Recognizing Polygonal Parts from Width Measurements

Esther M. Arkin* Martin Held† Joseph S. B. Mitchell‡ Steven S. Skiena §

State University of New York, Stony Brook, NY 11794

Abstract

Automatic recognition of parts is an important problem in many industrial applications. One model of the problem is: Given a finite set of polygonal parts, use a set of "width" measurements taken by a parallel-jaw gripper to determine which part is present. We study the problem of computing efficient strategies ("grasp plans"), with the goal to minimize the number of measurements necessary in the worst case. We show that finding a minimum length grasp plan is NP-hard, and give a polynomial time approximation algorithm that is simple and produces a solution that is within a log factor from optimal.

1 Introduction

In automated manufacturing it is often necessary to recognize parts and their orientation; see [1, 2, 3, 4, 6, 9]. In this paper we discuss a model suggested in a few recent robotics papers [3, 9, 4], in which a finite set of polygonal parts is given and one considers a parallel-jaw gripper that can grasp any polygonal part in a finite number of stable grasps. A grasp is called *stable* if at least 3 vertices of the part are in contact with the gripper jaws, and any further closing of the gripper would deform the part. See Figure 1 for an example of two unstable and one stable grasps of a triangle. We assume that a gripper positioned at some orientation of its parallel jaws can exert force causing a part to rotate until it reaches a position in which it is stably grasped. A measurement is then taken of the distance between the gripper jaws, which we call a *width* of the part. We wish to find a sequence of angles for the gripper, conditional on the measurements obtained, for efficiently recognizing a part from the given library of parts. (In [3, 9, 4], "width" is called *diameter*.)

We assume that all parts are convex polygons; this is without loss of generality, since the measurements ob-

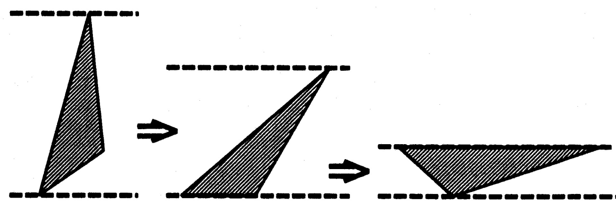

Figure 1: Unstable and stable grasps

tained by gripping an arbitrary polygonal part are identical to those obtained by gripping a part that is the convex hull of the polygon. Hence, in a stable grasp, at least one side (edge of the polygonal part) is flush against one of the jaws.

It is easy to see that, for a given set of width measurements, there is more than one polygonal part that is consistent with these measurements. In fact, Rao and Goldberg [9] show that there are an infinite (uncountable) number of polygonal parts consistent with any set of measurements, most of which have parallel sides. They further show that, given a set of width measurements, deciding whether there exists a polygon with no parallel sides, consistent with these measurements is NP-complete.

These results motivated [9, 4] to study the problem of identifying a part from a known library of u parts, $\mathcal{P} = \{P_1, P_2, \dots P_u\}$, using a minimum number of measurements. Following their definitions, a ***grasp action*** at angle α consists of rotating the jaws of the gripper to an angle of α with the x-axis, closing the gripper so that the part is in a stable grasp and measuring the width. A ***grasp plan*** is a tree of grasp actions, where each internal node corresponds to a grasp action. Alternatively, we can think of each node as a set of candidate polygonal parts from our library, where the root of the tree corresponds to the entire library and the leaves correspond to single parts. The *length* of a grasp plan is the depth of this tree. Note that if all grasp actions on all parts yield distinct width measurements, then a single grasp serves to do discrimination; thus, the need to devise efficient grasp plans arises from a type of "degeneracy" (or near degeneracy) that exists in the library of parts (and is quite common in industrial settings).

As an example, consider a library of three parts — a square of side length 1, a square of side length 2, and a

*estie@ams.sunysb.edu. Partially supported by NSF grant CCR-9204585.

†held@ams.sunysb.edu. Supported by NSF Grant DMS-9312098. On sabbatical leave from Universität Salzburg, Salzburg, Austria.

