ELSEVIER

Computational Geometry 10 (1998) 249–262

Computational Geometry
Theory and Applications

Computing fence designs for orienting parts ☆

Robert-Paul Berretty [a,1], Ken Goldberg [b,2], Mark H. Overmars [a], A. Frank van der Stappen [a,*]

[a] *Department of Computer Science, Utrecht University, P.O. Box 80089, 3508 TB Utrecht, The Netherlands*
[b] *Industrial Engineering and Operations Research, University of California at Berkeley, Berkeley, CA 94720, USA*

Communicated by C.M. Hoffmann; submitted 15 August 1997; accepted 3 December 1997

Abstract

A common task in automated manufacturing processes is to orient parts prior to assembly. We consider sensorless orientation of a polygonal part by a sequence of fences. We show that any polygonal part can be oriented by a sequence of fences placed along a conveyor belt, thereby settling a conjecture by Wiegley et al. (1997), and present the first polynomial-time algorithm to compute the shortest such sequence. The algorithm is easy to implement and runs in time $O(n^3 \log n)$, where n is the number of vertices of the part. © 1998 Elsevier Science B.V.

Keywords: Robotics; Parts feeding; Fence design; Planning

1. Introduction

Many automated manufacturing processes require parts to be oriented prior to assembly. A part feeder takes in a stream of identical parts in arbitrary orientations and outputs them in a uniform orientation. Part feeders often use data obtained from some kind of sensing device to accomplish their task. We consider the problem of *sensorless orientation* of parts, in which the initial pose of the part is assumed to be unknown. In sensorless manipulation, parts are positioned and/or oriented using passive mechanical compliance. The input is a description of the part shape and the output is a sequence of open-loop actions that moves a part from an unknown initial pose into a unique final pose. Among the sensorless part feeders considered in literature are the parallel-jaw gripper [7,10], a single pushing jaw [2,11,12,14], a conveyor belt with a sequence of (stationary) fences placed along its sides [6,15,17], a

☆ Research is supported by NATO Collaborative Research Grant CRG 951224.

* Corresponding author.

[1] Supported by the Dutch Organization for Scientific Research (N.W.O.)

[2] Supported in part by the National Science Foundation under IRI-9612491 and by Presidential Faculty Fellow Award IRI-9553197.

PII S0925-7721(98)00010-8

conveyor belt with a single rotational fence (1JOC) [1], a tilting tray [9,13], and vibratory plates and programmable vector fields [4,5].

The pushing jaw [2,11,12,14] orients a part by an alternating sequence of pushes and jaw reorientations. The problem of sensorless orientation by a pushing jaw is to find a sequence of push directions that will move the part from an arbitrary initial orientation into a single known final orientation. Such a sequence is referred to as a *push plan*. Goldberg [10] showed that any polygonal part can be oriented by a sequence of pushes. Chen and Ierardi [7] proved that any polygonal part with n vertices can be oriented by $O(n)$ pushes. They showed that this bound is tight by constructing (pathological) n-gons that require $\Omega(n)$ pushes to be oriented. Goldberg gave an algorithm for computing the shortest push plan for a polygon. His algorithm runs in $O(n^2)$ time.

The problem of *fence design* is to determine a sequence of fence orientations (see Fig. 1) such that the fences with these orientations align the part as it moves down a conveyor belt and slides along these fences [6,15,17]. The motion of the belt effectively turns each slide into a push action by the fence in the direction normal to the fence. The fact that the direction of the push, i.e., the normal at the fence, must have a non-zero component in the direction opposite to the motion of the belt imposes a restriction on successive push directions. Fence design can be regarded as finding a constrained sequence of push directions (see Section 2.2 for the actual constraints). The additional constraints make fence design considerably more difficult than sensorless orientation by a pushing jaw. Wiegley et al. [17] conjectured that a fence design exists for any polygonal part. They gave an exponential algorithm for computing the shortest sequence of fences for a given part.

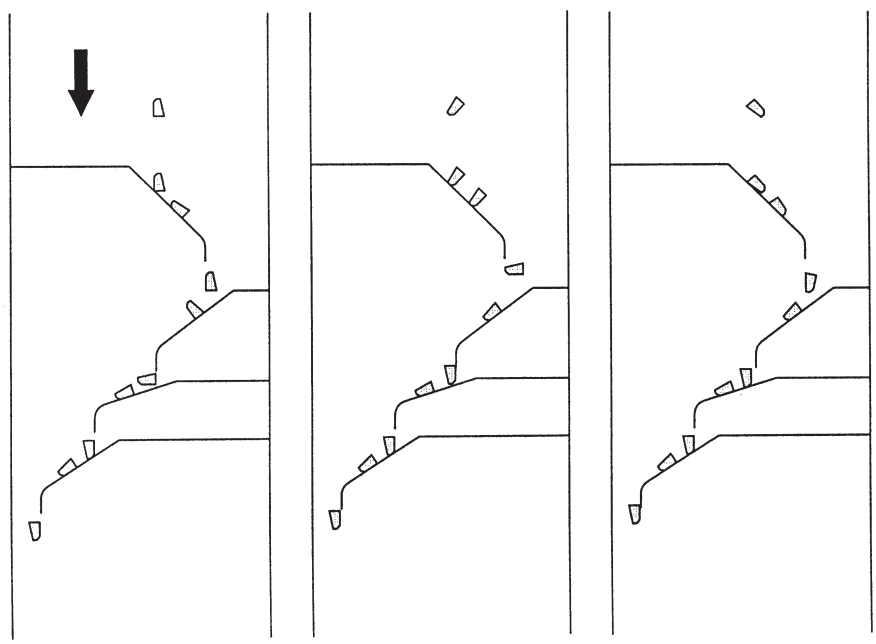


Fig. 1. Three overhead views of the same conveyor belt and fence design. The traversals for three different initial orientations of the same part are displayed. The traversals show that the part ends up in the same orientation in each of the three cases.

