Discrete Comput Geom 1:155–182 (1986)

Discrete & Computational Geometry

A Polynomial Solution for the Potato-peeling Problem*

J. S. Chang and C. K. Yap

Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012

Abstract. The potato-peeling problem asks for the largest convex polygon contained inside a given simple polygon. We give an $O(n^7)$ time algorithm to this problem, answering a question of Goodman. We also give an $O(n^6)$ time algorithm if the desired polygon is maximized with respect to perimeter.

1. Introduction

In computational geometry, optimization problems are often posed in a continuous (as opposed to discrete or combinatorial) setting. One can resort to numerical methods to give approximate solutions to any degree of accuracy or solve the problem symbolically and reduce the problem to root-finding or to a decision procedure for Tarski's language for elementary geometry and algebra [2, 13]. A recent paper by Sharir and Schorr [19] shows a case where no combinatorial finiteness criterion (except indirectly, by a reduction to Tarski's language) is known: It is the problem of finding shortest paths between a pair of points among polyhedral bodies in space. The preferred method for these problems, however, is to find a combinatorial "finiteness criterion" for each problem. One example where such an approach works (extremely well, in practice) is linear programming. In this paper, we address another such problem, the *potato-peeling problem* described in the abstract. This problem was first posed (in a more general

*Work in this paper has been supported in part by NSF grants #DCR-84-01898 and #DCR-84-01633, the Office of Naval Research Grant N00014-82-K-0381, and by grants from Digital Equipment Corporation, the Sloan Foundation, the System Development Foundation, and the IBM Corporation. This paper contains the main results of the paper "A Polynomial Solution for Potato-Peeling and other Polygon Inclusion and Enclosure Problems" presented in the 25th Foundation of Computer Science Conference, 1984, Florida. The second half of that paper is submitted for publication elsewhere [1].

form than we have stated it) by Goodman [11] who obtained various mathematical properties of solutions to the problem. The only partial solution in [11] gives the criteria for a finite solution if the polygon has $n \leq 5$ sides. Quite independently, Woo [21] studied the same problem dubbing it the "convex skull" problem.[1] Not only will we show that the problem is finite in general, but we will derive a polynomial time algorithm.

Our computational model, as is typical in this subject, assumes a random-access computer with infinite precision real arithmetic. Observe that the notion of a "finiteness criterion" is a relative one, in view of the infinite precision. Furthermore, we assume that the solution of simple trigonometric equations takes $O(1)$ steps (this amounts to assuming the availability of trigonometric functions and their inverses). For example, given angles α, β and constant c, we assume that we can in $O(1)$ time find the angle θ satisfying

$$\frac{\sin(\theta + \alpha)}{\sin(\theta - \beta)} = c.$$

It turns out that our method applies to other problems that have been studied. The general framework for these problems can be posed as follows. Let $\mathscr{P}, \mathscr{Q}$ be families of polygons, and let μ be a real function on polygons with the property that for all P, Q in $\mathscr{Q}$:

$$P \subseteq Q \Rightarrow \mu(P) \leq \mu(Q).$$

Note that this property holds if μ measures the area. If the polygons in $\mathscr{Q}$ are convex, the property also holds when μ measures the perimeter. In this paper, polygons are assumed simple in the sense that they are self-avoiding, and polygonal regions (also called "polygons" when the context is clear) are simple in the sense that the boundary of each region forms a simple polygon. The class of (polygon) *inclusion* and *enclosure problems*[2] are defined as follows:

$\mathrm{Inc}(\mathscr{P}, \mathscr{Q}, \mu)$: Given $P \in \mathscr{P}$, find the μ-largest $Q \in \mathscr{Q}$ that is included in P.

$\mathrm{Enc}(\mathscr{P}, \mathscr{Q}, \mu)$: Given $P \in \mathscr{P}$, find the μ-smallest $Q \in \mathscr{Q}$ that encloses P.

These two classes of problems are "duals" in some sense but we know of no systematic way whereby an algorithm for a problem can be transformed to one for its dual. For instance, the dual of potato-peeling problem is the usual problem of computing the convex hull. We review some of the inclusion and enclosure problems that have been studied.

(1) We mainly focus on recent results within the milieu of computational geometry. However, it should be pointed out that there is a related much larger and older literature arising from the field of operations research. It should be

[1] Indeed, we are indebted to T. Woo who first brought this problem to our attention. Later, M. Sharir pointed out the work of J. Goodman.

[2] Alternatively, these might be called the *inscription* and the *circumscription problems*.

clear that our problems are closely related to the "stock-cutting problems" which are concerned with cutting a sheet of material into smaller subparts under various constraints (such as all subparts are congruent to a given shape) and are subject to some optimality criteria. See [10] and the references therein. As pointed out in [10] the enclosure problem is a key subproblem in the more general stock-cutting problems. More generally, the enclosure and inclusion problems can be viewed as polygon approximation problems. For example, for the purpose of detecting collision in robotics we typically approximate a complicated shape by a simpler enclosing body.

(2) The potato-peeling problem is the case, Inc($\mathscr{P}_{\text{all}}$, $\mathscr{P}_{\text{con}}$, area) where $\mathscr{P}_{\text{all}}$ is the family of all simple polygons and $\mathscr{P}_{\text{con}}$ is the family of all convex polygons. It turns out that we can also solve the potato-peeling problem in the case where perimeter rather than area is the measure. A variation of the potato-peeling problem does not fall under the above notion of inclusion problems: Find the largest convex subregion Q of the given P subject to the constraint that Q is obtained from P by at most k cuts. Rectilinear versions of the potato-peeling problem have been addressed in [6, 15, 22]. (Note: [6] formulates its problem slightly differently but [15] shows its connection to potato peeling.)

(3) Dobkin and Snyder [9] considered the inclusion problem Inc($\mathscr{P}_{\text{con}}$, $\mathscr{P}_3$, area) where $\mathscr{P}_3$ denotes the class of all triangles and in general $\mathscr{P}_k$ denotes the class of all convex k-gons. Their algorithm runs in linear time. This result was extended by Boyce *et al.* [3] to the problems Inc($\mathscr{P}_{\text{con}}$, $\mathscr{P}_k$, area) and Inc($\mathscr{P}_{\text{con}}$, $\mathscr{P}_k$, perimeter) for any fixed k. The running time of these algorithms is $O(kn \log^2 n)$. Note that unlike the potato-peeling problem, the finiteness of these problems is easy to show: it follows from the fact that the vertices of any maximal k-gon must be a subset of the vertices of the input polygon. The techniques of [3, 9] are not sufficient for the more general problem of Inc($\mathscr{P}_{\text{all}}$, $\mathscr{P}_k$, area) since they rely on the convexity of the input polygon.

(4) Klee and Laskowski [12] considered the enclosure problem Enc($\mathscr{P}_{\text{all}}$, $\mathscr{P}_3$, area) and derived an $O(n \log^2 n)$ solution. O'Rourke *et al.* [16] improved it to linear time. DePano [8] described how the method in [12] extends to solve Enc($\mathscr{P}_{\text{all}}$, $\mathscr{P}_k$, area) for all k in $O(n^{k-2} \log^2 n)$ time (which is exponential in k). Chang and Yap [5] improved DePano's result to $O(n^3 \log k)$. By further refinement, we obtain the bound of $O(n^2 \log n \log k)$ in [1]. We remark that Dori and Ben-Bassat [10] claimed to have a linear time solution to this problem. However, their optimality proof is faulty; indeed O'Rourke [17] has provided some counterexamples.

(5) For the problem of finding the largest rectangle containing a given polygon, Toussaint [20] improved a previous quadratic time solution to linear time. In general, let $\Theta = (\theta_1, \ldots, \theta_k)$ be any sequence of angles with each $\theta_i < \pi$ and $\sum_{i=1}^{k} \theta_i = (k-2)\pi$. Let $\mathscr{P}_\Theta$ denote the family of convex k-gons whose interior angles are given by the sequence Θ. Thus we have the problem of finding smallest polygon from $\mathscr{P}_\Theta$ enclosing a given convex polygon: Enc($\mathscr{P}_{\text{con}}$, $\mathscr{P}_\Theta$, μ). DePano and Aggarwal [7] have solved some of these problems. For the three-dimensional versions of these problems, [18] describes an $O(n^3)$ algorithm for the smallest rectangular box enclosing a polyhedron.

(6) And finally problems such as finding the smallest square containing a given polygon, are also interesting. In general, let P be any polygon and

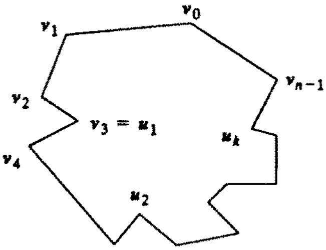

Fig. 1. Polygon $P = (v_0, \ldots, v_{n-1})$.

shape(P) be the family of polygons obtained by the transformations of scaling, rotation, translation and reflection of P. Then we have the *fixed shape problems* $\text{Inc}(\mathscr{P}_{\text{all}}, \text{shape}(P), \mu)$ and $\text{Enc}(\mathscr{P}_{\text{all}}, \text{shape}(P), \mu)$ for any convex P. The paper [7] addresses some of these problem.

2. Preliminaries

For the rest of this paper, unless otherwise stated, we assume a fixed but arbitrary polygon $P = (v_0, v_1, \ldots, v_{n-1})$ with n corners. P has $k \geq 0$ *reflex* corners: $v_{i_1}, v_{i_2}, \ldots, v_{i_k} (0 \leq i_1 < i_2 < \cdots < n)$. Write u_j for v_{i_j} (Fig. 1).

Given P, we are to find any maximum area convex subset Q contained in P; it is intuitively clear, but rigorously proved in [11], that Q is a convex polygon. For instance if P is convex than Q is unique and equal to P.

First we introduce some notations. A *chord* of P is a maximal line segment fully contained in P (note that there could be line segments in P with both end points on the boundary of P which are not chords). A chord is *extremal* if it contains two or more corners of P. In particular, an edge of P is always contained in an extremal chord. It is clear that a maximum area convex polygon must be the intersection of P and m half-planes defined by m chords of $P(m \leq k)$ as follows: Let $C_1, C_2, \ldots, C_m$ be chords of P such that each C_i passes through a distinct reflex corner of P. For any chord C of P passing through a unique reflex corner u, let C^+ denote the closed half-plane determined by C such that for a sufficiently small disc D centered at u, we have $D \cap C^+ \subseteq P$. For a chord C that passes through more than one reflex corner, the context will make it clear which half-plane is intended. Thus the convex polygon determined by the chords $C_1, C_2, \ldots, C_m$ is $P \cap (\bigcap_{i=1}^{m} C_i^+)$.

We first answer a simple case of the problem where the given polygon P has just one reflex corner u_1. From the above observation, the problem amounts to determining the chord C through u_1 that maximizes the area of $P \cap C^+$. Let Ξ be the set of extremal chords through u_1. Clearly $|\Xi| < n$. Consider the *butterfly* region B determined by a pair of adjacent chords au_1a' and bu_1b' taken from Ξ (Fig. 2).