‡jsbm@ams.sunysb.edu. Partially supported by NSF grant CCR-9204585, and by grants from Hughes Aircraft and Boeing Computer Services.

§skiena@cs.sunysb.eduu. Partially supported by NSF grant CCR-9109289 and ONR award 400x116yip01.

rectangle whose side lengths are 1 and 2. Clearly any grasp action yields a width measurement of 1 or 2. An optimal grasp plan in this case will consist of a first grasp action, yielding two children of the root, corresponding to width measurements 1 and 2. Each of these nodes corresponds to two possible parts at one possible orientation. In both cases, a second grasp action at angle $\alpha = 90^o$ will suffice to determine uniquely the part and its orientation. See Figure 2.

Note, however, that there are sets of shapes (e.g., a square of side length 1 and an equilateral triangle of altitude 1) for which it is impossible to use width measurements to identify which shape is present. Thus, in the remainder of the paper, we assume that the given library $\mathcal{P}$ consists of parts that *are* identifiable using width measurements, or at least that our recognition problem is limited to determining the equivalence class of a part.

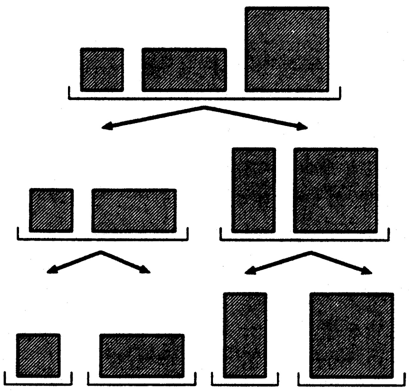

Figure 2: Grasp tree example

Let n be the total number of edges of the (convex) polygonal parts in the library $\mathcal{P}$. Govindan and Rao [4] and Rao and Goldberg [9] give two algorithms: one constructs an optimal plan in time $O(n^4 2^n)$, and the other constructs a suboptimal plan in time $O(n^2 \log n)$. Govindan and Rao [4] conjecture that the problem of finding an optimal plan is NP-hard, and leave open the problem of finding a suboptimal grasp plan with a good performance guarantee.

In this paper, we resolve both open problems. In Section 2, we prove that finding an optimal grasp plan is NP-hard; in Section 3, we give a simple polynomial-time algorithm to obtain a provably good grasp plan, whose length is within a logarithmic factor of optimal.

2 Proof of Hardness

In this section we show that the problem of finding an optimal grasp plan is NP-hard. The proof is modeled after the one given in Arkin et al. [2]. We use a reduction from the ABSTRACT DECISION TREE PROBLEM defined and shown to be NP-complete by Hyafil and Rivest [5]: Let $\mathcal{U} = \{1, 2, \dots u\}$ be a universal set, and $\mathcal{T} = \{T_1, T_2, \dots, T_m\}$ a set of tests. For each test j and each element i, we either have $T_j(i) =$ "true" or "false". We also let T_j denote the set of elements for which the test T_j is true. The problem is to construct an identification procedure for the elements in $\mathcal{U}$ such that the number of tests used is minimum. An identification procedure can be thought of as a binary decision tree, and the problem is to minimize its height.

Theorem 1 *The problem of finding an optimal grasp plan for a set of (convex) polygonal parts is NP-hard.*

