geometric notions. Familiarity with Clarkson's paper [Cla93] might be helpful to the reader. Let P be a convex polytope. We say that a facet F of P is *visible* from a point q if for every $p \in F$, the segment $\overline{pq}$ does not meet P. Following [Cla93], we define an *L-facet* to be a facet visible from a point of L. We also orient facets so that their positive side contains the polytope P. Let $R \subset U$ be a potential cover for L. The following lemmas were shown in [Cla93] for L a set of points in convex position, but the proofs actually apply to any convex set L. As in [Cla93], we also assume that no $d+1$ points of U lie on the same hyperplane. To ensure this, we can use perturbation methods such as those described in [ECS97, EM90, Yap90]. In fact, such methods have been used in practice; see for example the first two methods cited above.

Lemma 3.1 (Clarkson) *Let L be any convex set not contained in* $\operatorname{conv} R$*, with a point of L on the negative side of some facet F of* $\operatorname{conv} R$ *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 Clarkson's algorithm, and its derivatives. It essentially says that by finding an L-facet F and the set U_F, we have gained some information about C since one of its members must be in the relatively small set U_F. Finding the facet F is done in FIND L-FACET and the method depends on L. We describe the variations of FIND L-FACET for different sets L in the next section.

The algorithm FIND COVER will proceed as follows and its steps are summarized in the box below. In each iteration of the algorithm, we take a random sample R of U, of an appropriate expected size,

[1] In the following, for a positive number x, $\lg(x)$ denotes the logarithm base 2 of x, $\ln(x)$ denotes the natural logarithm, and $\log(x)$ denotes the logarithm base 10. The latter will be used inside the $O(\cdot)$ notation instead of the other two for uniformity.

and test (in FIND L-FACET) whether L (or some scaled version of L) is contained in conv R. If this test succeeds, we output R as the cover. If this test fails, we have gained some information about the optimal cover, and we take advantage of it by increasing the probability of selecting points from U_F as described below.

The selection of R is accomplished as follows. With each element p of U we associate a weight $w(p)$, initially 1, and denote by $w(V)$ the sum of weights of elements of a set V. Each point p of R is selected independently from U with probability

$$\Pr(p) = 1 - \left(1 - \frac{w_p}{w(U)}\right)^s \leq s\frac{w_p}{w(U)}.$$

The expected size of R is thus $\sum_{p \in U} \Pr(p) \leq \sum_{p \in U} s\frac{w_p}{w(U)} = s$. We set s to be $4cd \ln c$, which ensures that the next two lemmas hold.

Once R is chosen, and FIND L-FACET obtains an L-facet F of conv R, we double the weights of the points of U that see this facet (*i.e.* the set U_F) provided this set is not too large, i.e. provided $w(U_F) \leq w(U)/(\alpha c)$. Here, α is a constant to be specified later. An iteration when this occurs will be called *successful.* This condition is required for Lemma 3.2 to guarantee that on average at least half of the iterations are successful, and hence guarantee that the loop terminates. It is here that we take advantage of the information gained about the set U_F.

We now consider the running time of FIND COVER. We first 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. It will be shown that the number of successful iterations of the loop sufficient to find a cover of size $4cd \ln c$, if it exists, is $O(c \log(n/c))$. This number is chosen in Lemma 3.3 to guarantee success if a cover of size c exists. If a cover is not found in the number of successful iterations specified, FIND COVER fails; otherwise, it returns a cover of expected size $4cd \ln c$.