We call u_1 the *center* of the butterfly and the line segments $[a, b]$ and $[a', b']$ the *tips* of the butterfly. The triangles Δu_1ab and $\Delta u_1a'b'$ form the two *wings* of the butterfly. A chord C *embedded in* B is determined by any point c in the tip $[a,b]$ together with c' in $[a', b']$ such that c, u_1, and c' are collinear; $C = cu_1c'$. The problem reduces to choosing for each butterfly a chord C embedded in it

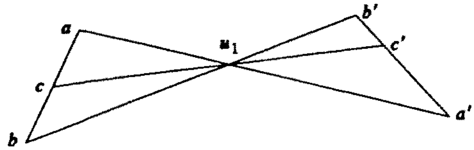

Fig. 2. Butterfly with center u_1.

such that the area $\alpha = C^+ \cap B$ is maximized. α is either the union of triangles $\triangle u_1bc$ and $\triangle u_1a'c'$ or the union of the triangles $\triangle u_1ac$ and $\triangle u_1b'c'$, depending on the orientation of C. The following lemma shows that in both cases either C is extremal or u_1 is the midpoint of the chord:

$$|u_1c| = |u_1c'|.$$

In the later case, we have a *balanced* chord.

The pair (L_0, L_1) of the lines through the two tips of B are the *supporting lines* of B. If L_0 and L_1 intersect at a point o and $o \notin C^+$ (pick any C embedded in B, say $[a, a']$), then we call the butterfly an *A-butterfly*. Otherwise it is a *V-butterfly*. See Fig. 3. (See also the next section for the general context for the A and V notations.)

Lemma 1 (Butterfly Lemma). *Given the butterfly B determined by an adjacent pair of extremal chords au_1a' and bu_1b', let $C = cu_1c'$ be a chord embedded in B maximizing α. If B is an A-butterfly, then C is either balanced or extremal. Otherwise B is a V-butterfly and C is extremal.*

Proof. Consider the case where B is an A-butterfly. First without loss of generality, assume $|au_1| \leq |a'u_1|$. If $|bu_1| \geq |b'u_1|$, then by a simple continuity argument there is a unique balanced chord C^* embedded in B. It is easy to see that any other chord C determines a smaller area than C^* (see also [4]). Otherwise $|bu_1| < |b'u_1|$ and it is not hard to see that the *extremal* chord $[b, b']$ maximizes α. In the case of a V-butterfly, if C is not extremal we can perturb C so as to enlarge α. Hence we conclude that C must be equal to one of the two extremal chords. □

This lemma clearly leads to a linear time algorithm for the potato-peeling problem if P has one reflex corner. In the next section, we look at the general case and at A- and V-butterflies in a more general setting.

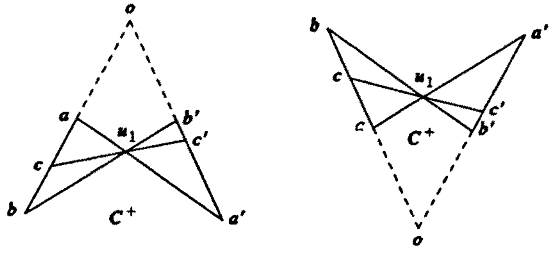

Fig. 3. A- and V-butterflies.

3. Series of Butterflies and Chains of Chords

In this section we will give a finiteness criterion for the potato-peeling problem. First let us introduce the terminology related to "sequences of butterflies" and "sequences of chords." It turns out that sequences of butterflies can be classified into two types with quite different algorithmic properties. For simplicity, we assume that no three corners are collinear. As in the previous section, for each reflex corner u of the polygon P, we can form a circular list of all extremal chords through u. A *butterfly* of P is the region determined by a pair of adjacent chords through some u having the obvious shape. Our definitions for butterflies and chords are relative to some fixed polygon P.

Definition. A *series of butterflies* $\mathscr{B} = (B_1, \ldots, B_m)$, $m \geq 1$, is any sequence of butterflies satisfying

(1) Let c_i be the center of B_i for $1 \leq i \leq m$. Then $(c_1, c_2, \ldots, c_m)$ forms a convex polygon Q contained in P. Q is degenerate if $m = 1$ or 2.

(2) The two wings of each butterfly are ordered so that the "forward" wing of B_i intersects the "backward" wing of B_{i+1} for $i = 1, \ldots, m-1$.

Let C_0 (resp. C_1) be the tip of the backward (resp. forward) wing of B_1 (resp. B_m). Then (C_0, C_1) is called the (*pair of*) *supporting tips* of the series. If L_i is the line through C_i, then (L_0, L_1) is the pair of *supporting lines*. Note that it is possible for L_0 and L_1 to be parallel or even be equal. Let $m > 1$. If the pair of supporting lines are coincident or parallel, or if they intersect at a point on the side of the line $\overline{c_1 c_m}$ opposite to Q, we will say (L_0, L_1) and (C_0, C_1) are *V-shaped*. Otherwise we say they are *A-shaped*. If a series is supported by a pair of V-shaped lines, then it is a *V-series*. Otherwise it is an *A-series*. (Fig. 4 shows a V-series.) These definitions are seen to be extensions of the previous definition of an A- or V-butterfly. (Remark: V and A are chosen for the shapes of these letters, being mnemonic for the orientations of the supporting lines—if we imagine the line $\overline{c_1 c_m}$ as horizontal and Q as sitting above $\overline{c_1 c_m}$.)

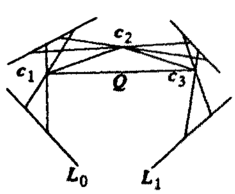

Fig. 4

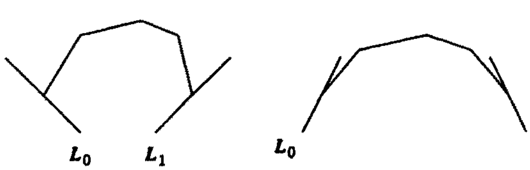

Fig. 5. A V-chain and an A-chain (showing only the truncated versions of the chords).

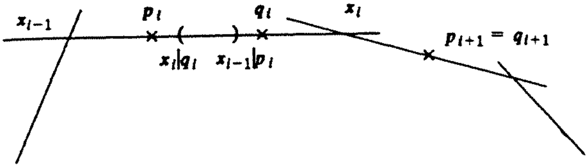

Fig. 6

Notation. Let p, q be distinct points. Then $p|q$ denotes the reflection of p about q, i.e. $p|q$ is a point on the line $\overline{pq}$ such that q is the midpoint between p and $p|q$.

Now we present some definitions for chords. Let (C_0, C_{m+1}) be a pair of chords, not necessarily distinct. A (C_0, C_{m+1})*-chain of chords* is a sequence of chords $(C_1, C_2, \ldots, C_m)$ such that C_i intersects C_{i+1} at x_i for $i = 0, \ldots, m$ and $Q = (x_0, x_1, \ldots, x_m)$ is a convex polygon. If (C_0, C_{m+1}) is A-shaped, then (C_0, C_{m+1})-chain is an *A-chain*. Otherwise it is a *V-chain*. (See Fig. 5.) If (C_0, C_{m+1}) is understood, we just say "chain." Q is called the *core* of the chain. Call $[x_{i-1}, x_i]$ the *truncated version* of the chord C_i $(i = 1, \ldots, m)$. The x_i's are the *nodes* of the chain. A chord C_i $(i = 1, \ldots, m)$ is said to be *balanced* in $\mathscr{C}$ if there exist reflex corners p_i, q_i (possibly $p_i = q_i$) with the following properties:

(1) p_i and q_i both lie in the truncated chord $[x_{i-1}, x_i]$. We may assume that p_i lies between x_{i-1} and q_i.
(2) The midpoint of $[x_{i-1}, x_i]$ lies in $[p_i, q_i]$. So if $p_i = q_i$ then p_i is the midpoint.
(3) (*Bracketing property*) $x_i|q_i$ lies between x_{i-1} and $x_{i-1}|p_i$. (See Fig. 6.) This property is so-called because we imagine $x_i|q_i$ to be a left bracket and $x_{i-1}|p_i$ to be a right bracket. We allow the case where $x_i|q_i = x_{i-1}|p_i$. Note that if $p_i = q_i$, then $x_{i-1}|p_i = x_i$ and $x_i|q_i = x_{i-1}$.

If $p_i \neq q_i$ we call (p_i, q_i) a *double-pivot*, otherwise it is a *single-pivot*. The concept of double-pivots is not relevant until the next section. The chain $\mathscr{C}$ is *balanced* if every chord in $\mathscr{C}$ is balanced in $\mathscr{C}$. If a balanced chain has only single-pivots then it is a *simply balanced* chain.

If $\mathscr{B} = (B_1, B_2, \ldots, B_m)$ is a series of butterflies, then a sequence of chords $(C_1, C_2, \ldots, C_m)$ is said to be *embedded* in $\mathscr{B}$ if each C_i is embedded in B_i. The sequence of chords in this definition need not be a chain (i.e., some C_i and C_{i+1} may not intersect).

Let $\mathscr{B} = (B_1, B_2, \ldots, B_m)$ be a series and $(C_1, C_2, \ldots, C_m)$ be a sequence of chords that is embedded in $\mathscr{B}$. We say the sequence is *optimal* (for $\mathscr{B}$) if the area of

$$P \cap \left(\bigcap_{i=1}^{m} C_i^+ \right)$$

is a local maximal, i.e., any sufficiently small perturbation of the chords produces a sequence with smaller area. A sequence of chords is *optimum* (for a series of butterflies) if its area is maximum over all sequences embedded in the series of

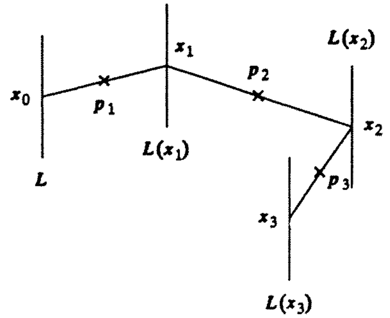

Fig. 7

butterflies. If the optimal or optimum sequence of chords turns out to be a chain, we call it an *optimal* or *optimum* chain.

Before proving a basic lemma next, we introduce a useful concept. Let L be a fixed line and let V be the corners of the polygon P and $\{p_1, \ldots, p_m\}$ be some subset of the reflex corners of P. An *$(L, p_1, \ldots, p_m)$-path* is a polygonal path of the form

$$\Pi = (x_0, x_1, \ldots, x_m),$$

where $x_0 \in L$ and for $i = 1, \ldots, m$ and p_i is the midpoint of $[x_{i-1}, x_i]$. For any point x and line L, let $L(x)$ denote the line through x parallel to L. Note that the line $L(x_i)$ is a function of L and $p_1, p_2, \ldots, p_i$ only (i.e., $L(x_i)$ is independent of $p_{i+1}, \ldots, p_m$ and the particular choice of Π). Furthermore for any $i = 0, \ldots, m$, any point x on $L(x_i)$ determines a unique $(L, p_1, \ldots, p_m)$-path and vice versa. In particular, the choice of any $x \in L$ determines a path. We can think of a path as a configuration in a system of "interconnecting levers;" each $[x_{i-1}, x_i]$ is a stretchable lever on the fixed axis p_i and nodes x_{i-1} and x_i are constrained to glide along parallel slots $L(x_{i-1})$ and $L(x_i)$, respectively. We call $x_0 \in L$ *critical* if the $(L, p_1, \ldots, p_m)$-path at x_0 has the property that for some i, either

(a) $[x_{i-1}, x_i]$ passes through some corner in $\{v_0, v_1, \ldots, v_{n-1}\} - \{p_1, \ldots, p_m\}$

or

(b) or x_{i-1}, x_i, x_{i+1} are collinear.

