BIT 30 (1990), 570–582

ON THE MINIMALITY OF POLYGON TRIANGULATION

CHIUYUAN CHEN and RUEI-CHUAN CHANG

Institute of Computer Science and Information Engineering, National Chiao Tung University Hsinchu, Taiwan, 30050 Republic of China

Institute of Computer and Information Science, National Chiao Tung University Hsinchu, Taiwan, 30050 Republic of China, and Institute of Information Science, Academia Sinica Nankang, Taipei, Taiwan, 11529 Republic of China

Abstract.

The problem of triangulating a polygon using the minimum number of triangles is treated. We show that the minimum number of triangles required to partition a simple n-gon is equal to $n + 2w - d - 2$, where w is the number of holes and d is the maximum number of independent degenerate triangles of the n-gon. We also propose an algorithm for constructing the minimum triangulation of a simple hole-free n-gon. The algorithm takes $O(n\log^2 n + DK^2)$ time, where D is the maximum number of vertices lying on the same line in the n-gon and K is the number of minimally degenerate triangles of the n-gon.

CR subjects classifications: G.1.6, I.1.2, I.3.5

1. Introduction.

Triangulating a simple polygon is one of the most important research problems in computational geometry. We shall first give a few basic definitions. A polygon is *simple* if there is no pair of nonconsecutive edges sharing a point. A *chord* of a polygon is a line segment connecting two vertices and lying in the polygon. (Here, a chord can touch the boundary of the polygon at one or several points, or have edges as subsets.) A *triangle* of a polygon is a region bounded by three chords, and a triangle is *degenerate* if its area is zero. A *triangulation* of a polygon partitions the interior of the polygon into triangles. We shall use $\Delta^*(P)$ to denote the minimum number of nondegenerate triangles required to partition a polygon P. For example, in Fig. 1a, $\overline{v_3v_7}$ is a chord and (v_3, v_1, v_7) is a degenerate triangle. Fig. 1b shows a minimum triangulation of Fig. 1a.

Many researchers [5, 6, 10] addressed the problem of partitioning a simple hole-free n-gon into $n - 2$ triangles. Recently, Tarjan et al. [10] proposed an $O(n\log\log n)$ time solution for this problem. Also, Fournier et al. [5] developed an

*) This research was partially supported by the National Science Council of the Republic of China under the Grant NSC77-0408-E009-16.

Received February 1988. Revised November 1988 and May 1990.

$O(n\log n)$ time algorithm to partition a simple n-gon which contains w polygon holes into $n + 2w - 2$ triangles.

While considering minimality, there is an interesting question: *How can we partition a simple polygon into the minimum number of nondegenerate triangles?* Lingas [7] has showed that partitioning a simple polygon which contains holes (polygon holes or point holes) into the minimum number of triangles is NP-hard. Nevertheless, Asano et al. [1] presented an approximation algorithm for this problem. Recently, Asano et al. [2] proposed an $O(n^3)$ time algorithm to partition a simple hole-free n-gon into the minimum number of triangles.

In this paper, we shall show that $\Delta^*(P) = n + 2w - d - 2$, where P is a simple n-gon with w polygon holes and d is the maximum number of independent degenerate triangles in P. Based on this minimality formula, we shall derive some interesting, tight bounds: if $w = 0$, then $\lceil n/3 \rceil \leq \Delta^*(P) \leq n - 2$; if $1 \leq w \leq \lceil n/6 \rceil - 1$, then $\lceil n/3 \rceil \leq \Delta^*(P) \leq n + 2w - 2$; and if $\lceil n/6 \rceil \leq w \leq \lfloor n/3 - 1 \rfloor$, then $2w + 1 \leq \Delta^*(P) \leq n + 2w - 2$. We shall also propose an algorithm for constructing the minimum triangulation of a simple hole-free n-gon. This algorithm takes at most $O(n\log^2 n + K^2)$ time if any four vertices of the n-gon are not collinear; the set of such polygons is not trivial, since the class of dependency graphs associated with minimally degenerate triangles (defined later) of these polygons is equal to the class of circle graphs. The algorithm takes $O(n\log^2 n + DK^2)$ time in general. Here D is the maximum number of vertices lying on the same line in the n-gon (for example, in Fig. 1a, D is five) and K is the number of minimally degenerate triangles of the n-gon (K is less than $2n - 1$). When K is $o(n)$, our algorithm improves from the previous best algorithm [2], which takes $O(n^3)$ time even when the given polygon contains only one degenerate triangle. Even if K is $O(n)$, our algorithm takes $O(n^3)$ time only when there are $O(n)$ vertices lying on the same line in the n-gon; usually, this degenerate case seldom appears.

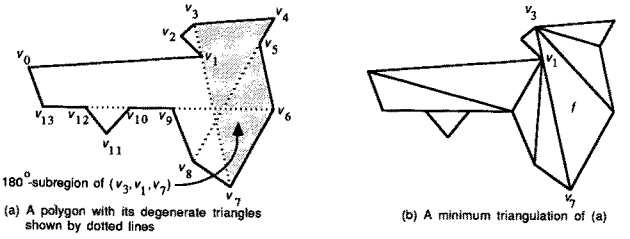

Fig. 1

The following conventions are used in this paper. Only polygon holes are considered. Polygons are assumed to be simple, with all vertices distinct and no three consecutive vertices collinear. If a triangle is not explicitly specified as degenerate, it is nondegenerate.

This paper is organized as follows: section 2 describes the minimality formula and derives bounds for $\Delta^*(P)$; section 3 proposes a minimum triangulation algorithm for hole-free polygons; concluding remarks are given in section 4.