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

    s = c4d ln c, {the expected size of cover}
    for all p ∈ U, let w_p = 1
    repeat for I successful iterations:
        Choose R ⊂ U at random. (see below)
        FIND L-FACET F of conv R.
        if no L-facet found, return R.
        Let U_F = points of U that see F
        if w(U_F) ≤ w(U)/(αc) then
            for all p ∈ U_F, let w_p = 2w_p
                {reweight}
            else {not a successful iteration}
end{FIND COVER}.
```

The correctness and time complexity of the general algorithm follow from the following lemmas, which are slightly modified versions of lemmas found in [Cla93]. We state them here for completeness and to bound some constants explicitly.

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 $w(U_F) \leq w(U)/(\alpha c)$), is at least 1/2.*

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

This fact does not use the geometry of the situation at all, and one wonders whether the size of the random sample could be reduced in our application where the points of U are in (non-strictly) convex position. The deterministic algorithm in [BG94] which uses ε-nets reduces the approximation factor to $O(1)$ for the three dimensional version of a closely related problem, see Section 1.3.

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 from above by $\frac{2\alpha}{2\alpha-3} c \lg(n/c)$, which is $4c\lg(n/c)$ for $\alpha = 2$, $1501c\lg(n/c)$ for $\alpha = 1.501$.*

Proof. At each iteration where weights are doubled, $w(U)$ increases by a factor of $1 + \frac{1}{\alpha c} < e^{1/\alpha c} < 2^{\frac{3}{2\alpha c}}$. After I successful iterations, $w(U) < n2^{\frac{3I}{2\alpha c}}$. But by Lemma 3.1, U_F contains an element of C whose weight is doubled. Letting t_p be the number of times the weight of p double, after I successful iterations, $w(C) \geq \sum_{p \in C} 2^{t_p}$, with $\sum_{p \in C} t_p = I$. But by convexity of the exponential function $w(C) \geq c2^{I/c}$. But $w(C) \leq w(U)$, hence $c2^{I/c} < n2^{\frac{3I}{2\alpha c}}$. Letting $\alpha = 3/2 + \epsilon$, we get $I \leq \frac{2\alpha}{2\alpha-3} c\lg(n/c) = (1 + \frac{3}{2\epsilon}) c\lg(n/c)$. Substituting appropriate values for ϵ yields the stated bounds. □

This lemma together with Lemma 3.2 implies that the expected number of iterations of the loop in FIND COVER is $O(c\log(n/c))$. It is interesting to note the tradeoff in the constants between the constant in the running time, and the constant α, 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.

We can already mention here the component of the final running time incurred by finding and reweighting the points of U_F in each of the iteration of FIND COVER. Finding the points among the n points of U that are on one side of a hyper-plane can be done in time $O(dn)$ with a trivial algorithm. This is repeated $O(c\log(n/c))$ times over the entire algorithm. For variable d, this incurs a total cost of $O(ncd\log(n/c))$. However, for fixed d, using sophisticated data structures for half-plane range queries [Mat92], the time complexity can be significantly improved by trading off preprocessing time for query time (see [Cla93]). The resulting time complexity is

$$O\left(n^{1+\delta} + (nc)^{1/(1+\gamma/(1+\delta))}\right),$$

where δ is an arbitrarily small positive number and $\gamma = 1/\lfloor d/2 \rfloor$.

Finally, let us describe the algorithm OPTIMAL COVER which will solve the corresponding cover optimization, or **MinCover**, problems. Let c^* be the size of the smallest cover $C \subset U$ for L. To find c^* for fixed, un-scaled L, we use OPTIMAL COVER. The outer loop of OPTIMAL COVER finds an approximation of c^* up to a factor of $5/4$, and is described below.

```
OPTIMAL COVER
Input: Set U of potential cover vertices;
       object to be covered.
Output: Smallest cover, up to a factor of (5 + o(1))d ln c.