A minimal interval $I \subseteq L$ bounded by two critical points is called a *critical interval*. An $(L, p_1, \ldots, p_m)$-path is said to *belong* to I if its first node x_0 is in I. If the nodes $(x_0, x_1, \ldots, x_m)$ form the corners of a simple polygon Q in the indicated order, then we define the *area* of the path to be the area of Q. The area of Q depends only on $(L, p_1, \ldots, p_m)$ but not on the choice of Π. This *area-invariance* property is due to the fact that each p_i is the midpoint of the segment $[x_{i-1}, x_i]$, $i = 0, \ldots, m$ and hence as the "levers" go up and down, Q loses exactly as much area as it gains. We exploit this property to show:

Lemma 2 (*V-Lemma*). *Let $\mathscr{C} = (C_1, C_2, \ldots, C_m)$ be a chain that is optimal for a V-series of butterflies $\mathscr{B} = (B_1, B_2, \ldots, B_m)$. If the supporting lines of $\mathscr{B}$ are not parallel or coincident, then at least one of the C_i's is extremal. If the supporting lines are parallel or coincident, then we can modify the chords without decreasing the area defined by the chain so that at least one of the chords becomes extremal.*

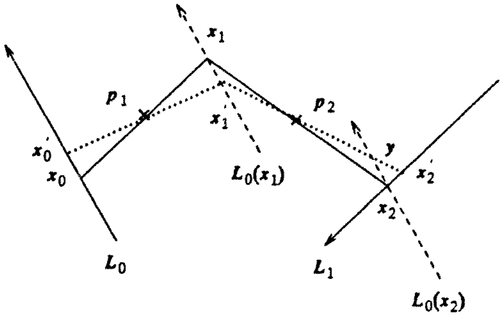

Fig. 8. The dotted chain represents the perturbation.

Proof. The case $m=1$ is the consequence of the Butterfly Lemma. So let $m>1$. For each $1 \le i \le m$, consider the butterfly B centered at p_i and supported by C_{i-1} and C_i. According to the Butterfly Lemma, C_i is either extremal or balanced when truncated. If C_i is extremal, then we are done. Thus assume without loss of generality that the truncated version of each chord C_i ($i=1,\dots,m$) is balanced. If $x_0, x_1, \dots, x_m$ are the nodes, then $(x_0, x_1, \dots, x_m)$ forms an $(L_0, p_1, \dots, p_m)$-path. However if L_0 and L_1 are not parallel, we can perturb the chord C_m out of balance while keeping all other chords balanced, resulting in another chain $\mathscr{C}' = (C_1', C_2', \dots, C_m')$ with nodes $(x_0', x_1', \dots, x_m')$. (See Fig. 8.)

Hence $(x_0', x_1', \dots, x_{m-1}')$ is an $(L_0, p_1, p_2, \dots, p_{m-1})$-path and C_m and C_m' are embedded in the V-shaped butterfly supported by $L_0(x_m)$ and $L_1(x_{m+1}) = L_1$. If m is even, and the perturbation x_0' is in a direction along L_0 away from the intersection of L_0 and L_1, then the area of $\mathscr{C}'$ is increased. If m is odd, we perturb in the opposite direction. More precisely, in addition to the area-invariance of the $(L_0, p_1, p_2, \dots, p_{m-1})$-path, C_m' contributes a gain in area that equals the area of $\triangle x_2 x_2' y$ in Fig. 8. Thus $\mathscr{C}'$ results in a larger area than $\mathscr{C}$, contradicting the local optimality of $\mathscr{C}$. Finally if the lines are parallel or coincident, we can perturb x_m along $L_0(x_m) = L_1$ without changing the area of the series since the perturbed series *is* an $(L_0, p_1, \dots, p_m)$-path. We can perturb x_m until the corresponding x_0 is critical, implying that one of the chords is extremal. □

We next investigate the considerably more subtle A-series of butterflies.

Lemma 3 (*A-Lemma*). *Let $\mathscr{B}$ be an A-series of butterflies. Then*

(a) *$\mathscr{B}$ has at most one simply balanced chain $\mathscr{C}^*$ embedded in it.*

(b) *Let $\mathscr{C}$ be an optimal sequence of chords embedded in $\mathscr{B}$. Then either $\mathscr{C}$ contains an extremal chord or $\mathscr{C}$ is a simply balanced chain* (*which is unique by* (a)).

Proof. (a) is proved in the next section in a slightly more general setting. To see (b), suppose C_i is an unbalanced chord in a chain with no extremal chords. Perturbing C_i toward the balance position (this is possible since C_i is not extremal), while keeping all other chords unchanged, increases the area of the series and therefore violates the local optimality. □

Lemma 4. *Let Q be a maximal convex polygon in P such that $Q = \bigcap_{i=1}^{m} C_i^+$ for some sequence $(C_0, C_1, \ldots, C_m)$ of chords of P. Then at least two chords in the sequence are extremal.*

Proof. Assume to the contrary that Q has 0 or 1 extremal chord. Note that $(C_2, \ldots, C_m)$ forms a (C_0, C_1)-chain. Without loss of generality, assume that this chain contains no extremal chords. Clearly $(C_2, \ldots, C_m)$ is a V-chain, and by the V-lemma, it is not optimal. □

The above lemmas provide us with a finiteness criterion in the sense that we can guess that Q is determined by a sequence $C_1, C_2, \ldots, C_k$ of extremal chords together with series of butterflies supported by (C_i, C_{i+1}) for $i = 1, \ldots, k$. This gives an exponential time algorithm *provided* that we can find simply balanced chains for any given series of butterflies in exponential time. We next show that such chains can in fact be found in polynomial time.

4. A Geometric Problem

The problem of finding the balanced chain in an A-series of butterflies can be reduced to an abstract geometric problem. First we transplant some notations from the previous section to a different geometric setting. We now assume a fixed set R of points in the plane. Let L^+ be the half-plane to the right of a directed line L. Given a pair of directed lines (L_0, L_1), let a *chord* denote a line segment contained in $L_0^+ \cap L_1^+$ passing through at least one point of R and with endpoints in L_0 and L_1. A chord is *extremal* if it passes through two or more points of R. As before the points of R in a chord are called the *pivots* of the chord. The definitions of chains and nodes are the same as in Section 3. Let $\mathscr{C} = (C_1, C_2, \ldots, C_m)$ be an (L_0, L_1)-chain with $(x_0, x_1, \ldots, x_m)$ as nodes. (See Fig. 9.) The convex polygon $Q = (x_0, x_1, \ldots, x_m)$ is the *core* of C. The polygonal path $(x_0, x_1, \ldots, x_m)$ partitions $L_0^+ \cap L_1^+$ into a finite and an infinite regions. Let $\rho(\mathscr{C})$ denote the open infinite region so defined. We say $\mathscr{C}$ is *empty* with respect to R if $\rho(\mathscr{C})$ does not contain any point of R. The definition of *balanced* chains is the same as in the previous section, with the set R playing the role of the corners of P. (As usual, assume for simplicity that no three points of R are collinear.) Now let us consider the following problem:

> Let R be the given set of points on the plane. For each pair of directed lines (L_0, L_1) such that L_0 and L_1 each passes through an ordered pair of points from R, find the balanced (L_0, L_1)-chain that is empty.

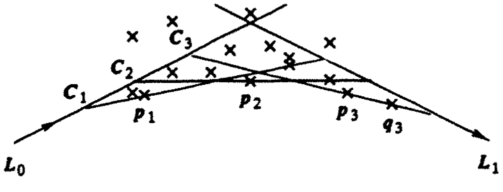

Fig. 9. (C_1, C_2, C_3) is an empty (L_0, L_1)-chain (the points of R are indicated by $\times$).

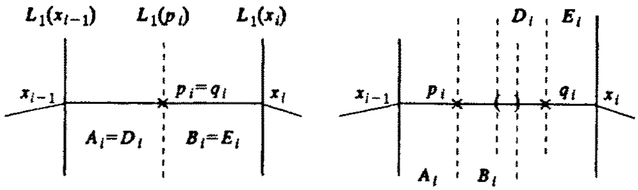

Fig. 10. The regions A_i, D_i, B_i and E_i (two cases).

Theorem 5. *Let* (L_0, L_1) *be a pair of directed lines and* R *be a set of points on the plane. Then there is a unique balanced* (L_0, L_1)*-chain that is empty with respect to* R.

Let us note that this theorem implies Lemma 3(a), as promised. We show a useful lemma along the way to proving this theorem. The following notations are needed for this lemma:

Notations. Let $\mathscr{C}$ be a balanced chain and let (p_i, q_i) be the pivot of the chord C_i in $\mathscr{C}$ (possibly $p_i = q_i$) and let $(x_0, x_1, \ldots, x_m)$ denote the nodes of $\mathscr{C}$. If $p_i \neq q_i$, then assume q_i lies between p_i and x_i. Recall that for any point p, $L_1(p)$ denotes the line through p and parallel to L_1. Define the region S_i to be the strip between the parallel lines $L_1(x_{i-1})$ and $L_1(x_i)$ where $L_1(x_{i-1})$ is excluded from S_i but $L_1(x_i)$ is included. Define the region A_i to be the strip between the parallel lines $L_1(x_{i-1})$ and $L_1(p_i)$: it is important to note that we exclude the line $L_1(x_{i-1})$ from A_i but include $L_1(p_i)$ in A_i. The region B_i is the strip between $L_1(p_i)$ and $L_1(x_{i-1}|p_i)$: again $L_1(p_i)$ is excluded but $L_1(x_{i-1}|p_i)$ is included. Similarly, D_i (resp. E_i) is the strip and between the lines $L_1(x_i|q_i)$ and $L_1(q_i)$ (resp. $L_1(q_i)$ and $L_1(x_i)$) where $L_1(x_i|q_i)$ (resp. $L_1(q_i)$) is excluded but $L_1(q_i)$ (resp. $L_1(x_i)$ is included. Thus $A_i \cap B_i = \varnothing$, $D_i \cap E_i = \varnothing$, and $S_i = A_i \cup B_i \cup D_i \cup E_i$. Recall that the chain $\mathscr{C}$ divides the quadrant $L_0^+ \cap L_1^+$ into a finite region and an infinite $\rho(\mathscr{C})$. It is convenient to regard this finite region to be *above* the chain and $\rho(\mathscr{C})$ to be *below* it. Thus each of the strips S_i, A_i, B_i, etc., will be broken up into two *half-strips*, above and below the chain, respectively. See Figs. 10 and 11 for these regions. Let $\alpha_i = A_i \cup D_i$ and $\beta_i = B_i \cup E_i$. Note

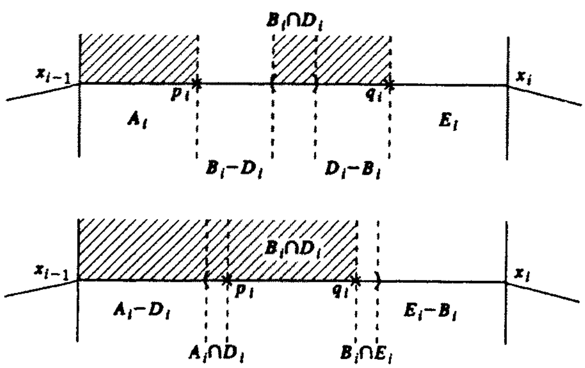

Fig. 11. The shaded areas are forbidden for nodes of $\mathscr{C}'$: two cases.