In this paper, we prove the conjecture that a fence design exists for any polygonal part. In addition, we give an $O(n^3 \log n)$ algorithm for computing a fence design of minimal length (in terms of the number of fences used). We show that fence designs of length $O(n)$ exist for a large class of parts. The algorithm is easy to implement and the resulting program returns fence designs for input parts within a fraction of a second. The program can be tuned to take into account certain quality requirements on the fence design, like minimum and maximum (successive) fence angles to prevent impractical steep and shallow fences and a long series of fences on a single side of the belt, which would require an impractically wide conveyor belt. The cost of the incorporation of quality measures is an increase of the algorithm's running time to $O(n^4)$.

Throughout the paper, we assume zero friction between the part and the fences. Since any push action acts on the convex hull of the part rather than on the part itself, we assume without loss of generality that the part under consideration is convex. Furthermore, we assume that the parts do not have meta-stable edges, i.e., the perpendicular projection of the center-of-mass on an edge does not intersect a vertex of the edge.

This paper is organized as follows. In Section 2, we first review the key notion of a push plan, and identify the constraints on the relative push angles for fence design. Section 3.1 discusses an $O(n^3 \log n)$ algorithm for computing the shortest fence design for a part. In Section 4, we focus on a large class of asymmetric polygonal parts and show that these parts can be oriented by fence designs of linear length (in the number of vertices). Section 5 generalizes the results of Section 4 to parts with some kind of symmetry.

2. Push plans and fence designs

2.1. The push function

In this section we focus on the push function of a part. Let P be a convex polygonal part with n vertices and center-of-mass c. We assume that a fixed coordinate frame is attached to P. Directions are expressed relative to this frame. The contact direction of a supporting line l of P is uniquely defined as the direction of the normal of l pointing into P. The radius function $r : [0, 2\pi) \to \mathbb{R}^+$ maps a direction ϕ onto the distance from c to the supporting line of P with contact direction ϕ (see Fig. 2). For a polygonal part, the radius function is a continuous piecewise sinusoidal function, and can be computed in $O(n)$ time by checking each vertex [12]. The final orientation of a part that is being pushed can be determined from its radius function.

In most cases, parts will start to rotate when pushed. If pushing in a certain direction does *not* cause the part to rotate, then the contact normal at one of its points of contact with the jaw passes through the center-of-mass [12]. We refer to the corresponding direction of the contact normal as an *equilibrium* push direction or orientation. If pushing does change the orientation, then this rotation changes the orientation of the pushing device relative to the part. We assume that pushing continues until the part stops rotating and settles in a stable equilibrium pose, which is an equilibrium with a preimage of non-zero length.

The *push function* $p : [0, 2\pi) \to [0, 2\pi)$ links every orientation ϕ to the orientation $p(\phi)$ in which the part P settles after being pushed by a jaw with initial contact direction ϕ (relative to the frame attached to P). The rotation of the part due to pushing causes the contact direction of the jaw to

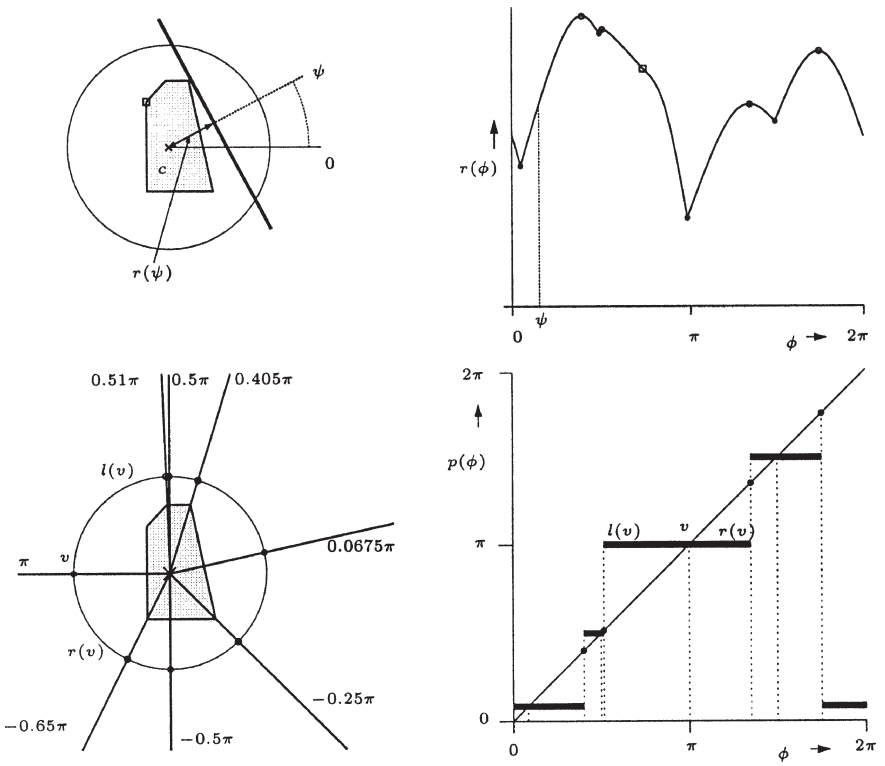


Fig. 2. At the top, a polygonal part and its radius function $r : [0, 2\pi) \to \mathbb{R}^+$. The boxed kink in the radius corresponds to the boxed vertex of the part, which is not a maximum of the radius function. Below, the same part and its push function, in a circular and regular representation.

change. The final orientation $p(\phi)$ of the part is the contact direction of the jaw after the part has settled. The equilibrium push directions are the fixed points of p.