1.   for i = lg d, lg d + 1, ...
2.        FIND COVER with c = (5/4)^i
3.        if FIND COVER succeeded, stop.
end{OPTIMAL COVER}.
```

Note that the time complexity of the last call of FIND COVER dominates asymptotically the earlier calls hence the asymptotic time complexity of OPTIMAL COVER is the same as that of FIND COVER.

4 Particular Measures

In this section, we describe several versions of FIND L-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 L-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 will be the returned L-facet. This takes time $O(|R|^{\lfloor d/2 \rfloor})$ for fixed $d \geq 4$ (or $O(|R| \log |R|)$ if $d \leq 3$) [CS89, Cha93]. For variable d, we can use the algorithm in [AF92], which finds f facets of the convex hull of $|R|$ points in dimension d in time $O(|R|df)$.

As mentioned before, Lemmas 3.1, 3.2, and 3.3 apply when we replace the polytope L by a ball $\mathcal{B}(r)$, and this guarantees correctness of the algorithm.

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:

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

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 $d2^d v^{\lfloor \frac{d}{2}\rfloor}$ (this can be easily derived from the Upper Bound Theorem [MS71]).

MinCover-P:
The geometric object L to be covered is a fixed polytope P with ℓ vertices, containing the origin. Clarkson's original algorithm solves this approximation problem. In this version, FIND L-FACET finds a facet F of conv R such that there is a vertex of P on the negative side of F, if P is not contained in conv R. For each point p of P, we shoot a ray $\overrightarrow{0p}$ and find the facet F of conv R which intersects the ray $\overrightarrow{0p}$, say at the point q. If p is on the negative side of F, FIND L-FACET returns F. If no such facet exists it is easily seen that $P \subset$ conv R and a cover has been found.

Each step above can be done in $O(LP(|R|, d))$ time which is $O(LP(4cd\log c, d))$ via linear programming using duality as described in detail in [Tei95]. The expected running time of FIND L-FACET is then $O(\ell d^3 c\log c + \ell e^{O(\sqrt{d\log d})})$ using the linear programming algorithm of [Gol95]. 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]. For any arbitrary positive δ, a batch of ℓ queries on a set of m half-spaces takes time $O\Big(m\log\log m + m\log\ell + (m\ell)^{1-\frac{1}{\lfloor d/2\rfloor+1}} + \ell\log^{1+d} m\Big)$. We then get an expected time for the entire algorithm of

$$\begin{aligned} O\Big(n^{1+\delta} &+ (nc)^{1/(1+\gamma/(1+\delta))} + c^2\log\frac{n}{c}\log c\log\ell \\ &+ c^2\log\frac{n}{c}\log c\log\log c + c\log\frac{n}{c}(\ell c\log c)^{1-\frac{1}{\lfloor d/2\rfloor+1}} \\ &+ (c\log\frac{n}{c})\ell\log^{1+d}(c\log c)\Big), \end{aligned}$$

a slight improvement over the existing bound $O(n^{1+\delta} + c\ell^{1+\delta} + c(\ell c)^{1/(1+\gamma/(1+\delta))} + (nc)^{1/(1+\gamma/(1+\delta))})$ [Cla93] for large ℓ. The analysis resulting in the above expression is similar to the one given for **MinCover-B** and uses the fact that the expected size of R is $4cd\ln c$. If $\ell = O(n)$, the above running time is more simply $O(n^{1+\delta} + c^2 n\log n\log c)$.

MaxScale-B:
Here we are given a desired cover size c, and we wish to maximize the radius r of $\mathcal{B}(r)$ subject to the condition that $\mathcal{B}(r)$ is contained in some cover of size c, taken from points of U. Let $r^*(c)$ be this maximal radius. We will use a version of FIND COVER rather than OPTIMAL COVER. Here again we return only an $O(d\log c)$ approximation to this cover. The corresponding version of FIND L-FACET

will be similar to the version FIND L-FACET described for the **MinCover-B** problem, but we always compute the facet *closest* to the origin and return this as an L-facet. We also remember the maximum among all the distances to the origin of the closest facets of each iteration, and the corresponding cover. Again we do only $4c\lg(n/c)$ successful iterations, *i.e.* expected $O(c\log(n/c))$ calls to FIND L-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 that for the **MinCover-B** approximation.

MaxScale-P:
The solution to this problem is very similar to the previous case with the ball replaced by a polytope P. We use FIND COVER to solve this problem, and modify FIND L-FACET so that it determines the largest possible scaling factor λ such that $\lambda P \subseteq \text{conv } R$. This is done by ℓ linear programming queries as in **MinCover-P**, but instead we compute $\lambda(R) = \max\{\lambda > 0 : \lambda v \in \text{conv } R\}$ for each vertex v of P. Taking the minimum of these over all ℓ vertices of P gives a value $\lambda(R)$ for this cover. This value along with the corresponding facet is returned and the final value corresponds to the largest scaling P can undergo while still remaining inside conv R.