2. The minimality formula and bounds for $\Delta^*(P)$.

First, a few basic notations are necessary. Let P be an n-gon with w holes. If (v_a, v_b, v_c) is a degenerate triangle of P, then v_b is the middle vertex of the three; (v_a, v_b, v_c) and (v_c, v_b, v_a) denote the same triangle. If T is a triangulation of P, $E(T)$ denotes the number of edges of T and $\Delta_T(P)$ denotes the number of nondegenerate triangles of T. T partitions the plane into connected regions; the closures of these nondegenerate regions are called *faces* of T. Let $F(T)$ denote the set of faces of T: call the faces which are in P *interior faces* and the others *exterior faces*. Let $\phi(T)$ denote the number of faces of T: exactly $w + 1$ faces are exterior. The *degree* $d(f)$ of a face f is the number of edges with which f is incident.

LEMMA 1: [1,8] *Let P be an n-gon with w holes. If P contains no degenerate triangles, then $\Delta^*(P) = n + 2w - 2$.*

A degenerate triangle is *minimally* degenerate if it contains no other degenerate triangles. For example, in Fig. 1a, (v_{10}, v_9, v_6) is a minimally degenerate triangle, but (v_{12}, v_9, v_6) is not. We shall concern ourselves with minimally degenerate triangles only.

The chords of a minimally degenerate triangle (v_a, v_b, v_c) divide the polygon into one or more subregions and split the interior angle v_b into subangles. Only one subangle is 180°, and one subregion contains this subangle. Call this subregion the 180°-*subregion* of (v_a, v_b, v_c) and the others *non*-180°-*subregions* of (v_a, v_b, v_c). See Fig. 1a.

A minimally degenerate triangle (v_a, v_b, v_c) of a polygon *exists* in a triangulation T of the polygon if (v_a, v_b, v_c) lies on the boundary of an interior face (a triangle) of T. For example, (v_3, v_1, v_7) exists in the triangulation in Fig. 1b and f is such a face.

Two minimally degenerate triangles of a polygon are *dependent* if they can not exist simultaneously in a triangulation. More precisely, two minimally degenerate triangles (v_a, v_b, v_c) and (v'_a, v'_b, v'_c) of a polygon are *dependent* if and only if these two triangles: (i) intersect; or (ii) (v'_a, v'_b, v'_c) touches (v_a, v_b, v_c) at v_b and (v'_a, v'_b, v'_c) lies in the 180°-subregion of (v_a, v_b, v_c), or (v_a, v_b, v_c) touches (v'_a, v'_b, v'_c) at v'_b and (v_a, v_b, v_c) lies in the 180°-subregion of (v'_a, v'_b, v'_c). For example, in Fig. 1a, (v_3, v_1, v_7) and (v_4, v_5, v_8) are dependent, but (v_3, v_1, v_7) and (v_{12}, v_{10}, v_9) are not.

THEOREM 1: *Two minimally degenerate triangles (v_a, v_b, v_c) and (v'_a, v'_b, v'_c) of a polygon are mutually independent if and only if these two triangles:*

(i) *do not touch or intersect at all;*

(ii) *touch at extreme vertices (share an extreme vertex);*

(iii) *overlap (share a side); or*

(iv) *(v'_a, v'_b, v'_c) touches (v_a, v_b, v_c) at v_b and (v'_a, v'_b, v'_c) lies in a non-180°-subregion of (v_a, v_b, v_c), or (v_a, v_b, v_c) touches (v'_a, v'_b, v'_c) at v'_b and (v_a, v_b, v_c) lies in a non-180°-subregion of (v'_a, v'_b, v'_c).*

PROOF. Immediate.

LEMMA 2: *Let P be an n-gon with w holes. If i mutually independent minimally degenerate triangles exist in a triangulation T of P, then $\Delta_T(P) \leq n + 2w - i - 2$.*

PROOF. Let m be the number of interior faces whose boundaries these i degenerate triangles lie on. Summation on degrees of the m faces is at least $3m + i$, and that of exterior faces is n. We have $n - E(T) + \phi(T) = 2$ and $\sum_{f \in F(T)} d(f) = 2E(T) \geq (3m + i) + 3(\phi(T) - (w + 1) - m) + n$. Thus $\Delta_T(P) \leq n + 2w - i - 2$. ■

A subset S of the vertex set of a graph is called an *independent set* if there is no edge of the graph joining any two vertices of S. A *maximum independent set* of a graph is an independent set with the maximum cardinality. A *dependency graph* associated with minimally degenerate triangles of a polygon is defined as follows: the vertex set of the graph is a set of all minimally degenerate triangles of the polygon, and two vertices are joined by an edge if and only if they are dependent. Let $G(P)$ denote the dependency graph of a polygon P.

THEOREM 2: *Let P be an n-gon with w-holes. Then $\Delta^*(P) = n + 2w - d - 2$, where d is the cardinality of a maximum independent set of $G(P)$.*

PROOF. It follows from Lemma 2 that $\Delta^*(P) \leq n + 2w - d - 2$. If $\Delta^*(P) < n + 2w - d - 2$, an independent set I of $G(P)$ can be constructed such that $|I| > d$; this contradicts the assumption that d is maximum. Therefore $\Delta^*(P) = n + 2w - d - 2$. ■

The following theorem derives tight bounds for $\Delta^*(P)$.

THEOREM 3: *Let P be an n-gon with w holes, and d be the cardinality of a maximum independent set of $G(P)$. The following bounds are tight:*
If $w = 0$, then $0 \leq d \leq n - \lceil n/3 \rceil - 2$ and $\lceil n/3 \rceil \leq \Delta^*(P) \leq n - 2$.
If $1 \leq w \leq \lceil n/6 \rceil - 1$, then $0 \leq d \leq n + 2w - \lceil n/3 \rceil - 2$ and $\lceil n/3 \rceil \leq \Delta^*(P) \leq n + 2w - 2$.
If $\lceil n/6 \rceil \leq w \leq \lfloor n/3 - 1 \rfloor$, then $0 \leq d \leq n - 3$ and $2w + 1 \leq \Delta^*(P) \leq n + 2w - 2$.