The push function p of a polygonal part consists of steps, which are intervals $I \subset [0, 2\pi)$ for which $p(\phi) = C$ for all $\phi \in I$ and some constant $C \in I$. The steps of the push function are easily constructed from the radius function r. If the part is pushed in a direction corresponding to a point of non-horizontal tangency of the radius function, then the part will rotate in the direction in which the radius decreases. The part finally settles in an orientation corresponding to a local minimum of the radius function. As a result, all points in the open interval I bounded by two consecutive local maxima of the radius function r map onto the orientation $\phi \in I$ corresponding to the unique local minimum of r on I. (Note that ϕ itself maps onto ϕ because it is a point of horizontal tangency.) This results in the steps of the push function. In preparation for the next sections, we define two open intervals $l(v) = \{\phi < v \mid p(\phi) = v\}$ and $r(v) = \{\phi > v \mid p(\phi) = v\}$ for each fixed point – or equilibrium orientation – v of the push function. We refer to these intervals as v's left and right environment, respectively. The interval $l(v)$ corresponds to the half-step left of $v = f(v)$ and $r(v)$ corresponds to the half-step right of $v = f(v)$ (see Fig. 2). Note that an equilibrium v is stable if $l(v) \cup r(v) \neq \emptyset$. Besides the steps, there are isolated points satisfying $p(\phi) = \phi$ in the push function, corresponding to local maxima of the radius function. Fig. 2 shows a polygonal part and its push function. The

behavior of the part while being pushed is fully described by the push function. In the full version [3] we show that any step function having only non-zero-length half-steps represents a polygonal part, and is therefore a valid push function.

A push function p is said to have period d if $p(\phi) = p((\phi + d) \bmod 2\pi)$ for all $\phi \in [0, 2\pi)$. Any part can only be oriented up to (periodic) symmetry in its push function.

We use the abbreviation p_α to denote the (shifted) push function defined by

$$p_\alpha(\phi) = p\big((\phi + \alpha) \bmod 2\pi\big),$$

for all $\phi \in [0, 2\pi)$. Note that $p_\alpha(\phi)$ is the final orientation of a part in initial orientation ϕ after a reorientation by α followed by a push. We can now define a *push plan*.

Definition 1. A *push plan* is a sequence $\alpha_1, \ldots, \alpha_m$ such that $p_{\alpha_m} \circ \cdots \circ p_{\alpha_1}(\phi) = \Phi$ for all $\phi \in [0, 2\pi)$ and some fixed $\Phi \in [0, 2\pi)$.

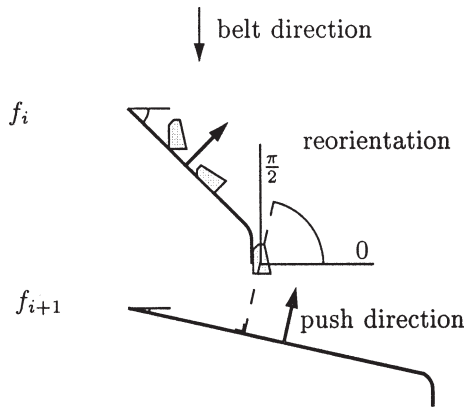
An important property of the push function is monotonicity, or the order preserving property [13]. A sequence $s_1, \ldots, s_n$ of elements of a set S is *ordered* if $s_1, \ldots, s_n$ are encountered in order when the generating cycle of S is traced *once*, starting from s_1. A function $p : S \to S$ is *monotonic* if for any ordered sequence $s_1, \ldots, s_n$ the sequence $p(s_1), \ldots, p(s_n)$ is also ordered.

2.2. Fence design

In this section we address the problem of designing a series of fences $f_1, \ldots, f_m$ that will orient P when it moves down a conveyor belt and slides along these fences $f_1, \ldots, f_m$. Let us assume that the conveyor belt moves horizontally from top to bottom, as indicated in the overhead view in Fig. 4. We distinguish between left fences, which are placed along the left belt side, and right fences, which are placed along the right side. The fence angle β_i of a fence f_i denotes the angle between the upward pointing vector opposing the motion of the belt and the normal to the fence with a positive component in upward direction. The motion of the belt turns the sliding of the part along a fence into a push by the fence. The direction of the push is – by the zero friction assumption – orthogonal to the fence with a positive component in the direction opposing the motion of the belt. Thus, the motion of the belt causes any push direction to have a positive component in the direction opposing the belt motion. We now transform this constraint on the push direction relative to the belt into a constraint on successive push directions relative to the part.

Sliding along a fence f_i causes one of P's edges e to align with the fence. The curved tip of the fence [6] guarantees that e is aligned with the belt sides as P leaves the fence. If f_i is a left fence then e faces the left belt side (see Fig. 3). If f_i is a right fence, it faces the right side. Assume f_i is a left fence. At the moment of leaving f_i, hence, after the push, the contact direction of f_i is perpendicular to the belt direction and towards the right belt side. So, the reorientation of the push is expressed relative to this direction. Fig. 3(a) shows that the reorientation α_{i+1} is in the range $(0, \pi/2)$ if we choose f_{i+1} to be a left fence. If we take a right fence f_{i+1}, then the reorientation is in the range $(\pi/2, \pi)$. A similar analysis can be done when P leaves a right fence and the edge e faces the left belt side. The result is given in the table of Fig. 3(b).

The table shows that the type of fence f_i imposes a bound on the reorientation α_{i+1}. Application of the same analysis to fences f_{i-1} and f_i and reorientation α_i leads to the following definition of a valid fence design [17].



(a)

f_i	α_{i+1}	f_{i+1}
left	$(0, \pi/2)$	left
left	$(\pi/2, \pi)$	right
right	$(-\pi, -\pi/2)$	left
right	$(-\pi/2, 0)$	right

(b)

Fig. 3. (a) For two successive left fences, the reorientation of the push direction lies in the range $(0, \pi/2)$. (b) The ranges of possible reorientations of the push direction for all pairs of fence types.

Definition 2. A *fence design* is a push plan $\alpha_1, \ldots, \alpha_m$ satisfying for all $1 \leqslant i < m$:
(i) $\alpha_i \in (0, \pi/2) \cup (-\pi, -\pi/2) \Rightarrow \alpha_{i+1} \in (0, \pi/2) \cup (\pi/2, \pi)$,
(ii) $\alpha_i \in (-\pi/2, 0) \cup (\pi/2, \pi) \Rightarrow \alpha_{i+1} \in (-\pi/2, 0) \cup (-\pi, -\pi/2)$.