After the required number of iterations in FIND COVER, we stop and return the maximum of all scaling factors, $\lambda^*(c)$. The running time for this version is the same as that for **MinCover-P**, and the same correctness argument also applies here. 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 **MaxScale-P** problem, along equally distributed rays on the set of directions (or points on a sphere.) We can use the upper bound construction of [KMY92] to find such a set of points or rays. This construction gives, say ℓ, appropriately placed points on a unit sphere whose residual radius is $1 - \frac{1}{17}(\frac{2d^2}{\ell})^{\frac{2}{d-1}}$. Thus to obtain an approximation factor of $1-\epsilon$ for the optimal radius (for a cover of size c), we need to make $\ell = \frac{2d^2}{(17\epsilon)^{\frac{d-1}{2}}}$ queries. Plugging this into the algorithms for polytope covering, we get the following theorem.

Theorem 4.2 *For fixed d, the optimal radius (for a cover of size c) for both* **MaxScale-B** *and* **MinCover-B** *can be $(1-\epsilon)$-approximated in expected time*

$$O(n^{1+\delta} + (nc)^{1/(1+\gamma/(1+\delta))} + \left[(\frac{1}{\epsilon})^{\frac{d-1}{2}}\right]c^2 \log\frac{n}{c}\log c)$$

using a simplified version of expression for the running time.

5 Allowing finger placement anywhere on the object

In this section, we give an approximation scheme that, when combined with the above algorithms, guarantees that the quality of the grasps returned by those algorithms are only a fixed ϵ from the optimum, where we now allow finger contact selection *over the entire surface of the object.* We call these ϵ-optimal grasps. The number of fingers required still satisfies the criteria given above. We start with two geometric lemmas. We first show that if the vertices of a convex set not containing the origin are perturbed by small amounts, the distance from the origin to this set is also altered by a small amount.

Lemma 5.1 *Let $A = \{p_1, p_2, \ldots, p_n\}$ be a set in $\mathbb{R}^d$, such that* conv A *is at a distance of at least d_0 from the origin, i.e. $\|p\| \geq d_0$, for all $p \in \text{conv } A$. Let $v_1, v_2, \ldots, v_n$ be vectors with with $\|v_i\| < \epsilon$, for all i, $1 \leq i \leq n$, with $\epsilon < d_0$ and let A_ϵ be set of points p_i perturbed by v_i respectively:*

$$A_\epsilon = \{p_1 + v_1, p_2 + v_2, \ldots, p_n + v_n\}.$$

Then $\|p\| \geq d_0 - \epsilon$ for all $p \in \text{conv } A_\epsilon$.

Proof. Let p be an arbitrary point of conv A_ϵ. Then by convexity

$$\begin{aligned} p &= \sum_{i=1}^{n} \lambda_i (p_i + v_i) \\ &\qquad\qquad \text{with } \textstyle\sum_i \lambda_i = 1,\ 0 \leq \lambda_i \leq 1 \\ &= \sum_{i=1}^{n} \lambda_i p_i + \sum_{i=1}^{n} \lambda_i v_i = q + v \end{aligned}$$

where $q \in \text{conv } A$ and $v \in \text{conv } \{v_1, v_2, \ldots, v_n\} \subset \mathcal{B}(\epsilon)$. Finally, using the triangle inequality, $\|q\| = \|p - v\| \leq \|p\| + \epsilon$ which implies $\|p\| \geq d_0 - \epsilon$. □

Note that in the special case when conv A contains the origin, we have $d_0 = 0$; the nontrivial cases involve instances with $d_0 > 0$ ($o \notin \text{conv } A$).

The second lemma uses the previous result to show a similar property of the boundary of a convex set containing the origin.

Lemma 5.2 *Let $A = \{p_1, p_2, \ldots, p_n\}$ be a set in $\mathbb{R}^d$ with residual radius $r(A) = d_0$. Let ϵ be a positive number with $\epsilon < d_0$, and $v_1, v_2, \ldots, v_n$ be vectors as in Lemma 5.1, and let A_ϵ be set of points p_i perturbed by v_i respectively:*

$$A_\epsilon = \{p_1 + v_1, p_2 + v_2, \ldots, p_n + v_n\}.$$

Then $r(A_\epsilon) \geq d_0 - \epsilon$.