PROOF. The proof is simple and is left to the reader.

COROLLARY 4: [1] *Let P be an n-gon with w holes, and let $\Delta(P)$ be $n + 2w - 2$. If P is hole -free, then $\Delta(P)/\Delta^*(P) < 3$. If P contains holes, then $\Delta(P)/\Delta^*(P) < 4$.*

This corollary also shows that, in a minimum triangulation, up to $n/3$ triangles are needed for a simple hole-free n-gon with many degenerate triangles.

3. A minimum triangulation algorithm for hole-free polygons.

The basic approach of the algorithm is to partition a polygon P with a set having the maximum number of mutually independent minimally degenerate triangles; then, partition each derived subpolygon P_i into $n_i - 2$ triangles (where n_i is the number of vertices of P_i).

Algorithm MINITRI.

Input: A hole-free n-gon P with its vertices $v_0, v_1, \ldots, v_{n-1}$ listed in clockwise order.

Output: A minimum triangulation of P.

Step 1: Find all minimally degenerate triangles of P.

Step 2: Construct a circular representation $C(P)$ of P by mapping the set of minimally degenerate triangles of P onto a set of chords of a circle.

Step 3: Apply Algorithm *MISMDT* (described later) to $C(P)$ to find a set I having the maximum number of mutually independent minimally degenerate triangles.

Step 4: With I, partition P into subpolygons $P_0, P_1, \ldots, P_{p-1}$.

Step 5: Partition each P_i into $n_i - 2$ triangles, where n_i is the number of vertices of P_i.

Two procedures of Chazelle and Guibas will be applied in step 1. Chazelle et al. [4] proposed a procedure (called *RAYSHOOTING*) that preprocesses an m-gon in $O(m\log m)$ time. Given a pair (v, u) where v is a vertex and u is a direction, the first intersection (if any) of the boundary of the m-gon with the ray emanating from v in direction u can be decided in $O(\log m)$ extra time. Let e denote an edge of an m-gon and e_i denote the maximal segment of e that is visible from the ith vertex of the m-gon. (Two points of a polygon are *visible* from each other if the line segment joining them does not intersect with the exterior of the polygon.) Chazelle et al. [4] also proposed an $O(m\log m)$ time algorithm (called *VSEGMENTS*) to derive $e_1, e_2, \ldots, e_m$.

Actually, there are three types of minimally degenerate triangles. A minimally degenerate triangle of P is of *type* (i) if none of its sides is an edge of P. It is of *type* (ii) if one of its sides is an edge of P and none of its vertices join an edge of P collinear to that side. It is of *type* (iii) if one of its sides is an edge of P and one of its vertices joins an edge of P collinear to that side. For example, in Fig. 1a, (v_3, v_1, v_7), (v_4, v_5, v_8) and (v_{12}, v_{10}, v_9) are of types (i), (ii)and (iii) respectively.

Type (ii) and (iii) minimally degenerate triangles are found by Procedure *RAYSHOOTING*. Initially, preprocess P. Then, for each concave vertex v_i of P, do the following tests:

(a) use $(v_i, \overrightarrow{v_{i-1}v_i})$ to find the intersection point. If the intersection point is a vertex (say v_j) of P, then (v_{i-1}, v_i, v_j) is a degenerate triangle;

(b) use $(v_i, \overrightarrow{v_{i+1}v_i})$ to find if there is a degenerate triangle, as in (a).

Type (i) minimally degenerate triangles are found by the following divide-and-conquer technique. Chazelle [3] showed that with $O(n\log n)$ preprocessing time, a chord c can be found in $O(n)$ time such that c partitions P into two

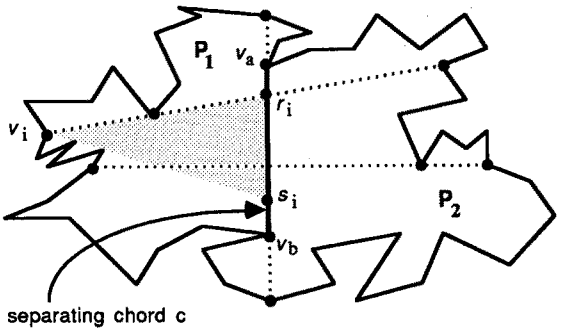

Fig. 2 Finding type (i) minimally degenerate triangles by the divide-and-conquer technique

subpolygons P_1 and P_2, where $|P_1| \leq 2n/3, |P_2| \leq 2n/3$, and $|P_1| + |P_2| = n + 2$. (See Fig. 2.) Suppose type (i) minimally degenerate triangles in P_1 and in P_2 are found. P_1 and P_2 must be merged along the separating chord c. Let $c = \overline{v_a v_b}$. There are two cases for type (i) minimally degenerate triangles:

(a) two vertices of the triangle are v_a and v_b, and the other is in P_1 or P_2;

(b) two vertices of the triangle belong to P_1 (or P_2), and the other belongs to P_2 (or P_1).

Type (i) minimally degenerate triangles for case (a) are found by Procedure *RAYSHOOTING*, using $(v_b, \overrightarrow{v_a v_b})$ and $(v_a, \overrightarrow{v_b v_a})$ as input. Type (i) minimally degenerate triangles for case (b) are found as follows. Initially, apply Procedure *VSEGMENTS* to P_1 and c (or P_2 and c); this yields the maximal segments $c_1, c_2, \ldots, c_{|P_1|}$ (or $c_1, \ldots, c_{|P_2|}$) that are visible from vertices in P_1 (or P_2). Then, preprocess P by Procedure *RAYSHOOTING*; for each vertex v_i whose maximal visible segment c_i is not empty, use $(r_i, \overrightarrow{v_i r_i})$ and $(s_i, \overrightarrow{v_i s_i})$ to find if there are degenerate triangles (where r_i and s_i are the endpoints of c_i).