Proof. We show that for any instance of the ABSTRACT DECISION TREE PROBLEM, there is an equivalent instance of the grasp planning problem. Given an instance of the ABSTRACT DECISION TREE PROBLEM, we build a library of polygonal parts, one for each element, as follows. Let M denote a regular $2(m+1)$-gon with sides of unit length, and consider the sides of M to be indexed $j = 1, \dots, 2(m+1)$, where edge j is between vertices v_{j-1} and v_j. (For ease of notation, we let $v_0 = v_{2(m+1)}$.) For each edge $j = 1, \dots, m+1$ of M, we construct either a "small" or a "smaller" triangle with base edge j. For this purpose, let y_j be the point on edge j at distance 1/2 from v_{j-1} (and therefore distance 1/2 from v_j). For each edge $j = 1, \dots, m+1$ let $x_j^1, x_j^2 \notin M$ be points "just outside" edge j, at distances ϵ^1/j and ϵ^2/j from y_j, with y_j the perpendicular projection of x_j^1, x_j^2 onto edge j. Let Δ_j and δ_j be the triangles determined by edge j and points x_j^1 and x_j^2, respectively. Choose ϵ^1 small enough so that $M \cup (\cup_j \Delta_j)$ is convex, and $\epsilon^2 < \epsilon^1/(m+1)$. (By this choice, $M \cup (\cup_j \delta_j)$ is also convex.) We think of Δ_j as the "small" triangle, and of δ_j as the "smaller" triangle.

Let P_i be the convex polygon that is the union of M and of small triangles Δ_j for each test $j = 1, \dots, m$ that is "true" for element i, and of smaller triangles δ_j for each test $j = 1, \dots, m$ that is "false" for element i. In other words, each edge $j = 1, \dots, m$ is "bumped out" by ϵ^1/j if j is true for element i and by ϵ^2/j otherwise. Finally, for edge $m+1$ include the small triangle Δ_{m+1} for all polygons P_i. Formally,

$$P_i = M \cup \left[\bigcup_{j\ :\ i \in T_j} \Delta_j \right] \cup \left[\bigcup_{j\ :\ i \notin T_j} \delta_j \right] \cup \Delta_{m+1}.$$

See Figure 3 for an example in which $m = 2$, so M is a hexagon. Assume $T_1(1) =$ "false" and $T_2(1) =$ "true". We show the part corresponding to element 1, in which edge 1 is bumped out by the smaller triangle δ_1, and edges 2 and 3 are bumped out by small triangles Δ_2 and Δ_3 respectively.

Note that each of the polygonal parts is obtained from M by replacing $m+1$ of its edges by $2(m+1)$, (the $m+1$ triangles), and thus each of the polygons in our

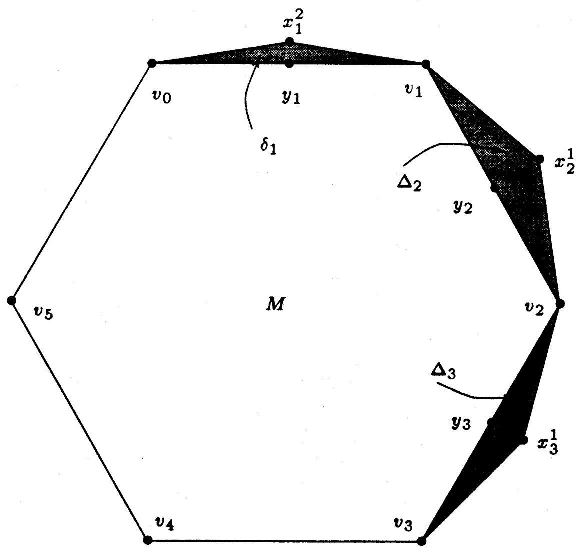

Figure 3: An example of the hardness construction

library has exactly $3(m+1)$ sides. Furthermore, no two edges are parallel. Consider possible grasps having one jaw resting on one of the two possible edges of triangle Δ_j ((v_{j-1}, x_j^1) or (x_j^1, v_j)): For small enough choices of ϵ^1, these grasps are not stable, as the projection of the vertex "across" from the edge (v_{m+j} or v_{m+j+1}) onto the line containing the edge of the triangle does not lie on the triangle edge. The same is clearly also true for grasps in which one jaw rests on an edge of triangle δ_j. Hence, each polygonal part will have exactly $(m+1)$ stable grasps, each yielding a different width measurement. Since M is symmetric, a grasp action on M at any angle yields the same measurement, call it w. For any polygonal part P_i there are two types of stable grasps:

- One jaw rests on an edge k of M, where $m+2 \leq k \leq 2(m+1)$, and the other on a point x_j^1 for some $1 \leq j \leq m+1$. This grasp has width measurement $C_j \stackrel{\text{def}}{=} w + \epsilon^1/j$.
- One jaw rests on an edge k of M, where $m+2 \leq k \leq 2(m+1)-1$, and the other on a point x_j^2 for some $1 \leq j \leq m$. This grasp has width measurement $c_j \stackrel{\text{def}}{=} w + \epsilon^2/j$.