The push plan on the left in Fig. 4 satisfies the constraints of Definition 2, and is therefore also a fence design.

3. Computing fence designs

3.1. A graph based approach

As every fence puts the part in a *stable* equilibrium orientation, the part is in one of these $m_s = \mathrm{O}(n)$ orientations as it travels from one fence to another. Let us label these stable equilibria $a_0, \ldots, a_{m_s-1}$. After a first fence, the part can be in any of the stable equilibria $a_0, \ldots, a_{m_s-1}$. The problem is to reduce the set of possible orientations of P to one stable equilibrium a_i by a sequence of fences. We build a directed graph on all possible *states* of the part as it travels from one fence to a next fence. A state consists of a set of possible orientations of the part plus the type (left or right) of the last fence, as the latter imposes a restriction on the reorientation of the push direction. Although there are $\mathrm{O}(2^{m_s})$ subsets of $\{a_0, \ldots, a_{m_s-1}\}$, we shall see that we can restrict ourselves to subsets consisting of sequences of adjacent stable equilibria. Any such sequence can be represented by the closed interval s defined by the first and last stable orientation a_i and a_j of the sequence. The resulting graph has $\mathrm{O}(m_s^2)$ nodes.

Consider two graph nodes (s, t) and (s', t'), where $s = [a_i, a_j]$ and s' are intervals of stable equilibria and t and t' are fence types. Let $I_{t,t'}$ be the open interval (of length $\pi/2$) of reorientations admitted by the successive fences of types t and t' according to the table in Section 2.2. There is a directed edge from (s, t) to (s', t') if there is an angle $\alpha \in I_{t,t'}$ such that a reorientation of the push direction by

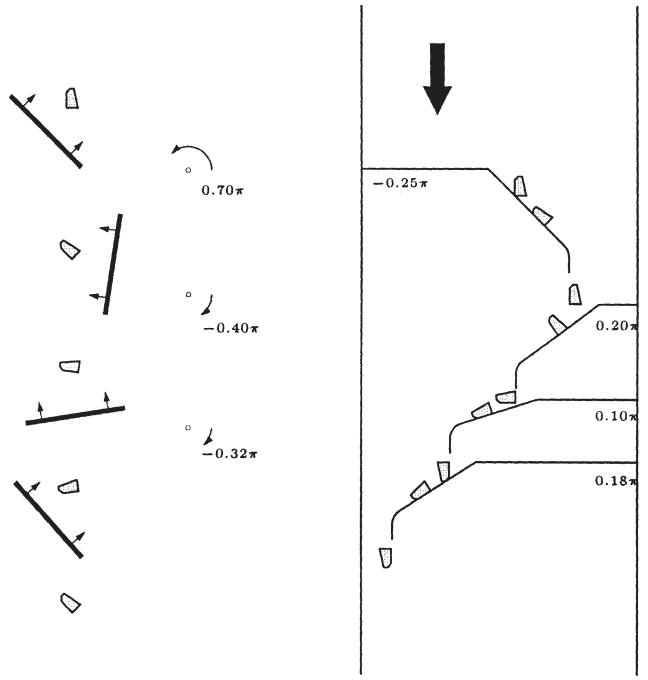


Fig. 4. The left picture shows a plan for a pushing jaw. The jaw reorientations are (from top to bottom) 0.70π, -0.40π and -0.32π. The conveyor belt on the right orients the same part. The fence angles, which are measured relative to the upward vector opposing the direction of belt motion, are -0.25π, 0.20π, 0.10π and 0.18π. (Note that the sequence of fence orientations is not the same as the sequence of orientations of the pushing jaw because the curved fence tip rotates the stable edge to become aligned with the belt direction.)

α followed by a push moves any stable equilibrium in s into a stable orientation in s'. To check this condition, we determine the preimage $(u, w) \supseteq s'$ of s' under the (monotonic) push function. Observe that if $|s| = a_j - a_i < w - u$, any reorientation in the open interval $r = (u - a_i, w - a_j)$ followed by a push will map s into s'. We add an edge from (s, t) to (s', t') if the intersection of r and the interval $I_{t,t'}$ of admitted reorientations is non-empty, and label this edge with $r \cap I_{t,t'}$. For convenience, we add a source and a sink to the graph. We connect the source to every node (s, t) in the graph for which s contains all m_s stable equilibria, and we connect every node (s, t) with $s = [a_i, a_i]$ to the sink. Every path from the source to the sink now represents a fence design; a fence design of minimum length corresponds to a shortest such path.

Theorem 3. *The shortest path in the graph corresponds to the shortest fence design, if a fence design exists.*

Proof. We prove that each path corresponds to a fence design and vice versa. The theorem follows immediately if we realize that every edge in the path corresponds to a fence in the design.

($\Rightarrow$) If there is a path, we must prove that there is a corresponding fence design. Since there is an edge from (s, t) to (s', t') if and only if the successive fences t and t' allow for a reorientation that maps s into s', this follows immediately from the construction of the graph.

($\Leftarrow$) If there is a fence design, we prove there is a path in the graph that represents this fence design. Let $f_1, \ldots, f_N$ be a fence design. We track the possible orientations of the fence design, and prove by induction that for every set of possible orientations, there is a node in the graph, and furthermore, there is a path from the source to such a node. Let A_i denote the set of all possible orientations of P leaving f_i. It is easy to see that for each A_i there are multiple nodes that include the set of possible orientations.

After the first fence f_1, all stable equilibria are possible. Since we added edges from the source to all nodes containing all stable orientations, these nodes are reachable.