In step 2, the boundary of P can be considered a circle and a minimally degenerate triangle (v_i, v_j, v_k) can be represented by the chord $\overline{v_i v_k}$. The circular representation of

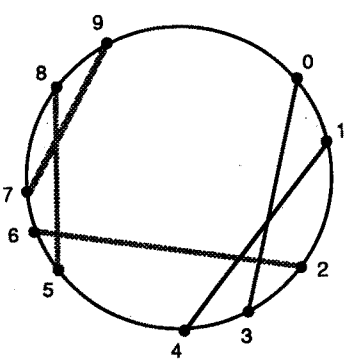

Fig. 3 Circular representation $C(P)$ of Fig. 1(a)

P (denoted by $C(P)$) can be constructed as shown in Fig. 3. (An endpoint with no chord incident upon it will be removed from the circle.) In Fig. 3, endpoints of the chords are numbered from zero to nine: corresponding to the vertices v_3, v_4, v_6, v_7, v_8, v_9, v_{10}, v_{10}, v_{12} and v_{13} respectively.

Properties of $C(P)$ are discussed here. Let (v_a, v_b, v_c) and (v'_a, v'_b, v'_c) be two minimally degenerate triangles of P; $\overline{v_a v_c}$ and $\overline{v'_a v'_c}$ will be their corresponding chords in $C(P)$. There are five different relationships between (v_a, v_b, v_c) and (v'_a, v'_b, v'_c):

(i) if they satisfy case (i) in Theorem 1, then $\overline{v_a v_c}$ and $\overline{v'_a v'_c}$ do not touch or intersect;

(ii) if they satisfy case (ii) in Theorem 1, then $\overline{v_a v_c}$ and $\overline{v'_a v'_c}$ touch at endpoints. In this case, we can move $\overline{v_a v_c}$ or $\overline{v'_a v'_c}$ slightly so that they will not touch any more;

(iii) if they satisfy case (iii) in Theorem 1, then two subcases exist. Without loss of generality, assume that $v_b = v'_a$ and $v_c = v'_b$. If the 180°-subregions of them are:
  (a) on the opposite sides of $\overline{v_a v'_c}$, then $\overline{v_a v_c}$ and $\overline{v'_a v'_c}$ do not touch or intersect;
  (b) on the same side of $\overline{v_a v'_c}$ (for example, (v_{13}, v_{12}, v_{10}) and (v_{12}, v_{10}, v_9) in Fig. 1a), then $\overline{v_a v_c}$ and $\overline{v'_a v'_c}$ intersect. This is what we do not desire;

(iv) if they satisfy case (iv) in Theorem 1, then $\overline{v_a v_c}$ and $\overline{v'_a v'_c}$ do not touch or intersect;

(v) if they are dependent, then $\overline{v_a v_c}$ and $\overline{v'_a v'_c}$ intersect.

$C(P)$ is similar to the circle-graph model for a circle graph: A graph G is a *circle graph* if there exists a set of chords C on a circle and a one-to-one correspondence between vertices of G and chords of C such that two vertices are joined by an edge if and only if their corresponding chords intersect. Set C is called a *circle-graph model* for G.

THEOREM 5: *The class of dependency graphs of hole-free polygons such that any four vertices of each polygon are not collinear is equal to the class of circle graphs. The class of dependency graphs of hole-free polygons properly contains the class of circle graphs.*

PROOF. Let P be a hole-free polygon such that any four vertices of P are not collinear. Obviously, P contains no minimally degenerate triangles which satisfy relationship (iii)(b). From previous discussion, $C(P)$ is a circle-graph model for $G(P)$. Thus $G(P)$ is a circle graph. Conversely, let G be a circle graph and C be

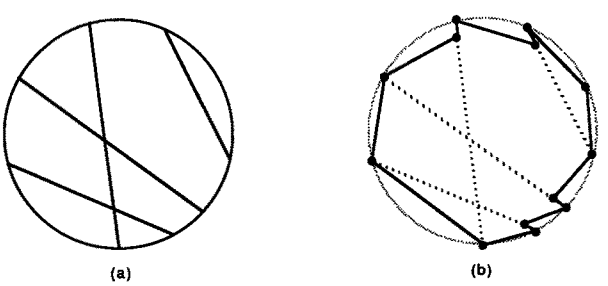

Fig. 4 Constructing a polygon whose dependency graph is equal to the given circle graph

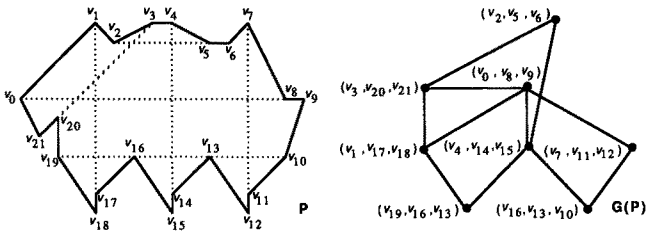

Fig. 5 A polygon whose dependency graph is not a circle graph

a circle-graph model for G as shown in Fig. 4a. A polygon P can be constructed from C as shown in Fig. 4b; each chord of C is transferred to a type (ii) minimally degenerate triangle in P. P is hole-free and any four vertices of P are not collinear. Graph G is exactly $G(P)$. Hence the former statement is proved.

It is not difficult to prove that the dependency graph of the polygon shown in Fig. 5 is not a circle graph. Therefore the latter statement is true. ■

Supowit [9] proposed an $O(K^2)$ time algorithm to find a maximum independent set of a circle graph with K vertices. His algorithm can be applied to $C(P)$ to find a maximum independent set of $G(P)$, if it can be extended to overcome the problem in relationship (iii)(b). The following algorithm, *MISMDT*, is such an extension.