Note that each polygonal part P_i has $(m+1)$ different width measurements from the set of $2m+1$ different possible measurements (C_{m+1}, plus C_j, c_j for $j = 1, \ldots, m$). Furthermore, all polygonal parts have width measurement C_{m+1}.

We are now ready to show that an abstract decision tree of height K exists if and only if there is a grasp plan of length $K+1$ for the constructed polygonal parts.

Consider a tree for an optimal grasp plan. The first grasp action yields a measurement that is one of the $2m+1$ possible measurements; hence, the root of the tree has that many children. After this first measurement, which is done at an arbitrary angle α, each polygonal part is consistent with this measurement in at most one possible orientation. Measurement C_{m+1} is consistent with each polygonal part in *exactly* one orientation. It is easy to see that the length of an optimal grasp plan is given by the height of the subtree rooted at this node: while this measurement tells us the orientation of the part in question, it yields no additional information, whereas other measurements may also eliminate some possible candidate parts.

Any further measurement (beyond the first) is equivalent to performing a test T_j for $j = 1, \ldots, m$ on the element i. If the measurement obtained is C_j, a "small-j" grasp, we conclude that $T_j(i)$ is true. Otherwise, if a "smaller-j" grasp, c_j, is obtained, we conclude that test $T_j(i)$ is false. Hence we can think of a grasp action in some angle α_j as answering whether test T_j is true or false. Thus, any optimal abstract decision tree has a corresponding grasp plan tree, in which the subtree rooted at the node corresponding to the first measurement being C_{m+1} has the same height as the abstract decision tree.

To summarize, we have shown that for every ABSTRACT DECISION TREE PROBLEM there is a grasp plan problem, thus showing that the problem of finding an optimal grasp plan is NP-hard. Clearly the decision version of the problem, namely, deciding whether there exists a grasp plan of length at most K, for some constant K, is therefore NP-complete. $\square$

3 Approximation Algorithm

Since finding a minimum length grasp plan is NP-hard, it is natural to attempt to devise approximation algorithms that are guaranteed to obtain a solution close to optimal. While several algorithms exist for designing grasp plans, no previous method has proven bounds on its worst-case performance.

We have seen that each candidate grasp action partitions the set of polygonal parts $\mathcal{P}$ into two or more sets, corresponding to parts that have a width measurement, at a certain angle, which is consistent with the measurement obtained. In other words, each node of the decision tree corresponds to a set of part/orientation candidate pairs. In particular, the root of the tree represents all u parts at all angles that have stable grasps, a set of size at most n. The leaves of the tree correspond to a single part, at one or more possible orientations. Let $\phi_i(v)$ be the number of possible orientations for part i at node v of the tree. We define the *weight* of a node v in the tree to be

$$wgt(v) = \sum_{1 \leq i < j \leq u} \phi_i(v) \cdot \phi_j(v).$$

The weight of a node v can be interpreted using a notion of an *ambiguity graph*, which is a u-partite graph, on at most n nodes, one corresponding to each part/orientation pair. An edge exists between two nodes if they correspond to different polygonal parts that are consistent with all measurements obtained in grasp actions so far; i.e., corresponding to nodes of the tree in the path between node v and the root. The weight of node v in the decision tree is simply the number of edges in the corresponding ambiguity graph. Since any grasp plan must distinguish between all pairs of parts, the weight of each leaf node in the tree must be zero. The weight of the root node is at most $\binom{n}{2}$, since there can be no more than this many edges in an ambiguity graph on n nodes.