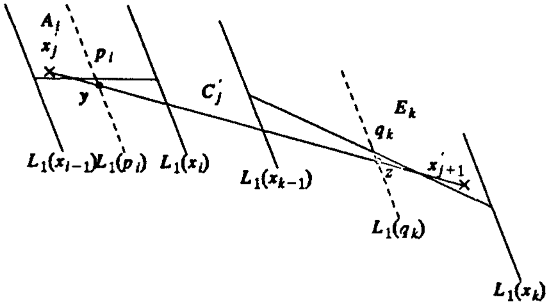

Fig. 12

that, except in the extreme case of $x_{i-1}|p_i = x_i|q_i$, $\alpha_i \cap \beta_i$ is nonempty because of the bracketing property. Also $p_i = q_i$ iff $A_i = D_i$ iff $B_i = E_i$.

Lemma 6. *Let L_0, L'_0, L_1 be distinct lines. Let $\mathscr{C} = (C_1, C_2, \ldots, C_m)$ (resp. $\mathscr{C}' = (C'_1, C'_2, \ldots, C'_{m'})$) be a balanced (L_0, L_1)-chain (resp. (L'_0, L_1)-chain). Let x'_js, p'_js be the nodes and pivots of $\mathscr{C}'$. If both $\mathscr{C}$ and $\mathscr{C}'$ are empty, then x'_j cannot lie above $\mathscr{C}$ in α_i for any i and j.*

Proof. Assume for the sake of contradiction that x'_j lie above $\mathscr{C}$ in α_i for some j and i. If $j = m'$, by definition $x'_{m'}$ lies in E_m and hence is not in any α_i. Choose j to be the largest index such that x'_j $(j < m')$ is above the chain in α_i for some i. Hence x'_{j+1} and x'_{j+2} are either lying below $\mathscr{C}$ or not in α_k for some k. We will prove that x'_{j+1} or x'_{j+2} lies above the chain in some α_k $(k > i)$, thereby contradicting our choice of j. We consider two cases next.

(a) Consider the case where x'_{j+1} is in S_k form some $k > i$. If x'_{j+1} lies below $\mathscr{C}$, then p'_{j+1} would be below $\mathscr{C}$. This contradicts the assumption that $\mathscr{C}$ is empty. Thus, x'_{j+1} lies above $\mathscr{C}$ in $S_k - \alpha_k$. Suppose x'_{j+1} lies above the chain in E_k (the proof for $B_k - \alpha_k$ is similar). C'_{j+1} must intersect $L_1(q_k)$ below the chain (otherwise q_k would be in $\rho(\mathscr{C}')$, contradicting the emptiness of $\mathscr{C}'$). Let z denote this intersection point. Similarly C'_{j+1} intersects $L_1(p_i)$ below C_i at some point y (see Fig. 12). Then the midpoint of the segment $[x'_j, x'_{j+1}]$ lies in $[y, z]$. Note that p'_{j+1} must lie to the right of y and q'_{j+1} to the left of z to satisfy the bracketing property. Since p'_{j+1} must lie to the left of q'_{j+1}, this implies both p'_{j+1} and q'_{j+1} lie below the chain $\mathscr{C}$, contradiction.

(b) Finally, consider the case when x'_{j+1} stays in S_i. If x'_{j+1} is below $\mathscr{C}$, then an argument similar to part (a) shows that x'_{j+2} is above $\mathscr{C}$ in α_k for some $k > i$. Otherwise x'_{j+1} above $\mathscr{C}$ in β_i and we have three possibilities: (i) $x'_j \in A_i$ and $x'_{j+1} \in \beta_i$, (ii) $x'_j \in D_i$ and $x'_{j+1} \in E_i$, (iii) both x'_j and x'_{j+1} are in $B_i \cap D_i$. Now (i) implies $p_i \in \rho(\mathscr{C}')$ and (ii) implies $q_i \in \rho(\mathscr{C}')$ contradicting the emptiness of $\mathscr{C}'$. If (iii) holds, then this contradicts our choice of j. □

Proof of Theorem 5. There are two parts to this theorem: (i) there cannot be more than one balanced chain and (ii) there exists at least one balanced chain.

(Uniqueness). Let $\mathscr{C} = (C_1, C_2, \ldots, C_m)$ be a balanced chain. We derive a contradiction by assuming the existence of another balanced chain $\mathscr{C}' = (C'_1, C'_2, \ldots, C'_{m'})$. Let p_i, x_i and p'_i, x'_i be the pivots and nodes of $\mathscr{C}$ and $\mathscr{C}'$,

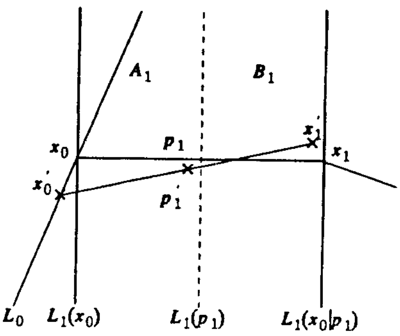

Fig. 13

respectively. To apply the previous lemma we only have to show that x_i' lies in α_{j+1} for some i, j. Initially assume $x_0' \neq x_0$. Suppose x_0' is below x_0 on L_0. (See Fig. 13). Observe that x_1' must lie above $\mathscr{C}$ otherwise p_1' is below $\mathscr{C}$. If x_1' lies in any α_i, we are done. So let x_1' lie in $\beta_i - \alpha_i$. Note that x_1' is to the right of $L_1(x_0|p_1)$. If x_1' is in B_1, then p_1' is below $\mathscr{C}$, a contradiction. If x_1' is in E_1, let z be the intersection of C_1' with $L_1(q_1)$. Then z must be below $\mathscr{C}$. Note that the midpoint of C_1' lies to left of z and hence p_1' lies to the left of z. This implies p_1' is below $\mathscr{C}$, a contradiction. Therefore x_0' cannot lie below x_0. If x_0' is above x_0 on L_0, then a symmetrical argument applies by exchanging the roles of $\mathscr{C}$ and $\mathscr{C}'$.

It remains to consider the possibility $x_0' = x_0$. Since $\mathscr{C} \neq \mathscr{C}'$, let x_{j+1}' be the first node such that $x_{j+1}' \neq x_{j+1}$. Suppose x_{j+1}' does not lie in the line through $[x_j, x_{j+1}]$. An analysis similar to the above shows that x_{j+1}' lies in α_k for some k. Finally suppose x_{j+1}' lies in the line through $[x_j, x_{j+1}]$: if it lies in the segment $[x_j, x_{j+1}]$, then x_{j+2}' must lie in some α_k by the same argument as the previous case. Otherwise, x_{j+1}' clearly is in A_{j+2} and again we have a contradiction.

(Existence). The existence proof can be regarded as an algorithm, although we do not know of a polynomial time bound on its complexity.

We now give a "scan line" algorithm for computing the balanced (L_0, L_1)-chain. A sequence of $(L_0(t), L_1)$-chains are computed where $L_0(t)$ is a line parallel to L_0 at a distance t to the left of L_0. Imagine the line $L_0(t)$ moving from the infinite left toward L_0 as t approach 0 from ∞. As $t \to 0$, $L_0(t) \to L_0$ and the $(L_0(t), L_1)$-chain becomes the (L_0, L_1)-chain. During the process, there are events, $t_0, t_1, \ldots, t_m$ that divide the scanning process into intervals where changes in the $(L_0(t), L_1)$-chain are smooth within each interval. More specifically, between events the pivots remain the same and the slope of each chord changes at a smooth rate (with respect to t). Initially the $(L_0(\infty), L_1)$-chain consists of an infinite chord parallel to L_1 through the point p_1 in $R \cap L_1^+$ that is farthest from L_1. Then this chord turns continuously counterclockwise about p_1 as $L_0(t)$ gradually moves toward L_0 until the chord hits a new point q_1 in R and a double pivot (p_1, q_1) is formed. This is the first critical moment t_0.

As $L_0(t)$ continues to move closer to L_0 and x_0 moves toward x_1 (as usual, x_i denotes the ith node of the balanced chain, C_i is the ith chord, etc.), $x_0|p_1$ moves toward both $x_1|q_1$ and p_1 and the midpoint r_1 of C_1 moves toward q_1. Two things can happen at the next critical moment $t = t_1$: either (a) $x_0|p_1$ meets

$x_1|q_1$ and the double-pivot chord "splits" into two single-pivot chords balanced at p_1 and q_1 or else (b) r_1 meets q_1 and the chord starts turning counterclockwise around q_1 "leaving" p_1 behind. (a) If the splitting event occurs first, we will have a chain of size 2, and as $L_0(t)$ moves further in, in order to balance C_1 and C_2, x_0 and x_2 move inwards and x_1 moves outwards along the direction of L_1. (Note: "inwards" and "outwards" are with respect to any point inside the core Q of the chain.) This continues until $t = t_2$ when a point in R is hit by either C_1 or C_2 and turns a single-pivot chord into a double-pivot chord. (b) If the leaving event occurs first, the double-pivot chord turns into a single-pivot chord while continuing to turn counterclockwise until $t = t_2$ when it hits another point of R and turns itself into a double-pivot chord. So this process of forming double pivots, splitting and leaving continues until $t = t_m$ when $L_0(t)$ reaches L_0.

In general, consider the $(L_0(t), L_1)$-chain $(C_1, C_2, \ldots, C_m)$. Let C_{j_0} (for some $j_0, 1 \le j_0 \le m+1$) denote the leftmost double-pivot chord where we choose $j_0 = m+1$ and $C_{j_0} = L_1$ if there are no double-pivot chords. As $L_0(t)$ moves, the even-numbered nodes $(x_{2i}, 2i < j_0)$ move inwards while odd-numbered nodes $(x_{2i+1}, 2i+1 < j_0)$ move outwards along the direction parallel to C_{j_0}. Observe that C_i, for $i = 1, \ldots, j_0 - 1$, turns clockwise if i is even and counterclockwise otherwise. This implies that the length of the truncated version of C_{j_0} decreases iff j_0 is even. The rest of the $(L_0(t), L_1)$-chain remains unchanged. We can classify the possible events into five categories:

(1) (Flattening) Two consecutive chords C_{2i} and C_{2i+1} could *flatten* out and become a double-pivot chord. It should be noted that C_{2i-1} and C_{2i} cannot flatten out.