Input of the algorithm is $C(P)$ and output is set I. Assume that $|C(P)| = K$. Endpoints of the chords in $C(P)$ are numbered from 0 to $2K-1$, clockwise around the circle. Denote the chord with endpoints a and b by $\overline{ab}$ (or $\overline{ba}$).

A graph $G = (V, E)$ associated with $C(P)$ is defined as follows: $V = \{v_{ab}: a < b$ and $\overline{ab} \in C(P)\}$, and $E = \{(v_{ab}, v_{a'b'}) \in V^2$: corresponding degenerate triangles of $\overline{ab}$ and $\overline{a'b'}$ are dependent$\}$.

Let G_{ij} be the subgraph of G which is induced by the set of vertices $\{v_{ab} \in V: i \le a, b \le j,\ 0 \le i, j \le 2K-1\}$, and $MIS(i,j)$ a maximum independent set of G_{ij}.

The function max is defined such that $\max\{S_0, S_1, \ldots, S_l\} = S_m$ if $|S_i| < |S_m|$ for $i = 0, 1, \ldots, m-1$ and $|S_m| \ge |S_i|$ for $i = m+1, m+2, \ldots, l$.

To solve the problem in relationship (iii)(b), two extra notations are introduced: *3b-chain* and $A_{ij}(S)$. A series of chords $(\overline{a_1b_1}, \overline{a_2b_2}, \ldots, \overline{a_kb_k})$ is called a *3b-chain* if either $k = 1$ or $\overline{a_ib_i}$ and $\overline{a_{i+1}b_{i+1}}$ satisfy relationship (iii)(b) for $1 \le i \le k-1$. $A_{ij}(S)$ denotes an independent set of G_{ij} formed by a 3*b*-chain S: assume that endpoints of the chords in S are $a_0, a_1, \ldots, a_{2|S|-1}$ (listed in clockwise order from i to j); then $A_{ij}(S) = \{v_{ab}: \overline{ab}$ is a chord of $S\} \cup MIS(i, a_0 - 1) \cup MIS(a_0 + 1, a_1 - 1) \cup \ldots \cup MIS(a_{2|S|-1} + 1, j)$. For example, in Fig. 3, $(\overline{2,6}, \overline{5,8})$ is a 3*b*-chain and $A_{0,8}((\overline{2,6}, \overline{5,8}))$ is $\{v_{2,6}, v_{5,8}\} \cup MIS(0,1) \cup MIS(3,4) \cup MIS(7,7)$.

At the very beginning, $MIS(i,j)$ is assigned to an empty set for each $i \ge j$. Later, $MIS(i,j)$ is computed for each pair i,j using dynamic programming, and it is assumed that $MIS(i_1,j)$ is computed before $MIS(i_2,j)$ when $i_1 < i_2$. $MIS(i,j)$ is computed as follows. Consider $\overline{kj} \in C(P)$. If k is not in range $[i, j-1]$, then the result is

$MIS(i,j) = MIS(i,j-1)$. If k is in range $[i,j-1]$, two possible cases exist:

(i) if $v_{kj} \notin MIS(i,j)$, then the result is $MIS(i,j) = MIS(i,j-1)$;

(ii) if $v_{kj} \in MIS(i,j)$, then $MIS(i,j)$ contains no vertices v_{ab} such that $a \in [i,k-1]$ and $b \in [k+1,j-1]$ unless $\overline{ab}$ and $\overline{kj}$ satisfy (iii)(b). Therefore the result is $MIS(i,j) = \max\{A_{ij}(S)$: S is a $3b$-chain that contains chord $\overline{kj}\}$.

Finally, the set I is derived from $MIS(0,2K-1)$. The following is a formal description of the algorithm above:

Algorithm MISMDT.

```
FOR j = 0 TO 2K − 1 DO
  BEGIN
  Let k be the number such that kj ∈ C(P)
  FOR i = 0 TO j − 1 DO
    IF i ≤ k ≤ j − 1 THEN
      BEGIN
      Compute A_ij(S) for all S, where S is a 3b-chain that contains chord kj
      MIS(i,j) = max {MIS(i,j − 1), max {A_ij(S): for all S}}
      END
    ELSE MIS(i,j) = MIS(i,j − 1)
  END
Compute I from MIS(0,2K − 1).
```

THEOREM 6: *Algorithm MISMDT finds a set having the maximum number of mutually independent minimally degenerate triangles of a polygon.*

PROOF. It is sufficient to prove that $MIS(i,j)$ is a maximum independent set of G_{ij} for $0 \le i \le j \le 2K-1$. $MIS(i,j)$ is an independent set. All that needs to be proved is that if M is a maximum independent set of G_{ij}, then $|MIS(i,j)| = |M|$. Since $|MIS(i,j)| \le |M|$ holds, only $|MIS(i,j)| \ge |M|$ must be proved. The proof is carried by induction on j.

When $j = 0$ then $MIS(0,0) = \phi$ is a maximum independent set of $G_{0,0}$.

Assume $j \ge 1$ and also that $MIS(i,l)$ is a maximum independent set of G_{il} for $0 \le i \le l < j$. When $i = j$, $MIS(j,j) = \phi$ is a maximum independent set of G_{jj}. When $i < j$, let M be a maximum independent set of G_{ij} and consider $\overline{kj} \in C(P)$.

If k is not in the range $[i,j-1]$, then $G_{ij} = G_{i,j-1}$ and $|M| = |MIS(i,j-1)|$. Since $MIS(i,j) = MIS(i,j-1)$ when k is not in the range $[i,j-1]$, $|MIS(i,j)| \ge |MIS(i,j-1)| = |M|$ holds.