We now assume that for fence f_i having type t in our fence design the nodes (s,t) with $s \supseteq A_i$ are reachable from the source. Let t' be the fence type of f_{i+1}. Let (s',t') be a node such that $s' \supseteq A_{i+1}$. Let (u,w) denote the preimage of s'. Since the push function is monotonic and by existence of the fence design which maps A_i onto A_j, there is an admitted reorientation α_{i+1} by f_{i+1} such that $(u-\alpha_{i+1}, w-\alpha_{i+1}) \supset A_i$. Therefore, let (s,t) be a node such that $(u-\alpha_{i+1}, w-\alpha_{i+1}) \supset s \supseteq A_i$. There is an edge from (s,t) to (s',t'), and there is a path from the source to (s,t). Since $s' \supseteq A_{i+1}$ is arbitrary, all (s',t') with $s' \supseteq A_{i+1}$ are reachable from the source. □

An important observation is that some graph edges are redundant if we are just interested in a fence design of minimum length. Consider a node (s,t) and all its outgoing edges to nodes $(s' = [a_i, a_j], t')$ for a fixed left endpoint a_i and a fixed fence type t'. We show that only one of these outgoing edges is sufficient. The following lemma is the key to this result.

Lemma 4. *Let (s,t), (s',t') and (s'',t') be nodes with $s' = [a_i, a_j]$ and $s'' = [a_i, a_k]$ and let $s' \supset s''$. If there are edges from (s,t) to both (s',t') and (s'',t'), then the edge from (s,t) to (s',t') can be deleted without affecting the length of the shortest path in the graph.*

Proof. Assume that τ is a path from source to sink containing the edge $((s,t),(s',t'))$ and assume $((s',t'),(s''',t'''))$ succeeds this edge in τ. Because $s' \supset s''$, there must also be an edge $((s'',t'),(s''',t'''))$ in the graph. Hence, we can replace the edges $((s,t),(s',t'))$ and $((s',t'),(s''',t'''))$ in τ by $((s,t),(s'',t'))$ and $((s'',t'),(s''',t'''))$ without affecting the length of τ. □

The repeated application of Lemma 4 to the graph (until no more edges can be deleted) leads to a reduced graph in which every node has just one outgoing edge per set of nodes with intervals with a common left endpoint and with a common fence type. The single edge from the initial graph that remains after the repeated application of Lemma 4 is the one to the node corresponding to the shortest interval. Since there are $O(m_s) = O(n)$ possible left endpoints and just two fence types, the number of outgoing edges from one node is reduced to $O(n)$. The total number of edges of the reduced graph is therefore $O(n^3)$.

Lemma 5. *The reduced graph contains $O(n^2)$ nodes and $O(n^3)$ edges.*

The (reduced) graph can be constructed in the following way. First we compute the push function and store it in such a way that preimages can be found in $O(1)$ time. For each node (s,t), left endpoint a_i, and fence type t', we must determine the shortest interval $s' = [a_i, a_j]$ such that an edge exists between (s,t) and (s',t'). We can do this by a binary search on j. Since checking whether an edge exists between a pair of nodes corresponds to computing the preimage of an interval (which can be

done in constant time), this binary search takes $O(\log n)$ time. As a similar binary search must be performed for each combination of a node (s,t), a left endpoint a_i, and a fence type t', the total time required to compute the graph edges is $O(n^3 \log n)$. A breadth-first search of the graph takes $O(n^3)$ time (see, e.g., [8]). This yields the following theorem.

Theorem 6. *The computation of the optimal fence design for a polygonal part P with n vertices takes $O(n^3 \log n)$ time.*

Let τ be a path in the graph from the source to the sink. Every edge of τ corresponds to a non-empty angular interval of possible reorientations of the push direction. We simply pick the midpoint of every such interval as the reorientation, and get a push plan which is a fence design. We can easily compute the fence angles from the reorientation angles on the path, using the properties of the fence framework [17].

3.2. Implementation

We implemented the described algorithm to test its behavior in practice. This turned out to be rather easy, using only some basic geometric computations for the push function, and some standard graph algorithms. The resulting code is very fast; it returned fence designs within a fraction of a second for all parts we tried. All fence designs shown in this paper were generated by the program. Our implementation offers the user the additional possibility of adding costs to graph edges. By doing so, the user can prevent the algorithm from outputting certain types of fence designs. Assigning high costs to edges between any pair of nodes of the same fence type t, for example, will cause the algorithm to output a sequence of alternating (left and right) fences if such a sequence exists. Alternating sequences are often preferred over sequences containing cascades of left (or right fences), as they generally allow for narrower conveyor belts (see Fig. 5). Different cost assignments can be found to prevent, e.g., unwanted steep and shallow fences. The costs make it impossible to apply Lemma 4 to reduce the graph size, as this may cause the removal of equally long or longer paths with lower cost from the graph. The size of the resulting graph is therefore $O(n^4)$. Dijkstra's algorithm (see, e.g., [8]) has been used to find the minimal cost path through the graph in time $O(n^4)$.

4. The existence of fence designs for asymmetric parts

In this section we concentrate on parts with push functions with a unique longest left environment $l(a)$ and a unique longest right environment $r(b)$. We prove that for these *asymmetric* parts a fence design always exists and has length $O(n)$. In Section 5 we deal with parts that are not asymmetric.