(2) (Hitting) One of the single-pivot chord could *hit* a point in R thus becomes a double-pivot chord.

(3) (Splitting) If j_0 is odd, the truncated version of C_{j_0} could shorten to such an extent that $x_{j_0-1}|p_{j_0}$ meets $x_{j_0}|q_{j_0}$. Then C_{j_0} loses the bracketing property and *splits* into two single-pivot chords.

(4) (Leaving p_{j_0}) When C_{j_0} is shortening, another situation could also arise. The midpoint r_{j_0} could meet q_{j_0} before splitting occurs. Subsequently, C_{j_0} will turn counterclockwise around q_{j_0} *leaving* p_{j_0} behind. In effect, C_{j_0} turns into a single-pivot chord balanced at q_{j_0}.

(5) (Leaving q_{j_0}) Finally, if j_0 is even the chord C_{j_0} is lengthening and the only event that can happen is the midpoint r_{j_0} moving left and meeting p_{j_0}. Subsequently, C_{j_0} will pivot clockwise around p_{j_0} *leaving* q_{j_0} behind.

The algorithm first establishes a balanced chain for $(L(t_0), L_1)$ and then repeats the following step until $L_0(t)$ reaches L_0:

Find the leftmost double-pivot chord, C_{j_0} on the current $(L_0(t), L_1)$-chain. In the case of a chain consisting of just single-pivot chords, take L_1 for C_{j_0}. It is not too hard to compute the values of t when flattening, leaving or splitting occur at each C_i, since those are determined by the chain alone. The hitting event at each C_i is more difficult to compute since it involves points not on the chain. But it is clear the point hit by C_i has to be on the convex hull of some subset of R containing p_i. So we partition the plane into j_0 strips and two half-planes by the lines $L_{j_0}(t_i)$, $i = 0, \ldots, j_0$ where $L_{j_0}(t_i)$ is the line through x_i parallel to C_{j_0}. Let H_i be the convex hull of those points of R in the strip between $L_{j_0}(t_{i-1})$ and $L_{j_0}(t_i)$. The point hit by an even-numbered chord C_{2i} is the point on H_{2i}

clockwise from p_{2i}; the point hit by C_{2i+1} is the next point on H_{2i+1} counterclockwise from p_{2i+1}. So for each i we can compute the value of t when an event involving C_i occurs. The next event at $t = t_{k+1}$ is determined by one with the smallest of such t values. To complete the present step, we just update the $(L_0(t), L_1)$-chain accordingly.

This concludes our proof of the existence of a unique balanced chain for any (L_0, L_1).

5. Decomposition of Balanced Chains

Let R be a fixed set of n points, and $\{p_1, \dots, p_m\} \subseteq R$. Recall the definition of $(L, p_1, \dots, p_m)$*-paths*, *critical points*, and *critical intervals* in Section 3. In this section we use these concepts to describe certain processes for composing a chain from smaller chains, and for decomposing a chain into smaller ones.

Lemma 7. *Given $(L_0, p_1, \dots, p_m)$, we can determine all the critical intervals of $(L_0, p_1, \dots, p_m)$-paths in $O(n \log n)$ time.*

Proof. Pick an arbitrary point x_0 on L_0. Form the unique path $\Pi = (x_0, x_1, \dots, x_m)$ and divide the plane into strips determined by the parallel lines $L_0(x_i)$. This takes linear time. For each $r \in R$, determine in $O(\log m) = O(\log n)$ time the index i, $1 \le i \le m$, such that r and p_i are in the same strip. Then we can in $O(1)$ time determine the critical point $y \in L_0$ corresponding to the $(L_0, p_1, \dots, p_m)$-path that passes through r. Also for each $i = 1, \dots, m-1$, we can in $O(1)$ time determine the critical point corresponding to the $(L_0, p_1, \dots, p_m)$-path where (x_{i-1}, x_i, x_{i+1}) are collinear. □

For two points x, y on the line $L_0(x_i)$ (for any i), it is convenient to say that they are $(L_0, p_1, \dots, p_m)$*-equivalent* if the path through x and y both belong to the same critical interval. We define the critical interval *containing* a chain to be the one which contains the first node x_0.

We now describe a *decomposition process*: Let $\mathscr{C} = (C_1, \dots, C_m)$ be a simply balanced (L_0, L_1)-chain below a set R of points with pivots $p_1, \dots, p_m$ and nodes $(x_0, x_1, \dots, x_m)$, $m > 0$. $\mathscr{C}$ determines a corresponding $(L_0, p_1, \dots, p_m)$-path, $(x_0, x_1, \dots, x_m)$. If m is even (resp. odd), we consider moving the point x_0 along L_0 in the direction towards (resp. away from) the intersection of L_0 and L_1. We move x_0 until it reach the first critical point x_0. Recall that if $(x'_0, x'_1, \dots, x'_m)$ is the $(L_0, p_1, \dots, p_m)$-path at x'_0, then for some $k \ge 0$ (see Fig. 14)

(a) either $[x'_k, x'_{k+1}]$ passes through a point r in $R - \{p_1, \dots, p_m\}$, or

(b) x'_k, x'_{k+1}, x'_{k+2} are collinear.

It should be noted at this moment that our decision to move x_0 in the chosen direction (depending on the parity of m) implies that x'_m lies outside the quadrant $L_0^+ \cap L_1^+$ and more importantly, the segment $[x'_{m-1}, x'_m]$ does not become parallel to L_1.

First consider possibility (a). For clarity, we will initially assume that possibilities (a) and (b) do not occur simultaneously and that r is unique. Let L_2 be the directed line from p_{k+1} to r. For $i = 1, \dots, k$, let C'_i denote the chord determined

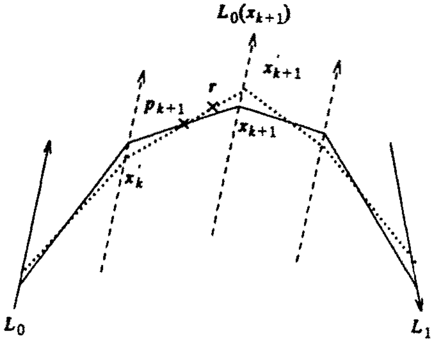

Fig. 14. $[x'_k, x'_{k+1}]$ passes through r.

by $[x'_{i-1}, x'_i]$. Observe that

$$\mathscr{C}' = (C'_1, \ldots, C'_k)$$

is a balanced (L_0, L_2)-chain. However $(C'_{k+1}, \ldots, C'_m)$ does not represent a balanced (L_2, L_1)-chain because the pivot p_m is at the midpoint of $[x'_{m-1}, x'_m]$ but x'_m is not on L_1. To obtain a balanced (L_2, L_1)-chain, we continue as follows: For $i = k+1, \ldots, m$, define the point

$$x''_i = L_1(x_i) \cap L_2(x'_i).$$

Note that the set of triangles (see Fig. 15)

$$\triangle x_i x'_i x''_i \qquad (i = k+1, \ldots, m)$$

are congruent. Let C''_i be the chord determined by $[x''_{i-1}, x''_i]$. Then we note that

$$\mathscr{C}'' = (C''_{k+1}, \ldots, C''_m)$$

is a balanced (L_2, L_1)-chain. Consider the $(L_2, p_{k+1}, p_{k+2}, \ldots, p_m)$-path corresponding to $\mathscr{C}''$: It is important to see from our construction that x_{k+1} and x'_{k+1} lies in the same critical interval with respect to $(L_2, p_{k+1}, p_{k+2}, \ldots, p_m)$, i.e., as we move from x''_{k+1} to x'_{k+1}, the $(L_2, p_{k+1}, p_{k+2}, \ldots, p_m)$-paths encountered along the way are noncritical.

Now consider possibility (b) where again we initially assume for simplicity that the k such that x'_k, x'_{k+1}, x'_{k+2} are collinear is unique. Let L_2 be the line through x'_k and x'_{k+1}. As before we immediately obtain an (L_0, L_2)-chain $\mathscr{C}'$ of length k. It is not hard to see that we can define an (L_2, L_1)-chain $\mathscr{C}''$ of length $m - k - 2$ by the same method as above.

This completes our decomposition process for $\mathscr{C}$. The resulting pair of chains $\mathscr{C}'$ and $\mathscr{C}''$ will be called the *decomposition* of $\mathscr{C}$. It should be noted however, that $\mathscr{C}'$ and $\mathscr{C}''$ are in general *not* below the set R. Rather $\mathscr{C}'$ and $\mathscr{C}''$ are below some sets $R(L_0, L_2)$ and $R(L_2, L_1)$, respectively, where $R(L_0, L_2) \cup R(L_2, L_1) = R$. As it turns out in our application, we *do* know the sets $R(L_0, L_2)$

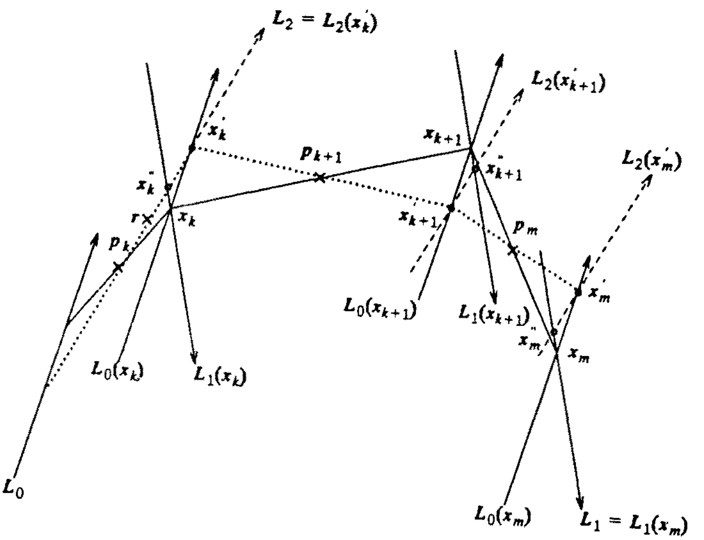

Fig. 15

and $R(L_2, L_1)$. To show the dependence of (L_0, L_1)-chains on the set R, we may also call it an (L_0, L_1, R)-chain. Thus we have shown constructively that every chain of length $m > 0$ can be decomposed into two chains of length $< m$.

We have assumed that k is unique in (a) and (b) above. It is not hard to provide the modification necessary for the general case. $\mathscr{C}$ is decomposed into more than two chains if the k in (a) and (b) is not unique.