A natural "greedy" heuristic in choosing an angle for a good grasp action is to select an angle α that partitions the possible candidate parts as evenly as possible. Specifically, at each node of the decision tree, we select a grasp action that minimizes the maximum weight of its children. In this section, we prove that the greedy heuristic always constructs a tree whose height is not more than a small (logarithmic) factor times the optimal height.

In [2, 1] the problem of identifying a geometric model from a library of models using probes as tests was studied. It was shown that for a similar decision tree problem this natural greedy strategy yields a log-factor approximation. The same proof can be used for an ABSTRACT DECISION TREE PROBLEM obeying certain monotonicity requirements. Although the grasp plan problem differs from those previously considered, in that it results in a multi-way rather than a binary tree, the same proof technique applies. (See Moret [8] for a survey of various heuristics for related decision tree problems.)

Theorem 2 *For u convex polygonal parts, $\mathcal{P} = \{P_1, P_2, \ldots P_u\}$ having a total of n vertices, the greedy heuristic grasp plan can be constructed in polynomial time.*

Proof. We identify each part-orientation pair with one of the (at most) n edges that is a possible contact edge with a jaw face, after the initial grasp (which we can assume is parallel to the x-axis, without loss of generality). For an edge e, we store *e.part* (the index of the part containing the edge e) and *e.stable* (the current edge of *e.part* that is in contact with a jaw face). Initially, *e.stable* $= e$, but *e.stable* may change as we apply grasps.

All grasps will be considered relative to the current jaw face. If edge *e.stable* is currently in contact with a jaw face, then each of the other edges of *e.part* defines a candidate grasp g, in that any grasp we apply at this stage will possibly result in one of the other edges of *e.part* being in contact with a jaw face. Thus, we can consider grasps to be associated with the edges of *e.part*. This results in $O(n^2)$ possible grasps.

We will construct a grasp plan (a tree). With node v, we keep track of several pieces of information:

- *v.parent* points to the parent node (if $v =$ *root* is the root, then *v.parent* = NIL);
- *v.edges* is a list of part-orientation pairs at node v;
- *v.num-edges* is the cardinality of the set *v.edges*;
- $\phi_i(v)$ is the number of times that part i ($i \in \{1, \ldots, u\}$) appears among the part-orientation pairs at v;
- *v.num-parts* is the total number of different parts present at v;
- *v.min-max-wgt* is the $wgt(v)$ obtained by the greedy strategy of minimizing the maximum weight of the children of v [we initialize *v.min-max-wgt* $= \infty$];
- *v.wgt* temporarily holds a weight associated with v;
- *v.greedy-grasp* is the grasp g that is selected at v by the greedy strategy; and,
- *v.widths* is the set of widths associated with the children of v, when we apply grasp *v.greedy-grasp*.

We maintain a list of "active" nodes, NODES. While NODES is non-empty, we do the following:

For each v in NODES do

1. For each candidate grasp action g, do
   (a) Set *v.min-max-wgt* $= \infty$, *max-wgt* $= 0$.
   (b) If v is not a leaf (i.e., if *v.num-parts* > 1), then, for each e in *v.edges*, do
      i. Compute *wgt* = *compute-grasp-action*(v, e, g).
      ii. If *wgt* > *max-wgt*, set *max-wgt* = *wgt*.
   (c) If *max-wgt* < *v.min-max-wgt*, then set *v.min-max-wgt* = *max-wgt* and *v.greedy-grasp* = g.
2. Set *children* = NIL, and then call *compute-grasp-action*$(v, e,$ *v.greedy-grasp*$)$. This creates a list, *children*, of new nodes that are the children of v that arise from grasp action *v.greedy-grasp*.
3. Append *children* to the list NODES.

To complete the description of the algorithm, we must describe the function *compute-grasp-action*(v, e, g), which computes the effect of applying grasp action g to the part-orientation pair corresponding to edge e (in contact with the jaw face) at v. The function creates a child node (if necessary), and does the appropriate "bookkeeping" to update $wgt(v)$:

Let i = *e.part*. Compute w, the width that is obtained from the grasp action g applied to part *e.part*, when *e.stable* is in contact with the jaw face. (This can be done in $O(1)$ time by looking it up in a table of size $O(n^3)$, which can be precomputed in time $O(n^3)$.)