Chen and Ierardi [7] use a sequence of equivalent basic actions to orient a polygonal part P with a unique longest right environment $r(a)$ of length α. Each basic action consists of a reorientation of the jaw by an angle of $\alpha-\mu$, with $\mu > 0$ such that $\alpha-\mu > |r(a')|$ for any equilibrium orientation $a' \neq a$, and a subsequent push. Note that a reorientation of the jaw by $\alpha-\mu$ corresponds to a change of the orientation of the part by $\alpha-\mu$. Every basic action puts the part into an equilibrium orientation. After each basic action, the part is therefore in one of a finite number of equilibrium orientations. Let us label the m equilibrium orientations $a_0,\ldots,a_{m-1}$ in order of *decreasing* angle starting from $a_0 = a$. After the first push, the part P can be in any of the equilibrium orientations $a_0,\ldots,a_{m-1}$. Chen and

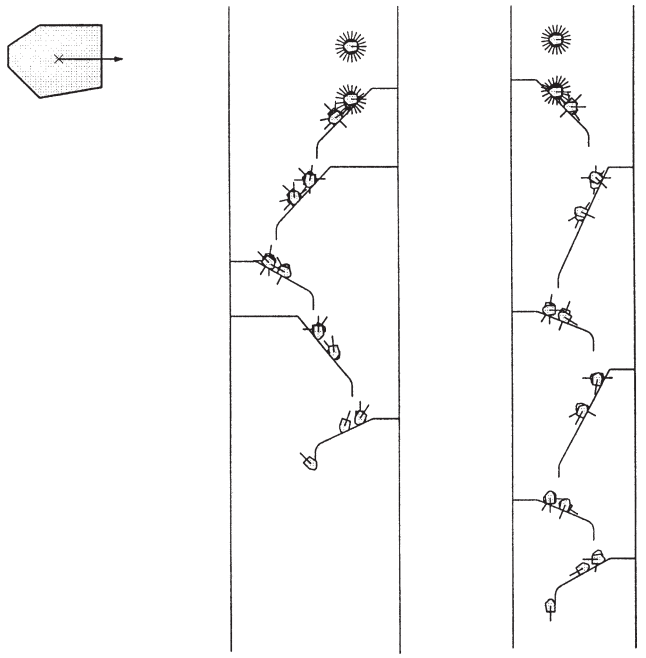


Fig. 5. An optimal design of five fences, and a design of six alternating fences allowing for a narrower belt. Every line segment emanating from the part represents a possible orientation of the part.

Ierardi show that every next basic action eliminates the last orientation in the sequence as possible orientation of the part. Assume that P is in one of the orientations $a_0, \ldots, a_k$, for some $k \geqslant 1$. The key idea behind the proof is that a next basic action will cause P, when in orientation a_0, to stay in orientation a_0 because $\alpha - \mu < |r(a_0)| = |r(a)|$, and, when in orientation a_i for some $1 \leqslant i \leqslant k$, to move into some orientation a_j with $0 \leqslant j \leqslant i-1$ because $\alpha - \mu > |r(a_i)|$. Upon completion of the basic action, the part will therefore be in one of the orientations $a_0, \ldots, a_{k-1}$. As a consequence, a total of $m+1$ basic actions suffices to put P into orientation $a_0 = a$. In other words, the sequence $(\alpha-\mu)^{m+1}$ is a valid push plan for P. In a similar manner, we could use reorientations by $-(\beta-\mu)$, and obtain the sequence $(-\beta+\mu)^{m+1}$, where β is the length of the longest left environment. The observation that $m = \mathrm{O}(n)$ leads to the result of Lemma 7.

Lemma 7 [7]. *A polygonal part P with n vertices can be oriented by $N = \mathrm{O}(n)$ pushes.*

In order for the push plan $(\alpha-\mu)^{m+1}$ or $(-\beta+\mu)^{m+1}$ to be a valid fence design, we have to show that it satisfies the constraints formulated in Definition 2. We observe first of all that there can be no more than three environments of length at least $\pi/2$, because the longest two left environments have different lengths and the longest two right environments have different lengths. As a result, there is at most one left environment of size at least $\pi/2$ or at most one right environment of size at least $\pi/2$. Assume without loss of generality that there is at most one right environment of size at least $\pi/2$. Although the length α of the longest right environment $r(a)$ can be at least $\pi/2$, the length α' of the second largest right environment $r(a')$ must be smaller than $\pi/2$. If we now choose μ such that $\alpha' < \alpha - \mu < \min\{\alpha, \pi/2\}$, then we get that $\alpha - \mu > |r(v)|$ for all equilibrium orientations $v \neq a$.

In addition, we clearly have that $\alpha - \mu < \pi/2$, which makes it easy to verify that $(\alpha - \mu)^{m+1}$ is a fence design.

Theorem 8. *An asymmetric polygonal part P with n vertices can be oriented by a fence design of length $N = \mathrm{O}(n)$.*

5. Arbitrary parts

The considerations in Section 4 show that we can orient a part by a linear length fence design if its push function has a unique longest left or right environment for which the second largest interval has a length smaller than $\pi/2$. For asymmetric parts, there always exists such an environment. If we deal with arbitrary parts, there can be several environments with the same size α, even with size greater than $\pi/2$. In this section we show that for every polygonal part there is a fence design that orients the part up to symmetry in its push function. We recall that m_s denotes the number of stable (equilibrium) orientations of P. We first show that we can orient P if the period of the push function is 2π. The plans can actually be used to orient any part up to the period in its push function. The method we use is similar to the method Chen and Ierardi introduced to generate push plans [7]. Recall that a fence design is a push plan satisfying constraints on the reorientations of the jaw. We will try to produce push plans that only use reorientations in either $(0, \pi/2)$ or $(-\pi/2, 0)$, as such plans clearly satisfy Definition 2. There are two problems with the implementation of the push plans of Chen and Ierardi as a fence design. If there are several right environments with size greater than $\pi/2$, we cannot orient the part. If there is no unique largest right environment, Chen and Ierardi apply their so called 'stretching lemma' which in general uses any reorientation of the pushing jaw. The stretching lemma shows that we can shift two possible orientations which both have a maximal left or right environment closer to each other with one single push; in other words: we can break the symmetry if there are multiple orientations with equally long maximal left or right environments. For a push plan Lemma 7 remains valid [7,16]. For fences this is not necessarily the case. We can, however, reduce the possible orientations of the part to those with maximal right environments or with right environments of length greater than $\pi/2$ by an alternating sequence of jaw applications and jaw reorientations by an angle $\min\{\alpha - \mu, \pi/2 - \mu\}$ (α and μ as in Section 4; no right environment shorter than $\pi/2$ is longer than $\pi/2 - \mu$). (We can similarly reduce the possible orientations to orientations with such left environments.) As observed, for arbitrary parts, the number of possible orientations can be larger than one. We have to use more sophisticated fence designs to further reduce the number of possible orientations.