Proof. Let $F = \text{conv } \{p_i : i \in I_F\}$ be a facet of conv A, where $I_F \subset \{1, 2, \ldots, n\}$ is the set of indices of the vertices of F. Define $\text{dist}(0, F) = \min\{\|p\| : p \in F\}$. Then $d = \text{dist}(0, F) \geq d_0$. Let $F_\epsilon = \text{conv } \{p_i + v_i : i \in I_F\}$. Then by Lemma 5.1, $\text{dist}(0, F_\epsilon) \geq d - \epsilon \geq d_0 - \epsilon$. Now note that conv A_ϵ = conv $\{F_\epsilon : F \text{ facet of } A\}$, in other words, for all facets F of conv A, $F_\epsilon \subset$ conv A_ϵ. Some of the perturbed facets F_ϵ where F is a facet of A may not lie on the boundary of conv A_ϵ, but any ray from the origin must intersect some F_ϵ at a point closer to the origin, (or at the same point) than it intersects the boundary of conv A_ϵ. This follows from the fact that $\mathcal{B}(d_0 - \epsilon)$ does not intersect any perturbed facets, and that facets are still adjacent after perturbation. Therefore any new facet of conv A_ϵ that is not the perturbed version of a facet of conv A also cannot be closer than $d_0 - \epsilon$ from the origin. We conclude that $r(A_\epsilon) = r(\{F : F \text{ facet of } A_\epsilon\}) \geq \min\{\text{dist}(0, F_\epsilon) : F \text{ facet of } A\} \geq d_0 - \epsilon$. □

From Lemma 5.2 follows

Corollary 5.3 *Let $G^* \subset \partial B$, be the optimal c-finger grasp of a body B. Let ϵ be a positive number with $\epsilon < r(\mathbf{\Gamma}(G^*))$. Let us approximate $\mathbf{\Gamma}(\partial B)$ by a set of points $W \subset \mathbf{\Gamma}(\partial B)$ such that any point of $\mathbf{\Gamma}(\partial B)$ is no farther than ϵ from some point of W. Then there exists a c- (or fewer) finger grasp $G \subset B$ of B ($\mathbf{\Gamma}(G) \subset W$) whose residual radius $r(\mathbf{\Gamma}(G)) \geq r(\mathbf{\Gamma}(G^*)) - \epsilon$.*

Proof. Let $\{q_1, q_2, \ldots, q_n\} \subset \mathbf{\Gamma}(\partial B)$ be the set of vertices of conv $\mathbf{\Gamma}(G^*)$ and $Q = \{q'_1, q'_2, \ldots, q'_n\}$ be another set of points such that $q'_i \in W$ and $\|q_i - q'_i\| \leq \epsilon$. Then letting $v_i = q'_i - q_i$ and applying Lemma 5.2, we conclude that $r(Q) \geq r(\mathbf{\Gamma}(G^*)) - \epsilon$. Hence there exists $G \subset \partial B$ such that $\mathbf{\Gamma}(G) = Q$ and $0 \in$ conv $\mathbf{\Gamma}(G)$. □

The optimal c-finger grasp where finger contact points are taken from S is no worse than the grasp obtained in Corollary 5.3. In other words, we can bound the error in the grasp quality caused by discretizing conv $\mathbf{\Gamma}(\partial B)$ by points of S. A similar argument applies to $\mathbf{\Gamma}^*$ in $\mathbb{R}^{6k}$.

It remains only to show how to construct such a set of points. It is easy to show that the image of a polygonal face of B is mapped by $\mathbf{\Gamma}$ to a polygon in $\mathbb{R}^6$, which is simply a rigid transformation of the original polygon (in its spanning plane) due to the fact that the forces are normal to the polygons. Thus the 2d-area and perimeter are not altered by this map. Furthermore distances on the polygon are also not altered by $\mathbf{\Gamma}$ because the first three coordinates of the difference of any two points on a polygon in force-torque space are zero since the corresponding forces (or normals) are equal.

We can define a uniform grid on each facet F of B, (hence also the corresponding polygon $\mathbf{\Gamma}(F)$ in $\mathbf{\Gamma}(\partial B)$). The grid has side length $\epsilon/\sqrt{2}$. Furthermore, we place a point along the perimeter of $\mathbf{\Gamma}(F)$ at each $\epsilon/2$ interval. The fact that this construction provides the required set of points, can be seen as follows. Every point in $\mathbf{\Gamma}(F)$ is either within $\epsilon/2$ from a grid point, or within $\epsilon/2$ of the boundary of $\mathbf{\Gamma}(F)$, hence within ϵ of a point on the perimeter of $\mathbf{\Gamma}(F)$. The number of points needed to approximate an arbitrary polygon P is therefore bounded by $2\,\textsc{area}(P)/\epsilon^2 + 2\,\textsc{perimeter}(P)/\epsilon$ and the number of points needed to approximate conv $\mathbf{\Gamma}(G)$ is bounded by

$$2\,\textsc{area}(\partial B)/\epsilon^2 + 2 \sum_{\text{facet } F \text{ of } B} \textsc{perimeter}(F)/\epsilon.$$

This scheme can be used with any algorithm which finds optimal grasps. Combined with the randomized algorithm of section 3 this approximation scheme yields an algorithm for finding good grasps, according to the criteria on the number of fingers set forth in section 1.2.

For example, for a sufficiently well behaved three dimensional object with area A, since the dimension d of the force-torque space is 6, we can find an ϵ-optimal grasp with $4d \log c^* = 24 \log c^*$ fingers (where c^* is the optimal) in time

$$O\Big(c(\frac{A}{\epsilon^2})^{1+\delta} + c\log(\frac{A}{\epsilon^2 c})(c\log c)^{\lfloor d/2 \rfloor}\Big)$$

(using a simplified expression) using **MinCover-B**. In the same time bound, the quality (residual radius) of the c^*-finger optimal grasp can be found exactly. It can also be found approximately (up to an error of ϵ – using the same ϵ as above) in time