If w is *not* in *v.widths*, then

1. Create a child node, v' [v'*.parent* $= v$, v'*.edges* $= (e)$, v'*.num-parts* $= 1$, v'*.num-edges* $= 1$, $\phi_i(v') = 1$, $\phi_j(v') = 0$ (for $j \neq$ *v.part*)], and add v' to the list *children*.

2. Add w to the list of widths, $v.widths$.
3. Set $v'.wgt = 0$.

else

1. Let v' be the child node of v that has width w.
2. Add e to the list $v'.edges$.
3. $v'.num\text{-}edges = v'.num\text{-}edges + 1$.
4. $\phi_i(v') = \phi_i(v') + 1$.
5. $v'.wgt = v'.wgt + (v.num\text{-}edges - \phi_i(v))$.

Return $v'.wgt$.

Since there are $O(n^2)$ choices for grasp action g and $O(n)$ choices for v, and $O(n)$ choices for edge e (in $v.edges$), the above algorithm clearly runs in polynomial ($O(n^4)$) time. □

We now show that the greedy heuristic gives a grasp plan of nearly optimal length:

Theorem 3 *For any instance of the grasp plan problem on u polygonal parts $\mathcal{P} = \{P_1, P_2, \dots P_u\}$ having a total of n vertices, the greedy heuristic constructs a grasp plan of length at most $2\lg n$ times that of an optimal grasp plan.*

Proof. Let l_{opt} (resp. l_{greedy}) be the length of an optimal (resp. greedy) grasp plan. Consider a decision tree T constructed by the greedy heuristic, and the corresponding weights on the nodes of T, as defined above. Clearly, for any parent and child nodes in T, the weight of the child is smaller than the weight of the parent. Hence the weights along any path down the tree T are monotonically decreasing. Consider a longest path, π, in the decision tree T, such that the ratio of the weight of the final node on the path to the initial node on the path is strictly greater than $1/2$. Let k denote the length (number of edges) of π, and W the weight of the initial node in π.

First, we obtain an upper bound on the height of the decision tree produced by the greedy algorithm; namely, we show that $l_{\text{greedy}} \leq 2(k+1)\lg n$. To see this, note that along any path of length $k+1$ or greater, at least half of the remaining weight is removed. Since the weight of the root of the tree is at most $\binom{n}{2} \leq n^2/2$, we conclude that any path of length

$$(k+1)\lg(n^2/2) = (k+1)(2\lg n - 1) \leq 2(k+1)\lg n - 1$$

reduces the weight of the nodes to at most 1. One more grasp action suffices to reduce the weight to zero.

Next, we obtain a lower bound on the height of the optimal decision tree; namely, we show that $l_{\text{opt}} \geq k+1$. Consider the final node, v, on the path π. By our definitions, its weight is $W/2 + w$ for some $w > 0$. By the pigeonhole principle, and the fact that T was constructed with the greedy heuristic, the last grasp action along π reduces the weight by at most

$$\frac{W - (W/2 + w)}{k} = \frac{(W/2) - w}{k}.$$

No edge below v in any decision tree can reduce the weight from parent to child by more than this amount, otherwise the greedy algorithm would have selected that grasp action. Thus, any decision tree rooted at v, even the optimal tree, must have height at least

$$\left\lceil \frac{W/2 + w}{(W/2 - w)/k} \right\rceil \geq k + 1.$$

Note that since v corresponds to a subset of the original part/orientation pairs, no decision tree for the full set of part/orientation pairs can be of smaller height.

Finally, we conclude,

$$\frac{l_{\text{greedy}}}{l_{\text{opt}}} \leq \frac{2(k+1)\lg n}{k+1} = 2\lg n.$$

□

It is interesting to note that the logarithmic bound is indeed tight in some instances, as the following example shows:

Theorem 4 *There are instances of the grasp plan problem for which the greedy heuristic produces a decision tree whose height is $\Omega(\lg n)$ times optimal.*

Proof. Our construction is based on an example given by Johnson, [7], showing that the log factor approximation given by the greedy algorithm for the SET COVER PROBLEM is tight. [2] use a similar construction.