If k is in the range $[i,j-1]$, there are two possible cases for M:

(i) if $v_{kj} \notin M$ exists, then $|M| = |MIS(i,j-1)|$. Recall that $MIS(i,j) = \max\{MIS(i,j-1), \max\{A_{ij}(S)$: S is a $3b$-chain that contains chord $\overline{kj}\}\}$. Thus $|MIS(i,j)| \ge |MIS(i,j-1)| = |M|$ holds;

(ii) if $v_{kj} \in M$ exists, it will be shown that there exists a $3b$-chain S' such that S' contains the chord $\overline{kj}$ and $|A_{ij}(S')|$ equals $|M|$; consequently $|MIS(i,j)| = |\max\{MIS(i,j-1), \max\{A_{ij}(S)$: S is a $3b$-chain that contains the chord $\overline{kj}\}\}| \ge$

$|A_{ij}(S')| = |M|$ holds. S' is constructed from M as follows: initially, let $S' = (\overline{kj})$; then S' will be extended as long as there exists a vertex $v_{ab} \in M$ such that $\overline{ab}$ satisfies relationship (iii)(b) with a chord already existent in S'; add $\overline{ab}$ to S'. For example, in Fig. 3, $M = \{v_{2,6}, v_{5,8}\}$ is a maximum independent set of $G_{0,8}$; initially $S' = (\overline{5,8})$, and finally $S' = (\overline{2,6}, \overline{5,8})$. Assume that endpoints of the chords in S' are $a_0, a_1, \ldots a_{2|S|-2}, j$ (listed in clockwise order from i to j). Then $M = \{v_{ab}: \overline{ab}$ is a chord of $S'\} \cup$ a maximum independent set of $G_{i,a_0-1} \cup$ a maximum independent set of $G_{a_0+1,a_1-1} \cup \ldots \cup$ a maximum independent set of $G_{a_{2|S|-2}+1,j-1}$. By definition, $A_{ij}(S') = \{v_{ab}: \overline{ab}$ is a chord of $S'\} \cup MIS(i, a_0 - 1) \cup MIS(a_0 + 1, a_1 - 1) \cup \ldots \cup MIS(a_{2|S|-2} + 1, j - 1)$. Since $MIS(i, l)$ is a maximum independent set of G_{il} for $0 \le i \le l < j$, $|A_{ij}(S')|$ equals $|M|$. Thus $|MIS(i,j)| \ge |M|$. ■

The validity of Algorithm *MINITRI* is implied from Algorithm *MISMDT*.

THEOREM 7: *Algorithm MISMDT takes $O(DK^2)$ time and Algorithm MINITRI takes $O(n\log^2 n + DK^2)$ time, where D is the maximum number of vertices lying on the same line in P and K is the number of minimally degenerate triangles of P.*

PROOF. First, Algorithm *MISMDT* is analyzed. As in [9], the elements of $MIS(i,j)$ and of $A_{ij}(S)$ are not explicitly stored when they are computed in the body of the interior loop of this algorithm. Instead, the elements of $MIS(i,j)$ and of $A_{ij}(S)$ are represented by pointers marked with "+" (or "−") to sets contained (or not contained) in them. (For example, if $MIS(i,j) = MIS(i,j-1)$, then $MIS(i,j)$ is represented by a pointer marked with "+" to $MIS(i,j-1)$.) Note that the sizes of $MIS(i,j)$ and of $A_{ij}(S)$ must be explicitly stored in the body of the interior loop. Elements of $MIS(0,2K-1)$ are explicitly enumerated in the last step.

For convenience, introduce the notation S_{ij} and let $\overline{kj} \in C(P)$. If k is not in the range $[i,j-1]$, then S_{ij} is set to the empty chain (); otherwise S_{ij} is set to the longest $3b$-chain which contains the chord $\overline{kj}$ and endpoints of the chords in S_{ij} are in range $[i,j]$. For example, in Fig. 3, $S_{0,8} = (\overline{2,6}, \overline{5,8})$.

Let $\overline{kj} \in C(P)$. While computing $MIS(i,j)$, these cases are considered (see Fig. 6):

(i) if $S_{ij} = ()$, then $MIS(i,j) = MIS(i,j-1)$. Thus computing $MIS(i,j)$ takes $O(1)$ time;

(ii) if $S_{ij} = (\overline{kj})$, then $MIS(i,j) = \max\{MIS(i,j-1), \max\{A_{ij}((\overline{kj}))\}\}$. Further $A_{ij}((\overline{kj})) = \{v_{kj}\} \cup MIS(i,k-1) \cup MIS(k+1,j-1)$. Computing $A_{ij}((\overline{kj}))$ takes $O(1)$ time; thus computing $MIS(i,j)$ takes $O(1)$ time;

(iii) if $S_{ij} = (\overline{a_1b_1}, \overline{kj})$ or $(\overline{kj}, \overline{a_1b_1})$, then $MIS(i,j) = \max\{MIS(i,j-1), \max\{A_{ij}((\overline{kj})), A_{ij}(S_{ij})\}\}$. $A_{ij}(S_{ij}) = \{v_{a_1b_1}, v_{kj}\} \cup MIS(i, a_1 - 1) \cup MIS(a_1 + 1, k - 1) \cup MIS(k+1, b_1 - 1) \cup MIS(b_1 + 1, j - 1)$. Computing each $A_{ij}(S)$ takes $O(1)$ time. Thus computing $MIS(i,j)$ takes $O(1)$ time;

(iv) if $S_{ij} = (\overline{a_sb_s}, \ldots, \overline{a_1b_1}, \overline{kj})$ and $s \ge 2$, then $MIS(i,j) = \max\{MIS(i,j-1), \max\{A_{ij}((\overline{kj})), A_{ij}((\overline{a_1b_1}, \overline{kj})), \ldots, A_{ij}((\overline{a_sb_s}, \ldots, \overline{a_1b_1}, \overline{kj}))\}\}$. Compute $A_{ij}((\overline{a_ub_u}, \ldots, \overline{a_1b_1}, \overline{kj}))$ for $u = 2, 3, \ldots, s$ as follows:

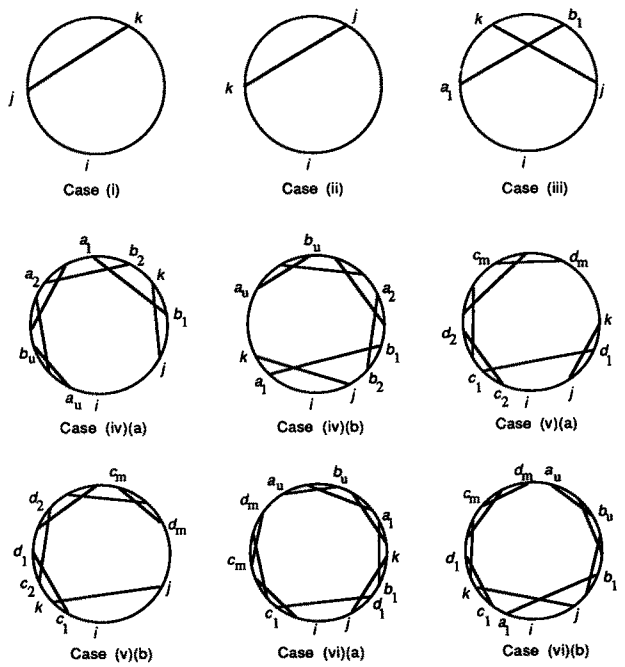

Fig. 6 Cases in computing MIS(i,j)

(a) if $k > a_u$, then $A_{ij}((\overline{a_u b_u}, \ldots, \overline{a_1 b_1}, \overline{kj})) = A_{ib_1}((\overline{a_u b_u}, \ldots, \overline{a_1 b_1})) - MIS(b_2 + 1, b_1 - 1) \cup \{v_{kj}\} \cup MIS(b_2 + 1, k - 1) \cup MIS(k + 1, b_1 - 1) \cup MIS(b_1 + 1, j - 1)$;

(b) if $k < a_u$, then $A_{ij}((\overline{a_u b_u}, \ldots, \overline{a_1 b_1}, \overline{kj})) = A_{ib_2}((\overline{a_u b_u}, \ldots, \overline{a_1 b_1})) - MIS(a_1 + 1, a_u - 1) \cup \{v_{kj}\} \cup MIS(a_1 + 1, k - 1) \cup MIS(k + 1, a_u - 1) \cup MIS(b_2 + 1, j - 1)$.

Computing each $A_{ij}((S))$ takes $O(1)$ time. Thus computing $MIS(i,j)$ takes $O(s + 1)$ time;

(v) if $S_{ij} = (\overline{kj}, \overline{c_1 d_1}, \ldots, \overline{c_t d_t})$ and $t \geq 2$, then $MIS(i,j) = \max\{MIS(i,j-1), \max\{A_{ij}((\overline{kj})), A_{ij}((\overline{kj}, \overline{c_1 d_1})), \ldots, A_{ij}((\overline{kj}, \overline{c_1 d_1}, \ldots, \overline{c_t d_t}))\}\}$. Compute $A_{ij}((\overline{kj}, \overline{c_1 d_1}, \ldots, \overline{c_m d_m}))$ for $m = 2, 3, \ldots, t$ as follows:

(a) if $k > c_m$, derive $A_{ij}((\overline{kj}, \overline{c_1 d_1}, \ldots, \overline{c_m d_m}))$ from $A_{id_1}((\overline{c_1 d_1}, \ldots, \overline{c_m d_m}))$;

(b) if $k < c_m$, derive $A_{ij}((\overline{kj}, \overline{c_1 d_1}, \ldots, \overline{c_m d_m}))$ from $A_{id_m}((\overline{c_1 d_1}, \ldots, \overline{c_m d_m}))$.

Computing each $A_{ij}((S))$ takes $O(1)$ time. Thus computing $MIS(i,j)$ takes $O(t + 1)$ time;

(vi) if $S_{ij} = (\overline{a_s b_s}, \ldots, \overline{a_1 b_1}, \overline{kj}, \overline{c_1 d_1}, \ldots, \overline{c_t d_t})$ and $s, t \geq 1$, then $MIS(i,j) = \max\{MIS(i,j-1), \max\{A_{ij}((\overline{kj})), A_{ij}((\overline{a_1 b_1}, \overline{kj})), \ldots, A_{ij}((\overline{a_s b_s}, \ldots, \overline{a_1 b_1}, \overline{kj})), A_{ij}((\overline{kj}, \overline{c_1 d_1})), \ldots, A_{ij}((\overline{kj}, \overline{c_1 d_1}, \ldots, \overline{c_t d_t})), A_{ij}((\overline{a_1 b_1}, \overline{kj}, \overline{c_1 d_1})), \ldots, A_{ij}((\overline{a_s b_s}, \ldots, \overline{a_1 b_1}, \overline{kj}, \overline{c_1 d_1}, \ldots, \overline{c_t d_t}))\}\}$. There are two cases: (a) $k > a_u$ and (b) $k < a_u$. In both cases, derive $A_{ij}((\overline{a_u b_u}, \ldots, \overline{a_1 b_1}, \overline{kj}, \overline{c_1 d_1}))$ from $A_{ij}((\overline{a_u b_u}, \ldots, \overline{a_1 b_1}, \overline{kj}))$ for $u = 1, 2, \ldots, s$; derive $A_{ij}((\overline{a_u b_u}, \ldots, \overline{a_1 b_1}, \overline{kj}, \overline{c_1 d_1}, \ldots, \overline{c_m d_m}))$ from $A_{ij}((\overline{a_u b_u}, \ldots,$

$\overline{a_1b_1}, \overline{kj}, \overline{c_1d_1}, \ldots, \overline{c_{m-1}d_{m-1}}$)) for $u = 1, 2, \ldots, s$ and $m = 2, 3, \ldots, t$. Computing each $A_{ij}(S)$ takes $O(1)$ time. Thus computing $MIS(i,j)$ takes $O((s+t+1)^2)$ time.