We next consider how the above process may be reversed, i.e., given two balanced chains, check if they form the decomposition of some chain $\mathscr{C}$ and if so, construct $\mathscr{C}$.

Let $\mathscr{C}'$ be the balanced (L_0, L_2)-chain below R' and $\mathscr{C}''$ the balanced (L_2, L_1)-chain below R''. If $\mathscr{C}'$ and $\mathscr{C}''$ form the decomposition of some $\mathscr{C}$ below the set $R = R' \cup R''$, then this could come about by the decomposition process in one of the two ways corresponding to possibilities (a) and (b) above.

(A) We first verify whether case (a) holds: With the usual notations for $\mathscr{C}'$ and $\mathscr{C}''$ (viz., $\mathscr{C}'$ is an (L_0, L_2, R')-chain of length m', C'' is an (L_2, L, R'')-chain of length m'', x_j' and x_j'' are the nodes of $\mathscr{C}'$ and $\mathscr{C}''$ respectively, etc.), let L_2 be a line through p and q in R. We first verify in constant time that $x'_{m'}$, p, q, x_0'' occur in that order in L_2 and that the two chains do not intersect (see Fig. 16). Then we proceed as follows: Let $m = m' + m'' + 1$ and set

$$p_i = \begin{cases} p_i' & \text{if } i = 1, \ldots, m' \\ p & \text{if } i = m'+1 \text{ and } m'' = \text{even} \\ q & \text{if } i = m'+1 \text{ and } m'' = \text{odd} \\ p''_{i-m'-1} & \text{if } i = m'+2, \ldots, m. \end{cases}$$

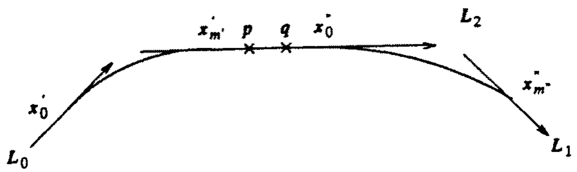

Fig. 16

Set

$$x'_{m'+1} = x'_{m'}|p_{m'+1},$$
$$x_{m'+1} = L_1(x''_0) \cap L_0(x''_{m'+1}).$$

Let Π' be the $(L_0, p'_1, \ldots, p'_{m'+1})$-path corresponding to moving from the node $x'_{m'+1}$ to $x_{m'+1}$ and Π'' the $(L_1, p''_1, p''_2, \ldots, p''_{m''})$-path corresponding to moving x''_0 to $x_{m'+1}$. (See Fig. 17.) Verify that $x_{m'+1}$ and $x'_{m'+1}$ are $(L_0, p'_1, \ldots, p'_{m'+1})$-equivalent and $x_{m'+1}$ and x''_0 are $(L_1, p''_1, p''_2, \ldots, p''_{m''})$-equivalent. It is easy to see that these two equivalence conditions hold if and only if $\mathscr{C}'$ and $\mathscr{C}''$ form a decomposition of the (L_0, L_1)-balanced chain $\mathscr{C}$.

(B) To verify if case (b) holds, we proceed in essentially the same way: Let $m = m' + m'' + 2$ and

$$p_i = \begin{cases} p'_i & \text{if } i = 1, \ldots, m' \\ p & \text{if } i = m' + 1 \\ q & \text{if } i = m' + 2 \\ p''_{i-m'-2} & \text{if } i > m' + 2. \end{cases}$$

Set $x'_{m'+1} = x'_{m'}|p$ and $x'_{m'+2} = x'_{m'+1}|q$. We should verify that the $\mathscr{C}'$ and $\mathscr{C}''$ do not intersect and

$$x'_{m'}, \quad p, \quad q, \quad x''_0, \quad x'_{m'+2}$$

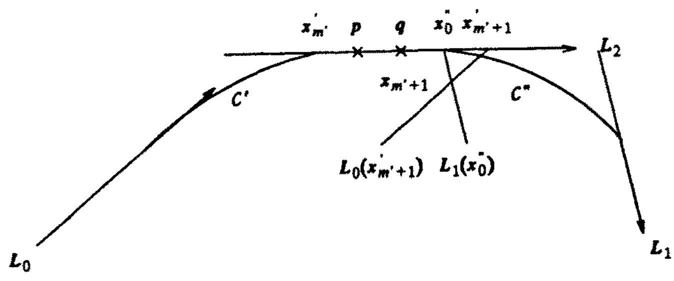

Fig. 17

occur in that order on L_2. This being the case, set

$$x_{m'+2} = L_1(x''_0) \cap L_0(x'_{m'+2}).$$

We must verify that $x_{m'+2}$ and $x'_{m'+2}$ are $(L_0, p'_1, \ldots, p'_{m'})$-equivalent and $x_{m'+2}$ and x''_0 are $(L_1, p''_{m''}, p''_{m''-1}, \ldots, p''_1)$-equivalent to ensure the validity of the *composition* process.

If $\mathscr{C}'$ and $\mathscr{C}''$ form the decomposition of $\mathscr{C}$ then we can easily reconstruct $\mathscr{C}$. Hence:

Lemma 8. *Given the balanced (L_0, L_2, R')-chain $\mathscr{C}'$, the balanced (L_2, L_1, R'')-chain $\mathscr{C}''$ and the critical intervals of these chains, it takes $O(1)$ time to verify if the balanced $(L_0, L_1, R' \cup R'')$-chain $\mathscr{C}$ decomposes into $\mathscr{C}'$ and $\mathscr{C}''$. It takes $O(n)$ time to construct $\mathscr{C}$ when it exists.*

Note that when computing critical intervals for $(L_0, p_1, \ldots, p_m)$-path on L_0, we need to extend the path to include one of the pivots on L_1. The reason is if the chain is to compose with another chain to its right, then those critical intervals are needed.

6. A Potato-Peeling Algorithm

By a *potato* of P, we mean any maximum area convex polygon contained in P. Although our goal is to compute the potato itself, it is convenient to describe the algorithm for computing the *area* of the potato. It is easy to modify the algorithm to compute the potato in addition to its area.

We now define some useful notations and data structures. For points r, s on the boundary of P, let $P[r, s]$ be the polygon whose corners are r and s together with the corners in P occurring clockwise from r to s. Note that $P[r, s]$ is a simple polygon iff the segment $[r, s]$ is contained in P. Let $C_{i,j}$ denote the oriented chord from v_i to v_j if the segment $[v_i, v_j] \subseteq P$, otherwise $C_{i,j}$ is undefined. For each corner v_i of P, define Ξ_i to be the set of extremal chords from v_i:

$$\Xi_i = \{C_{i,j} | C_{i,j} \text{ is defined}\}.$$

Clearly $|\Xi_i| < n$. It is easy to compute all the sets Ξ_i in $O(n^2 \log n)$ time. Let Ξ denote the union of all the Ξ_i's.

Let $(C_{i,j}, C_{k,l})$ be a pair of directed chords from Ξ. Then define $R(C_{i,j}, C_{k,l})$ be the corners of P clockwise from v_j to v_k (inclusive). Let L_0 and L_1 be the directed lines obtained by extending $C_{i,j}$ and $C_{k,l}$, respectively. From now on, we simply refer to the balanced $(L_0, L_1, R(C_{i,j}, C_{k,l}))$-chain below the set $R(C_{i,j}, C_{k,l})$ as the $(C_{i,j}, C_{k,l})$-chain.

Definition. A pair of chords $(C_{i,j}, C_{k,l})$ is *admissible* if the corners v_i, v_j, v_k, v_l occur in this cyclic order on the boundary of P. A $(C_{i,j}, C_{k,l})$-chain is also called *admissible* if $(C_{i,j}, C_{k,l})$ is admissible and the first and last nodes, x_0 and x_m (for some m), lie in $C_{i,j}$ and $C_{k,l}$, respectively. A (C, C'')-chain $\mathscr{C}$ and a (C'', C')-chain

$\mathscr{C}'$ are *compatible* if

(1) Both $\mathscr{C}$ and $\mathscr{C}'$ are admissible.

(2) The last node x_m (for some m) of $\mathscr{C}$, the first node x'_0 of $\mathscr{C}'$ and the pivots (p, q) of C'' occur in the following order:

$$x_m, p, q, x'_0.$$

For any chord $C_{i,j} \in \Xi$, we call the point midway between the double-pivot v_i and v_j the *reference* point of $C_{i,j}$.

Before giving an algorithm for the potato-peeling problem, we first introduce an area measure for polygons. Let $P = (v_0, v_1, \ldots, v_{n-1})$ be a polygon that is not necessarily simple, with n corners where (x_i, y_i) is the coordinate of corner v_i, $i = 0, \ldots, n-1$. The area of P is defined as

$$2 \times \text{AREA}(P) = \sum_{i=0}^{n-1} x_i(y_{i+1} - y_i).$$

This is called the *signed area* [14] of P. If P is simple polygon, then this definition gives the expected notion of area, with a positive sign if the corners are given in counterclockwise order and negative otherwise. Recall that we previously define the "area" of a chain or path. We now redefine the *area* of a (C, C')-*chain* to be AREA$((c, x_0, x_1, \ldots, x_m, c'))$ where c and c' are the respective reference points of C and C' and the x_i's are nodes of the (C, C')-chain.

The following algorithm is described in three main steps.

Step I. We introduce the matrix A indexed by pairs of chords such that for $C, C' \in \Xi$, $A(C, C')$ is the area of the unique balanced (C, C')-chain, if it is admissible. Otherwise $A(C, C') = -\infty$. Initially set $A(C, C')$ to $-\infty$ for all (C, C'). Then for each admissible pair (C, C'), find $R(C, C')$. It is easy by a brute force method to compute all the (C_0, C_1)-chains of lengths 0 or 1 in time $O(n^6)$. Note that the chain has length 0 precisely where $R(C, C') \cap C^+ \cap C'^+$ is empty.

To compute admissible chains of all lengths, we proceed in stages. The first stage is the computation of admissible chains of lengths 0 and 1. At stage $i+1$ we compute more admissible chains by composing admissible chains computed in the previous stages. To compute a (C, C')-chain at stage $i+1$, we iterate through all chords C'' checking whether the (C, C'')- and (C'', C')-chains have been computed in previous stages and whether they form a decomposition of the (C, C')-chain. If so, construct the (C, C')-chain. At the same time, we should compute the critical intervals for the (C, C')-chain with respect to the enlarged set of points, $R(C, C')$: this takes $O(n \log n)$ time as shown in Lemma 7. In order to facilitate composition of chains, we can determine at the same time the critical interval to which the first node of the (C, C')-chain belongs.