We build a set of u polygon parts, starting as in Theorem 1 with a regular m-gon, M. We define $G \stackrel{\text{def}}{=} 2(2^{K+1} - 1) + K + 3$ and set $m \stackrel{\text{def}}{=} 2(2^K G + 1)$. We show later that $K = \Omega(\lg n)$. For convenience we refer to the sides of M by their indices $0, 1, \dots, m-1$. We think of the sides $1, \dots, m/2 - 1$ as grouped into 2^K groups with G sides per group. Each part consists of the union of M and $\frac{m}{2}$ small "non-special" triangles, one triangle Δ based on side 0, and a triangle Δ_l on each side $l = 1, \dots, m/2 - 1$. There are three exceptions for each part: three "special" triangles which replace three of the non-special triangles within one particular group.

The parts are divided according to three categories: class, type, and flavor. There are 2^K classes, each containing $2(2^{K+1} - 1)$ parts, hence the total number of parts is $u = 2^K \cdot 2(2^{K+1} - 1) \approx 2^{2K+2}$. Within each class c, the parts are divided into types $t = 0, \dots, K$, such that the number of parts of class c, type t is $2 \cdot 2^t$. Half of these are of flavor 1, and the remaining half are of flavor 2. Note that the total number of parts of class c is indeed $\sum_{t=0}^{K} 2 \cdot 2^t = 2(2^{K+1} - 1)$.

We now describe the 3 special triangles of a part of class c, type t, flavor f. Note that there is one group of G sides of the m-gon for each class of parts. For every class c, we further partition the group of sides corresponding to class c into three subgroups: The first subgroup containing $2(2^{K+1} - 1)$ sides, one side for each part in class c; the second subgroup containing $K+1$ sides, one side for each part of type $t = 0, 1, \ldots, K$ in class c; and the third subgroup containing 2 sides, one for each flavor $f = 1, 2$ of class c.

The three special triangles of a part in class c will be based at sides of the c-th group of sides, one special triangle in each subgroup, as follows: Let $1 \leq l \leq 2(2^{K+1} - 1)$ be the number of a part in class c. The first special triangle $\Delta_c(l)$ is placed on the l-th side of the group (and hence in the first subgroup). The second special triangle $\Delta_{c,t}$ is placed on side $2(2^{K+1}-1)+t+1$ of the group, which is the $(t+1)$-st side of the second subgroup, for $t = 0, \ldots, K$. The third special triangle Δ_c^f is placed on side $2(2^{K+1}-1)+K+1+f = G+1-f$ of the group, which is the f-th side of the third subgroup. We choose the triangles so that triangles with different "names" ((c, t, f), or l) result in grasps of different width, and such that all parts are convex.

Note that by our construction, the first grasp uniquely identifies the orientation of each possible part. As in Theorem 1, the length of an optimal, as well as a greedy grasp plan, (and any reasonable plan) is determined by the height of the subtree rooted at the child of the root corresponding to a width measurement generated by triangle Δ, which is common to all parts. We concentrate on the subtree rooted at this node. One (possibly optimal) strategy is to measure at triangles Δ_c^f of which there are $2 \cdot 2^K$. As soon as a special triangle is identified, the class and flavor of the part are known, and the plan can be completed by measuring at all sides corresponding to parts of this class and flavor. Thus, at most $2^{K+1} - 1$ additional measurements suffice, and the length of this grasp plan is at most 2^{K+2}.