A $3b$-chain is *maximal* if it is not properly contained in any other $3b$-chain. For example, in Fig. 3, ($\overline{2,6}$, $\overline{5,8}$, $\overline{7,9}$) is a maximal $3b$-chain, but ($\overline{5,8}$, $\overline{7,9}$) is not.

It is not difficult to show that total time complexity contributed by chords in a maximal $3b$-chain S_M is $O(K|S_M|^2)$. Each chord $\overline{kj}$ belongs to exactly one maximal $3b$-chain. The sum of the lengths of all maximal $3b$-chains is K. Thus the body of the interior loop (with summation over S_M being a maximal $3b$-chain) takes $\sum O(K|S_M|^2) \leq \sum O(KD|S_M|) = O(DK^2)$ time. In the last step, the elements of $MIS(0,2K-1)$ are explicitly enumerated by traversing back from $MIS(0,2K-1)$ along the pointers; this takes $O(K)$ time. Therefore Algorithm *MISMDT* takes $O(DK^2)$ time.

Algorithm *MINITRI* is now analyzed. Step 1 takes a total $O(n\log^2 n)$ time. Finding type (ii) and type (iii) minimally degenerate triangles takes $O(n\log n)$ time. Let $T(n)$ denote time complexity of finding type (i) minimally degenerate triangles in an n-gon. Then $T(n) \leq T(|P_1|) + T(|P_2|) + O((|P_1| + |P_2|)\log(|P_1| + |P_2|))$ and $T(5) = 1$ follow. Thus we find $T(n) = O(n\log^2 n)$.

Steps 2, 3 and 4 of Algorithm *MINITRI* takes $O(n+K)$, $O(DK^2)$ and $O(K)$ time respectively. Step 5 takes $O(n\log n)$ time if the algorithm in [6] is used to triangulate each subpolygon P_i. Thus Algorithm *MINITRI* takes $O(n\log^2 n + DK^2)$ time. ■

It is obvious that if any four vertices of the given polygon are not collinear, Algorithm *MISMDT* takes $O(K^2)$ time and Algorithm *MINITRI* takes $O(n\log^2 n + K^2)$ time.

Space needed by Algorithm *MINITRI* is $O(n + DK^2)$: steps 1, 2, 4, and 5 need $O(n)$ space, and step 3 (Algorithm *MISMDT*) needs $O(DK^2)$ space. Space needed by [2] is $O(n^2)$.

The following lemma states a bound for K.

LEMMA 3: *An n-gon contains less than $2n-1$ minimally degenerate triangles.*

PROOF. Let t_1, t_2, t_3 be the number of type (i), (ii), (iii) minimally degenerate triangles of the n-gon respectively. This lemma then follows from $t_2 + t_3 < n$ and $t_1 < n$. ■

4. Concluding remarks.

The purpose of this paper is twofold: *Theoretically*, we have described the minimum triangulation problem in a definite way, and derived some interesting, tight bounds for the minimum number of triangles required to partition a polygon. *Practically*, we have proposed a minimum triangulation algorithm which runs faster than the previously best algorithm [2] when K is $o(n)$. Even if K is $O(n)$, our algorithm takes the same time as [2] only when there are $O(n)$ vertices lying on the same line in the polygon; usually, this degenerate case seldom appears. Also, when

the number of degenerate triangles is small, our algorithm takes only $O(n\log^2 n)$ time; so it performs well even in comparison with the standard triangulation (triangulating a polygon into $n-2$ triangles), which takes $O(n\log\log n)$ time [10]. Here n is the input size, and K is the number of minimally degenerate triangles of the polygon.

Intuitively, we can extend our result by proposing a strategy for the following problem: partitioning a rectilinear polygon into the minimum number of convex quadrilaterals, which is an important guard problem. Finally, what the minimality formula in this paper will be if point holes are allowed is still an open problem.

Acknowledgements.

The authors wish to express their gratitude to the anonymous referees for their useful comments that have greatly improved this paper.

REFERENCES

1. T. Asano, T. Asano, and Y. Ohsuga, *Partitioning a polygonal region into a minimum number of triangles*, The Transactions of the IECE of Japan, Vol. E 67, No. 4 (1984), pp. 232–233.
2. T. Asano, T. Asano, and R. Y. Pinter, *Polygon triangulation: efficiency and minimality*, Journal of Algorithms, Vol. 7, No. 2 (1986), pp. 221–231.
3. B. M. Chazelle, *A theorem for polygon cutting with applications*, Proceedings of the 23rd IEEE Annual Symposium on the Foundations of Computer Science (1983), pp. 339–349.
4. B. M. Chazelle and L. J. Guibas, *Visibility and intersection problems in plane geometry*, Discrete Computational Geometry, Vol. 4, No. 6 (1989), pp. 551–581.
5. A. Fournier and D. Y. Montuno, *Triangulating simple polygons and equivalent problems*, ACM Transactions on Graphics, Vol. 3, No. 2 (1984), pp. 153–174.
6. M. R. Garey, D. S. Johnson, F. P. Preparata, and R. E. Tarjan, *Triangulating a simple polygon*, Information Processing Letters, Vol. 7, No. 4 (1978), pp. 175–179.
7. A. Lingas, *The power of non-rectilinear holes*, Proceedings of the 9th Colloquium on Automata, Languages and Programming, Aarhus (1982), pp. 369–383.
8. J. O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, Inc., New York (1987).
9. K. J. Supowit, *Finding a maximum planar subset of a set of nets in a channel*, IEEE Transactions on Computer-Aided Design, Vol. 6, No. 1 (1987), pp. 93–94.
10. R. E. Tarjan, and C. J. Van Wyk, *An O(n loglog n)-time algorithm for triangulating a simple polygon*, SIAM Journal on Computing, Vol. 17, No. 1 (1988), pp. 143–178.