To analyze the complexity of this procedure, we divide the cost into two parts: (i) the cost for verifying the possibility of composition and (ii) the cost of actually composing the chains and computing the critical intervals. There are $O(n)$ stages. At each stage, we go through all triples (C, C'', C') verifying if it is possible to compose the (C, C'')- and (C'', C')-chains. Hence, at each stage, there are n^6 instances of testing for a possible composition where each test takes $O(1)$ time. Since there are n stages, the cost of part (i) is $O(n^7)$. Because of the

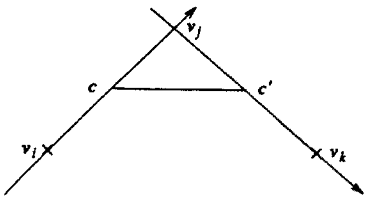

Fig. 18

uniqueness of the balanced chains, there are at most $O(n^4)$ chains that are composed during the entire procedure. The cost of part (ii) is $O(n^5 \log n)$ since each composition and computation of critical intervals take $O(n \log n)$ time. Thus the total cost of computing the A matrix is $O(n^7)$.

Step II. First we introduce the matrix M with entries indexed by pairs (C, C') of extremal chords. $M(C, C')$ is the maximum area of admissible (C, C')-chains: $M(C, C') = -\infty$ if (C, C') is not an admissible pair of chords. M can be computed in stages where in stage s $(s = 1, 2, \ldots, n-1)$ we compute the entries $M(C, C')$ where

$$C \in \Xi_i, \quad C' \in \Xi_j, \quad \text{and } j - i = s (\bmod n).$$

Note that in stage $s = 1$, we have $j = i + 1 (\bmod n)$ and the constraint that (C, C') forms an admissible pair implies that C must be the chord $C_{i,j}$. Let c and c' be the reference points of C and C', respectively. Then $M(C, C')$ is given by the area of $\triangle c v_j c'$. (See Fig. 18.) In general, for stage $s > 1$, we use the recursive formula:

$$M(C, C') = \max\left\{A(C, C'), \max_{C''}\{M(C, C'') + M(C'', C') + \text{AREA}(\triangle cc''c')\}\right\},$$

where C'' ranges over Ξ_k, k has the range $i < k < j$, and c'' is the reference point of C''. (See Fig. 19.) To justify this formula, note that $M(C, C')$ is either determined by the balanced (C, C')-chain or else it is determined by a chain that has an extremal chord $C'' \in \Xi_k$. Note that we do not check compatibility between $M(C, C'')$ and $M(C'', C')$. If they are not compatible, then they cannot form an optimal chain. Thus it is not necessary to exclude them from the maximization. It takes $O(n^2)$ steps to carry out the maximization for each entry of M. Thus the whole matrix M takes $O(n^6)$ steps in total to compute.

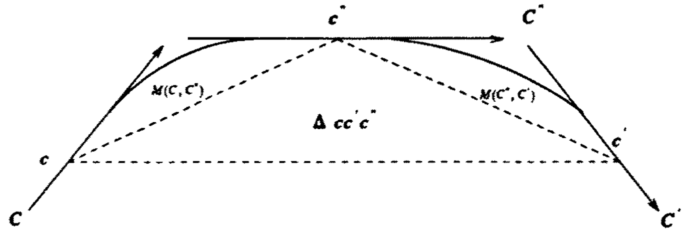

Fig. 19

Step III. The potato-peeling problem is now solved: We know that there are $m \geq 2$ extremal chords that form part of the boundary of the potato. It can be found in $O(n^4)$ time as follows:

$$\max\{M(C,C') + M(C',C): C \in \Xi_i, C' \in \Xi_j, \text{ for all } i \text{ and } j\}.$$

Notice again we do not check compatibility between $M(C,C')$ and $M(C',C)$.

7. Potato Peeling—Perimeter Measure

The potato-peeling problem under the perimeter measure can be solved in the slightly better time bound $O(n^6)$ by using essentially the same techniques. However, some additional properties here make a much simpler algorithm possible.

First of all, we give a finiteness criterion for the problem. Let us consider the simplest case of the perimeter optimization problem where the given polygon has just one reflex corner, say v_0. As with the area measure, the problem is to determine the chord $C = [c, c']$ through v_0 that maximizes the perimeter of the convex polygon $P \cap C^+$.

We shall prove that C may be assumed to be one of the two extremal chords of the butterfly containing C. Thus in contrast to the area measure, we need not consider a third possibility (such as C being balanced).

Consider the butterfly B that contains C. Let L_0 and L_1 be the two supporting lines of B. If L_0 and L_1, the two supporting lines of B, are parallel, then the perimeter is independent of the choice of C; we are done. Hence assume that L_0 and L_1 intersect at a point o. Referring to Fig. 20, let

$$\begin{aligned} \alpha &= \angle a'ov_0, \\ \beta &= \pi - \angle aov_0, \\ \theta &= \angle ov_0c. \end{aligned}$$

Note that $0 \leq \alpha < \beta < \pi$. Then θ satisfies $\alpha < \theta < \beta$. And by the law of sines we

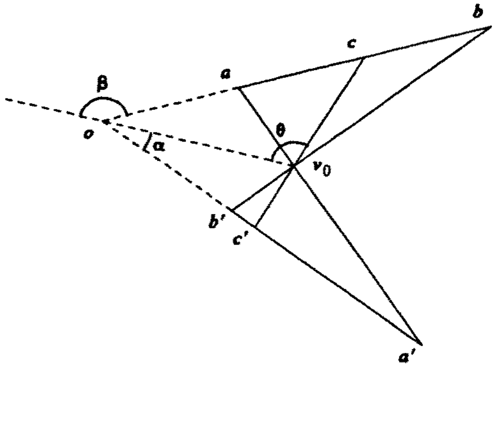

Fig. 20. The perimeter determined by a chord in a butterfly.

have

$$\frac{|oc|}{\sin\theta} = \frac{|\nu_0 c|}{\sin(\pi-\beta)} = \frac{|o\nu_0|}{\sin(\beta-\theta)},$$

$$\frac{|\nu_0 c'|}{\sin\alpha} = \frac{|oc'|}{\sin(\pi-\theta)} = \frac{|o\nu_0|}{\sin(\theta-\alpha)}.$$

Without loss of generality, let us assume $|o\nu_0| = 1$. Hence we have

$$|cc'| = |c\nu_0| + |\nu_0 c'| = \frac{\sin\beta}{\sin(\beta-\theta)} + \frac{\sin\alpha}{\sin(\theta-\alpha)},$$

$$|oc| = \frac{\sin\theta}{\sin(\beta-\theta)},$$

$$|oc'| = \frac{\sin\theta}{\sin(\theta-\alpha)}.$$

Notation. Given any two points a and b on the boundary of P, let $S[a,b]$ denote the length of polygonal path clockwise between a and b if the path contains no reflex vertex, and $-\infty$ otherwise.

We now obtain the perimeter L as a function of θ. It suffices to show that L has no local maximum to prove that C is extremal. We consider the following two cases:

(1) B is a V-butterfly: C^+ contains o and the perimeter $L(\theta)$ of the convex polygon determined by the chord C is

$$\begin{aligned}
L(\theta) &= |ac| + |cc'| + |c'b'| + S[b',a] \\
&= |oc| + |cc'| + |c'o| - |oa| - |b'o| + S[b',a] \\
&= |oc| + |cc'| + |c'o| + \text{constant} \\
&= \frac{\sin\theta+\sin\alpha}{\sin(\theta-\alpha)} + \frac{\sin\theta+\sin\beta}{\sin(\beta-\theta)} + \text{constant},
\end{aligned}$$

$$\begin{aligned}
\frac{dL}{d\theta} &= \frac{\cos\theta\sin(\theta-\alpha)-(\sin\theta+\sin\alpha)\cos(\theta-\alpha)}{\sin^2(\theta-\alpha)} \\
&\quad + \frac{\cos\theta\sin(\beta-\theta)+(\sin\theta+\sin\beta)\cos(\beta-\theta)}{\sin^2(\beta-\theta)} \\
&= -\frac{\sin\alpha(1+\cos(\theta-\alpha))}{\sin^2(\theta-\alpha)} + \frac{\sin\beta(1+\cos(\beta-\theta))}{\sin^2(\beta-\theta)} \\
&= \frac{-\sin\alpha}{1-\cos(\theta-\alpha)} + \frac{\sin\beta}{1-\cos(\beta-\theta)},
\end{aligned}$$

$$\frac{d^2L}{d\theta^2} = \frac{\sin\alpha\sin(\theta-\alpha)}{(1-\cos(\theta-\alpha))^2} + \frac{\sin\beta\sin(\beta-\theta)}{(1-\cos(\beta-\theta))^2}.$$

Since α, β, $\theta-\alpha$, and $\beta-\theta$ are all in the first two quadrants, $d^2L/d\theta^2 > 0$.

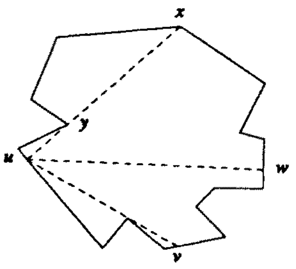

Fig. 21. Chord $[u, v]$ is semiextremal, but $[u, w]$ is not.

(2) B is an A-butterfly: Similarly we have

$$\begin{aligned} L(\theta) &= |a'c'| + |c'c| + |cb| + S[b, a'] \\ &= -|c'o| + |c'c| - |co| + |a'o| + |bo| + S[b, a'] \\ &= -|oc'| + |c'c| - |co| + \text{constant} \\ &= \frac{\sin\alpha - \sin\theta}{\sin(\theta - \alpha)} + \frac{\sin\beta - \sin\theta}{\sin(\beta - \theta)} + \text{constant}, \end{aligned}$$

$$\frac{dL}{d\theta} = \frac{\sin\alpha}{1 + \cos(\theta - \alpha)} - \frac{\sin\beta}{1 + \cos(\beta - \theta)},$$

$$\frac{d^2L}{d\theta^2} = \frac{\sin\alpha\sin(\theta - \alpha)}{(1 + \cos(\theta - \alpha))^2} + \frac{\sin\beta\sin(\beta - \theta)}{(1 + \cos(\beta - \theta))^2} > 0.$$

In both cases $L(\theta)$ does not have any local maximum. We have proved the following lemma:

Lemma 9. *For a polygon with one reflex corner, the maximal perimeter is determined by an extremal chord through the reflex corner.*

A chord is *semiextremal* if it passes through a reflex corner and shares a common endpoint with an extremal chord (See Fig. 21.)

Lemma 10. *Let* $\mathscr{C} = (C_1, C_2, \ldots, C_m)$ *be an optimal chain of chords for a series of butterflies* $(B_1, B_2, \ldots, B_m)$. *For each* i, $1 \le i \le m$, *either both* C_i *and* C_{i+1} *are extremal, or one of them is extremal and the other is semiextremal.*

Proof. By definition of a chain, C_i and C_{i+1} must intersect. We say two line segments *overlap* if their intersection has positive length. There are two cases:

(1) B_i and B_{i+1} "intersect fully" i.e., the forward tip of B_i does not overlap the backward tip of B_{i+1} (See Fig. 22). If we consider C_i to be fixed, then Lemma 9 implies that C_{i+1} must be extremal in $P \cap C_i^+$. But since B_i and B_{i+1} do not share a tip, we see that C_{i+1} must in fact be extremal in P. Similarly, C_i is extremal.