The goal of the greedy algorithm is to try to measure at special triangles that appear in as many parts that have not yet been eliminated from consideration as possible. Instead of measuring at a possible special triangle Δ_c^f which appears in $2^{K+1} - 1$ parts, the greedy heuristic would measure at the possible special triangles $\Delta_{c,K}$, for $c = 1, \ldots, 2^K$, each of which is contained in 2^{K+1} parts. If such a special triangle is not found, the greedy algorithm would next measure at possible special triangles $\Delta_{c,K-1}$, for $c = 1, \ldots, 2^K$, each of which is contained in 2^K parts, instead of possible special triangles Δ_c^f. Although for each pair (c, f) there are $2^{K+1} - 1$ parts containing it, 2^K have already been eliminated from consideration by the measurements taken so far. Thus, the special triangles Δ_c^f are present in only $2^{K+1} - 1 - 2^K = 2^K - 1$ possible parts.

This process continues for $t = K-1, K-2, \ldots, 0$. Thus, a greedy algorithm, in the worst case, will perform at least $(K+1) \cdot 2^K$ measurements. The grasp plan obtained by the greedy approach has length at least $\approx K \cdot 2^K$, whereas the alternative approach yields a tree of height at most 2^{K+2}. The ratio of the heights of these trees is $\approx K/4$.

Recall that the number of parts is $u \approx 2^{2(K+1)}$, and the number of sides of each part is $\frac{3}{2}m$, where $m = 2(2^K G + 1)$, and $G = 2(2^{K+1} - 1) + K + 3$. Thus, the total number, n, of sides of all polygonal parts is $n = O(2^{2(K+1)} \cdot 2(2^K \cdot 2 \cdot 2^{K+1})) = O(2^{4K+5})$. Finally, the ratio of the heights of the two trees is approximately $K/4 \approx (\lg n - 5)/16 = \Omega(\lg n)$. □

4 Conclusion

We have resolved open problems posed in [4, 9], by showing that it is NP-hard to compute an optimal grasp plan for identifying polygonal parts, and by providing a provably good approximation algorithm for the problem.

It would be interesting to examine which other methods of "probing" a library of parts yield similar results. Furthermore, it would be interesting to study the effect of measurement uncertainty on the recognition problem.

Acknowledgement We would like to thank Ken Goldberg for useful comments that improved the paper.

References

[1] E.M. Arkin, M. Goodrich, J.S.B. Mitchell, D. Mount, C.D. Piatko, and S.S. Skiena, "Point probe decision trees for geometric concept classes". In *Proc. 3rd Workshop Algorithms Data Struct.*, volume 709 of *Lecture Notes in Computer Science*, pages 95–106. Springer-Verlag, 1993."

[2] E.M. Arkin, H. Meijer, J.S.B. Mitchell, D. Rappaport, and S.S. Skiena, "Decision Trees for Geometric Models". *Proc. 9th Annual ACM Symposium on Computational Geometry*, San Diego, CA, May 1993, pp 369-378.

[3] J.F. Canny, and K.Y. Goldberg, "RISC for Industrial Robotics: Recent Results and Open Problems". *Proc. IEEE International Conference on Robotics and Automation*, San Diego, CA, May 8-13, 1994.

[4] R. Govindan, and A.S. Rao, "Optimal Strategies for Recognizing Polygonal Parts". *Proc. IEEE International Conference on Robotics and Automation*, San Diego, CA, May 8-13, 1994.

[5] L. Hyafil, and R.L. Rivest, "Constructing Optimal Binary Decision Trees is NP-Complete". *Information Processing Letters*, **5**, 15-17, 1976.

[6] Y-B. Jia and M. Erdmann, "Geometric Sensing of Known Planar Shapes", Technical Report CMU-RI-TR-94-24, Robotics Institute, Carnegie-Mellon University, July 1994.

[7] D.S. Johnson, "Approximation Algorithms for Combinatorial Problems". *J. Comp. Sys. Sciences*, **9**, 256-278, 1974.

[8] B. Moret, "Decision Trees and Diagrams". *Computing Surveys*, 593-623, 1982.

[9] A.S. Rao, and K.Y. Goldberg, "Shape from Diameter: Recognizing Polygonal Parts with a Parallel-Jaw Gripper". *The International Journal of Robotics Research*, **13**, No. 1, 16-37, 1994.