We start with some definitions which we use throughout the rest of this section. Let α again be the length of the longest right environment and let β be the length of the longest left environment. Furthermore, let $\mathcal{A} = a_0, \ldots, a_{m_s-1}$ be the cyclic sequence of stable equilibria, cut at some arbitrary orientation and ordered by *increasing* angle. The sequence $\mathcal{A}$ is said to have right cycle d if and only if $|r(a_i)| = |r(a_{i+d})|$ for all $0 \leqslant i < m_s$. (Indexing is modulo m_s.) Similarly, the sequence $\mathcal{A}$ has left cycle d if and only if $|l(a_i)| = |l(a_{i+d})|$ for all $0 \leqslant i < m_s$. We denote by $\mathcal{R}$ the set of orientations in $\mathcal{A}$ with right environments of length α, or length greater than or equal to $\pi/2$. We define the right measure $M_{\mathcal{R}}$ of an interval $[v, w]$ of orientations by

$$M_{\mathcal{R}}\big([v, w]\big) = \big|\{a \in \mathcal{R} \mid v \leqslant a < w\}\big|.$$

In a similar manner, we can define a set $\mathcal{L}$ of orientations with left environments of length β, and a left measure $M_{\mathcal{L}}$. We let $\mu > 0$ be a small constant such that $\alpha - \mu$ and $\alpha + \mu$ are both smaller than $\pi/2$ but larger than any environment of length less than $\pi/2$. In addition, the constant μ is smaller than the length of any environment. We recall that polygonal parts without meta-stable edges have push functions without left and right environments of zero length.

Our push plans for arbitrary parts consist of three types of basic building blocks. These building blocks are referred to as MOVE, SHIFT and REDUCE.

- Suppose that $[v, w]$ with $v, w \in \mathcal{A}$ is the current interval of possible orientations. Then, the push plan $(\min(\alpha, \pi/2) - \mu)^{m_s}$ MOVEs the interval $[v, w]$ into an interval $[v', w']$ with $v', w' \in \mathcal{R}$ of equal right measure.
- Suppose that $[v = a_i, w = a_j],\ v, w \in \mathcal{A}$, is the current interval of possible orientations. Then, the push plan $(|r(v)| + \mu)$ SHIFTs the interval $[v, w]$ into an interval $[v', w']$ with $v' = a_{i+1},\ w' \in \mathcal{A}$. If $v, w \in \mathcal{R}$ then $w' = a_{j+1}$ of and the right measure of $[v', w']$ equals the right measure of $[v, w]$. Notice that in that case a SHIFT followed by a MOVE maps $[v, w]$ into an interval $[v'', w'']$ with $v'', w'' \in \mathcal{R}$ and $v' \neq v,\ w' \neq w$ (provided that $\mathcal{R}$ has at least two elements) of equal right measure.
- Suppose that $[v, w]$ with $v, w \in \mathcal{R}$ is the current interval of possible orientations. We want to define an operation which REDUCEs an interval $[v, w]$ to some interval $[v', w']$ with $v', w' \in \mathcal{A}$ of smaller measure. The REDUCE exploits specific asymmetries that are present in the push function to achieve the reduction of the measure. Different classes of push functions lead to different REDUCEs.

The objective is to use these building blocks in a push plan that is guaranteed to result in an interval of possible orientations of measure zero.

According to the "stretching lemma" of Chen and Ierardi [7], any interval $[v, w],\ v \neq w$, of possible orientations can be mapped onto a shorter interval of possible orientations $[v', w']$ by a single push. An additional push will then map $[v', w']$ onto an interval $[v'', w'']$ satisfying $M_{\mathcal{R}}([v'', w'']) < M_{\mathcal{R}}([v, w])$. The problem in applying these ideas in fence design is that the two required reorientations of the push direction may not be achievable by a sequence of fences. We will use the SHIFT and MOVE plans to overcome this problem. We classify the push functions, based on the left and right cycle of the stable equilibria, and the sizes of the environments. The implementation of the REDUCE is the main difference between the distinct classes of push functions. We distinguish the following classes of push functions.

(1) Either the left or the right cycle is m_s, and no left or right environment is longer than $\pi/2$ in that cycle.
(2) Both the left and the right cycle are smaller than m_s.
(3) Both the left and the right cycle are m_s and there is more than one environment of length greater than $\pi/2$.

We will only globally describe the approach for each of the three cases. The detailed proof is lengthy and rather involved. Interested readers can find the detailed proof in [3].

Let us first concentrate on push functions in the first and second class. Suppose that $[v, w]$ with $v, w \in \mathcal{R}$ is the current interval of possible orientations. Let v' and w' be the smallest orientation in $\mathcal{R}$ larger than v and w, respectively. If the sequence of right environments of orientations between v and v' differs from the the sequence of right environments between w and w', then there exists a simple strategy that maps $[v, w]$ either onto $[v', u]$ with $u \in \mathcal{A} \setminus \mathcal{R}$ and $w < u < w'$ or onto $[u, w']$ with $u \in \mathcal{A} \setminus \mathcal{R}$ and $v < u < v'$. Moreover, it can be shown that any interval $[v, w]$ with $v, w \in \mathcal{R}$ can be transformed – by means of SHIFT and MOVE plans – into a different interval $[v'', w'']$ with

$v'', w'' \in \mathcal{R}$ of equal measure, such that the simple strategy is guaranteed to map $[v'', w'']$ onto $[v', u]$, with $u \in \mathcal{A} \setminus \mathcal{R}$ and $w < u < w'$, and where v' and w' are the smallest orientations in $\mathcal{R}$ larger than v and w, respectively. A subsequent reorientation by $\alpha + \mu$ followed by a push will eliminate v' as a possible orientation of the part without adding new possible orientations from $\mathcal{R}$. As a consequence, the measure of the resulting interval of possible orientations is smaller than $M_{\mathcal{R}}([v', u])$. For push functions from the first class, this strategy (or its equivalent using left environments) suffices to reduce the interval of possible orientations to an interval of measure zero. The reorientations of the push direction in the entire scheme are restricted to $(0, \pi/2)$ or $(-\pi/2, 0)$, which makes the sequence of pushes a valid fence design. If the left and right cycle are both smaller than m_s, then we must eventually switch our attention from the right environments to left environments to break the rotational symmetry of the right environments (or vice versa). It turns out that such a switch can be accomplished without violating the reorientation constraints for fence designs.