(2) The forward tip of B_i overlaps the backward tip of B_{i+1}. First we assume that the two overlapping tips are identical (i.e., $[a, b]$ in Fig. 23). Let c be the intersection of C_i and C_{i+1}. It is not hard to see that Lemma 9 implies that c lies on the boundary of $B_i \cap B_{i+1}$. Suppose c does not lie on the shared tip T ($T = [a, b]$). Again Lemma 9 implies that C_i and C_{i+1} must be extremal in their respective butterflies.

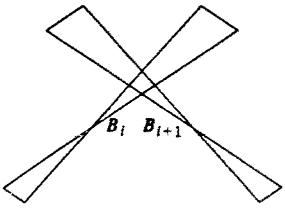

Fig. 22. Two butterflies that intersect fully.

Now consider the case where c is on the shared tip. The perimeter $L = L(c)$ determined by c is

$$\begin{aligned} L(c) &= |cc_i| + |cc_{i+1}| + |c_i a_i| + |c_{i+1} b_{i+1}| + S[b_{i+1}, a_i] \\ &= (|cc_i| + |c_i a_i| + |cb|) + (|cc_{i+1}| + |c_{i+1} b_{i+1}| + |ca|) \\ &\quad - |ab| + S[b_{i+1}, a_i] \\ &= L_i(c) + L_{i+1}(c) + \text{constant}, \end{aligned}$$

where $L_i(c)$ and $L_{i+1}(c)$ are the perimeter functions for the butterflies B_i and B_{i+1}, respectively. From the proof of Lemma 9 we know that both $L_i(c)$ and $L_{i+1}(c)$ have positive second derivatives. Hence $L(c)$ has positive second derivative. We conclude that L has no local maximum for c in the range $[a, b]$ and the maximal perimeter is determined by an endpoint of the shared tip. Hence both C_i and C_{i+1} are extremal. Finally if we drop the assumption that the overlapping tips of B_i and B_{i+1} are identical, the analysis can be modified in the obvious way to show that one of C_i and C_{i+1} is extremal and the other is semiextremal.

We conclude that in all cases at least one of C_i and C_{i+1} is extremal and the other is either extremal or semiextremal. □

Notation. For vertices v_i, v_j of P, let $P[i, j]$ denote the simple polygon formed from the vertices of P clockwisely between v_i and v_j. $P[i, j]$ is undefined if $[v_i, v_j]$ is not fully contained in P. For chords $C_i \in \Xi_i$, $C_j \in \Xi_j$, let $P[i, j, C_i, C_j]$ denote the connected component of $P[i, j] \cap C_i^+ \cap C_j^+$ that is bounded by $[v_i, v_j]$. Note that $P[i, j] \cap C_i^+ \cap C_j^+$ need not be a connected region.

Define $M_{i,j}^0(C_i, C_j)$ to be the perimeter of the polygon $P[i, j, C_i, C_j]$ (not counting the length of the edge $[v_j, v_i]$) if $P[i, j]$ is defined; and $-\infty$ otherwise.

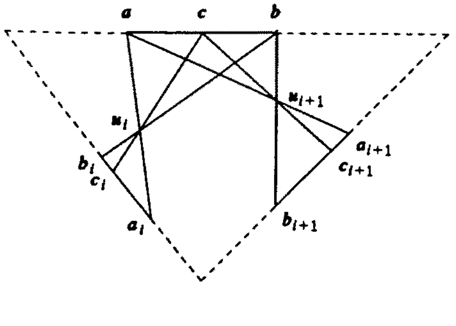

Fig. 23. Two butterflies with overlapping tips.

M^0 can be computed in $O(n^4)$ time. Let $M^*_{i,j}(C_i, C_j)$ denote the perimeter of the largest convex polygon contained in $P[i, j, C_i, C_j]$ (again the edge $[j, i]$ is not counted). For fixed i and j, we can regard each $M^*_{i,j}(C_i, C_j)$ as an $n \times n$ matrix.

Now we are ready to present an $O(n^6)$ time algorithm for finding the largest perimeter potato. Again, instead of computing the potato itself, we will pretend that we are only computing its perimeter.

Our problem is essentially reduced to computing M^*. To show this reduction, suppose that the potato in P is determined by a set of chords where at least two are extremal. So the maximum perimeter is given by

$$\max_{i,j \in U, C_i \in \Xi_i, C_j \in \Xi_j} \left\{ M^*_{i,j}(C_i, C_j) + M^*_{j,i}(C_j, C_i) \right\},$$

where U is the set of reflex vertices of P. This expression can be evaluated in $O(n^4)$ time, given M^*. The case where there is at most one extremal chord in the potato can be done in $O(n^5)$ time using a brute force method. Note that this case implies that the potato is determined by at most three chords. It remains to show how to determine M^*.

To compute M^*, we define two additional $n \times n$ matrixes M^1 and M^2. With i, j, C_i, C_j as before, and for $m = 1, 2$, we define $M^m_{i,j}(C_i, C_j)$ to be the perimeter of the largest convex polygon contained in $P[i, j, C_i, C_j]$ determined by at most m semiextremal chords in addition to C_i, C_j (but no other extremal chords). Again the length of the edge $[i, j]$ is not counted as part of the perimeter. We then have the following formula:

$$M^*_{i,j}(C_i, C_j) = \max_{i<k<j, C_k \in \Xi_k} \left\{ M^1_{i,j}(C_i, C_j), M^2_{i,j}(C_i, C_j), M^*_{i,k}(C_i, C_k) + M^*_{k,j}(C_k, C_j) \right\}.$$

This formula is justified by the fact that if the potato in $P[i, j, C_i, C_j]$ is determined by three or more chords, then at least one is extremal. As in the case of the area measure, we can recursively compute the entries of M^* in n stages. We initialize M^* to M^0, i.e., $M^*_{i,j}(C_i, C_j) = M^0_{i,j}(C_i, C_j)$, to start this recursion. It is easy to see that $O(n^6)$ time suffices for the overall computation, assuming the availability of M^0, M^1, and M^2.

Both M^1 and M^2 can be computed in $O(n^6)$ steps. We now describe briefly how M^2 is computed. For reflex vertices v_i, v_j and extremal chord $C_i \in \Xi_i$, $C_j \in \Xi_j$, let C and C' be the two extremal chord that determine $M^2_{i,j}(C_i, C_j)$. We

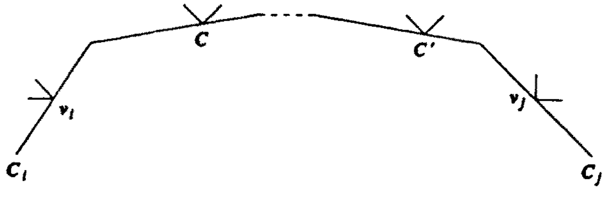

Fig. 24. Computing M^2.

observe that C and C' are disjoint and share endpoints respectively with C_i and C_j. (See Fig. 24.) It is clear that C and C' can be found in $O(n^2)$ steps by an exhausted search, provided that all the semiextremal chords are precomputed. Since there are $O(n^4)$ entries in M^2, it takes $O(n^6)$ steps to compute M^2. We can similarly compute M^1.

8. Conclusion

This paper gives the first polynomial-time solution to the potato-peeling problem. We have introduced the interesting geometric concept of balanced chains which holds the key to the problem. Computing these chains is the bottleneck to a faster algorithm for the problem. Our solution also exploits dynamic programming in several key steps.

References

1. A. Aggarwal, J. S. Chang, and C. K. Yap, Minimum area circumscribing polygons, to appear in a special issue of Visual Computer: International J. of Computer Graphics. Also, NYU-Courant Institute Robotics Lab. Report No. 42, May 1985.
2. M. Ben-Or, D. Kozen, and J. Reif, The complexity of elementary algebra and geometry, 16th STOC, 457–464, 1984.
3. J. E. Boyce, D. P. Dobkin, III, R. L. Drysdale, and L J. Guibas, Finding extremal polygons, 14th STOC, 282–289, 1982.
4. G. D. Chakerian, and L. H. Lange, Geometric extremum problems, Math. Mag. 44 (1971) 57–69.
5. J. S. Chang, and C. K. Yap, A polynomial solution to potato-peeling and other polygon inclusion and enclosure problems, 25th FOCS, 408–416, May 1984.
6. B. Chazelle, R. L. Drysdale, and D. T. Lee, Computing the Largest Empty Rectangle, Proc. of the Symposium on Theoretic Aspects of Comp. Sci., Paris, April, 1984.
7. A. DePano, and A. Aggarwal, Finding restricted K-envelopes for Convex Polygons, Proc. of the 22nd Allerton Conference on Comm. Control and Computing, 1984.
8. A. Depano, Approximations of Polygons and Polyhedra: Potentials for Research, Manuscript, 1984.
9. D. P. Dobkin and L. Snyder, On a general method for maximizing among certain geometric problems, 20th FOCS, 9–17, 1979.
10. D. Dori and M. Ben-Bassat, Circumscribing a convex polygon by a polygon of fewer sides with minimal area addition, Computer Vision, Graphics and Image Processing 24 (1983) 131–159.
11. J. E. Goodman, On the largest convex polygon contained in a non-convex n-gon, or How to peel a potato, Geometriae Dedicata 11 (1981) 99–106.
12. V. Klee and M. C. Laskowski, Finding the smallest triangles containing a given convex polygon, J. Algorithms, 457–464, to appear.
13. D. Kozen and C. K. Yap, Algebraic cell decomposition in NC, 26th FOCS, 1985.
14. A. M. Lopshits, Computation of Areas of Oriented Figures, DC Heath, 1963.
15. M. McKenna, J. O'Rourke, and S. Suri, Finding the Largest Rectangle in an Orthogonal Polygon, Tech. Report JHU/EECS-85/09, Dept. of Elec. Eng. and Comp. Sci., The Johns Hopkins University, 1985.
16. J. O'Rourke, A. Aggarwal, S. Maddila, and M. Baldwin, An Optimal Algorithm for Finding Minimal Enclosing Triangles, Technical Report JHU/EECS-84/08, Dept. of Elec. Eng. and Comp. Sci., The Johns Hopkins University, May 1984.
17. J. O'Rourke, Counterexamples to a Minimal Circumscription Algorithm, Manuscript, June 1984.

18. J. O'Rourke, Finding Minimal Enclosing Boxes, Technical Report, Dept. of Elec. Eng. and Comp. Sci., The Johns Hopkins University, 1984.
19. M. Sharir and A. Schorr, On shortest paths in polyhedral spaces, 16th STOC, 144–153, 1984.
20. G. T. Toussaint, Pattern Recognition and geometrical complexity, 5th International Conf. Pattern Recognition, 1324–1347 1979.
21. T. Woo, The Convex Skull Problem, Manuscript, 1983.
22. D. Wood and C. K. Yap, Computing a Convex Skull of an Orthogonal Polygon, Proc. of the Symposium on Computational Geometry, Baltimore, Maryland, 311–316, June, 1985.

Received June 28, 1985, and in revised form September 23, 1985.