$$O\Big(c(\frac{A}{\epsilon^2})^{1+\delta} + \frac{1}{4}(\frac{1}{\epsilon})^{5/2} c^2 \log\frac{A}{\epsilon^2 c}\log c\Big)$$

by further simplifying the expression of Theorem 4.2.

6 Acknowledgments and Concluding Remarks

Grasping and dextrous manipulation occupy an important position in robotics and automation and the underlying theory has begun to play an even important role in manufacturing as medium scale robot controlled batch processing has to rely on robust clamping, fixturing and assembly technology. While the existing algorithmic work deals with the question of graspability (i.e., does there exist some grasp?) quite well, the question of how to obtain robust grasp has resisted efficient algorithmic solution. In spite of pessimistic algorithmic complexity results, it was hoped that it should be possible to get "good" suboptimal results efficiently. Even in this direction, the best existing result was only of theoretical interest as the number of fingers for which those algorithms worked were astronomically large. This paper obtains significant improvement over the earlier results by providing algorithmic techniques that provide "good" grasps and also for reasonably few fingers. The underlying approach is general enough to work for several different variations of the measure of goodness of grasps and also for related notion of fixturing and assembly. We believe that these algorithms can be used in practice in the traditional industrial environment reliably. As a related byproduct, we also present some nice generalization of Clarkson's randomized approach in solving the geometric covering problem and exhibit its close relation to grasping problems in the general setting. We expect that this connection can (will) be further exploited and will provide better geometric as well as robotics algorithms.

In the immediate future, we plan to explore some of the problems that remain unresolved. For example, a bottleneck in the ball version of the approximation problems is finding the facet of conv R closest to the origin. It would be interesting to find a faster method without approximating. 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. It would be interesting to find a more natural definition. 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 Ken Clarkson, Ricky Pollack, and Chee Yap for helpful discussions on this subject, and the anonymous referees for their generous suggestions which helped improve the exposition of this paper.

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 Comput. Sci.*, pages 246–252. Springer-Verlag, 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.

[CS89] K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.

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

[ECS97] I. Z. Emiris, J. F. Canny, and R. Seidel. Efficient perturbations for handling geometric degeneracies. *Algorithmica*, 19(1–2):219–242, September 1997.

[EM90] H. Edelsbrunner and E. P. Mücke. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graph.*, 9:66–104, 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.

[GLS88] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin Heidelberg, 1988. Second edition 1994.

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

[Grü67] B. Grünbaum. *Convex Polytopes.* Wiley, New York, NY, 1967.

[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.

[Mat92] J. Matoušek. Reporting points in halfspaces. *Comput. Geom. Theory Appl.*, 2(3):169–186, 1992.

[Mis91] B. Mishra. Workholding – analysis, design and planning. In *IROS '91*, pages 53–57, Osaka, Japan, November 1991.

[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.

[MLS94] R. Murray, Z. Li, and S. Sastry. *A Mathematical Introduction to Robotic Manipulation.* CRC Press, 1994.

[MNP90] X. Markenscoff, L. Ni, and C. 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(4):541–558, 1987.

[Ngu87] V.-D. Nguyen. Constructing force-closure grasps in 3D. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 240–245, 1987.

[PSS+95] J. Ponce, S. Sullivan, A. Sudsang, J.-D. Boissonnat, and J.-P. Merlet. Algorithms for computing force-closure grasps of polyhedral objects. In K. Goldberg, D. Halperin, J.C. Latombe, and R. Wilson, editors, *Algorithmic Foundations of Robotics*, pages 167–184. A.K. Peters, 1995.

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

[Yap90] C. K. Yap. A geometric consistency theorem for a symbolic perturbation scheme. *J. Comput. Syst. Sci.*, 40:2–18, 1990.

[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.