The third class of push functions requires some modifications to the MOVE, SHIFT and REDUCE framework. There are, however, at most three environments of length greater than $\pi/2$, which makes it possible to treat the different cases one by one, and provide dedicated push plans which use both left and right fences. These dedicated push plans satisfy the reorientation constraints and are therefore valid fence designs.

Theorem 9 summarizes the result of this section.

Theorem 9. *Any polygonal part can be oriented up to rotational symmetry by a fence design.*

A consequence of Theorem 9 is that the algorithm for computing fence designs presented in Section 3.1 always outputs a fence design. Since the graph used in the algorithm has $O(n^2)$ nodes, the length of the shortest path is $O(n^2)$.

Theorem 10. *Any polygonal part with n vertices can be oriented up to rotational symmetry by a fence design of length $O(n^2)$. The optimal fence design can be computed in $O(n^3 \log n)$ time.*

6. Conclusion

In this paper we investigated the problem of sensorless part orientation by sequences of pushes. We showed that any polygonal part can be oriented by a sequence of fences placed along a conveyor belt. We presented the first polynomial-time algorithm for computing the shortest fence design for any given polygonal part. The algorithm is easy to implement and runs in time $O(n^3 \log n)$. The structure of the algorithm yields an $O(n^2)$ bound on the length of the shortest fence design. We showed that for asymmetric parts the length is actually bounded by $O(n)$. It remains an open problem whether an $O(n)$ bound exists for parts that are not asymmetric.

Although pathological polygons can be constructed that lead to push plans and fence designs of length $\Omega(n)$, it turns out that the length of most plans remains far below the worst-case length. In [16], van der Stappen et al. deduced a bound on the plan length that depends only on the shape of the part under consideration. It turns out that a small number of actions is sufficient for most parts. The analysis also applies to curved parts, providing the first complexity bound for non-polygonal parts. The results generalize to fence designs for parts with acyclic left and right environments. It remains an open question whether this bound can be transferred to arbitrary parts.

Acknowledgements

We thank Matt Mason and Kevin Lynch for several interesting discussions and Jeff Wiegley and Anil Rao for useful feedback.

References

[1] S. Akella, W. Huang, K.M. Lynch, M.T. Mason, Sensorless parts feeding with a one joint robot, in: J.-P. Laumond, M. Overmars (Eds.), Algorithms for Robotic Motion and Manipulation, A.K. Peters, 1996, pp. 229–238.
[2] S. Akella, M.T. Mason, Posing polygonal objects in the plane by pushing, in: IEEE International Conference on Robotics and Automation, Nice, 1992, pp. 2255–2268.
[3] R.-P. Berretty, K. Goldberg, M. Overmars, A.F. van der Stappen, Computing fence designs for orienting parts, Technical Report, UU-CS-1997-41, Department of Computer Science, Utrecht University, 1997.
[4] K.-F. Böhringer, V. Bhatt, K.Y. Goldberg, Sensorless manipulation using transverse vibrations of a plate, in: Proceedings IEEE International Conference on Robotics and Automation, Nagoya, Japan, 1995, pp. 1989–1996.
[5] K.-F. Böhringer, B.R. Donald, N.C. MacDonald, Upper and lower bounds for programmable vector fields with applications to mems and vibratory plate part feeders, in: J.-P. Laumond, M. Overmars (Eds.), Algorithms for Robotic Motion and Manipulation, A.K. Peters, 1996, pp. 255–276.
[6] M. Brokowski, M.A. Peshkin, K. Goldberg, Optimal curved fences for part alignment on a belt, ASME Trans. Mech. Design 117 (March 1995).
[7] Y.-B. Chen, D.J. Ierardi, The complexity of oblivious plans for orienting and distinguishing polygonal parts, Algorithmica 14 (1995) 367–397.
[8] T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introduction to Algorithms, The MIT Press, Cambridge, MA, 1990.
[9] M.A. Erdmann, M.T. Mason, An exploration of sensorless manipulation, IEEE J. Robotics Automation 4 (1988) 367–379.
[10] K. Goldberg, Orienting polygonal parts without sensors, Algorithmica 10 (2) (1993) 201–225.
[11] K.M. Lynch, M.T. Mason, Stable pushing: Mechanics, controllability, and planning, Internat. J. Robotics Res. 6 (15) (1996) 533–556.
[12] M. Mason, Manipulator grasping and pushing operations, Ph.D. Thesis, MIT, 1982.
[13] B.K. Natarajan, An algorithmic approach to the automated design of parts orienters, in: IEEE Annual Symposium on Foundations of Computer Science, 1986, pp. 132–142.
[14] M.A. Peshkin, A.C. Sanderson, The motion of a pushed sliding workpiece, IEEE J. Robotics Automation 4 (6) (1988) 569–598.
[15] M.A. Peshkin, A.C. Sanderson, Planning robotic manipulation strategies for workpieces that slide, IEEE J. Robotics Automation 4 (5) (1988) 524–531.
[16] A.F. van der Stappen, K. Goldberg, M. Overmars, Geometric eccentricity and the complexity of manipulation plans, Technical Report UU-CS-1996-49, Department of Computer Science, Utrecht University, 1996. To appear in Algorithmica.
[17] J. Wiegley, K. Goldberg, M. Peshkin, M. Brokowski, A complete algorithm for designing passive fences to orient parts, Assembly Automation 17 (2) (1997) 129–136.