Approximation Algorithms for Covering Polygons with Squares and Similar Problems

Christos Levcopoulos[⋆] Joachim Gudmundsson[⋆⋆]

Department of Computer Science, Lund University, Box 118, S-221 00 Lund, Sweden

Abstract. We consider the problem of covering polygons, without any acute interior angles, using a preferably minimum number of squares. The squares must lie entirely within the polygon.

Let P be an arbitrary input polygon, with n vertices, coverable by squares. Let $\mu(P)$ denote the minimum number of squares required to cover P. In the first part of this paper we show an $O(n \log n + \mu(P))$ time algorithm which produces at most $11n + \mu(P)$ squares to cover P. In the hole-free case our algorithm runs in linear time and produces a cover which is within an $O(\alpha(n))$ approximation factor of the optimal, where $\alpha(n)$ is the extremely slowly growing inverse of Ackermann's function. In parallel our algorithm runs in $O(\log n)$ randomized time using $O(\max(\mu(P), n))$ processors. We also present an algorithm which guarantees a constant (14) approximation factor running in $O(n^2 + \mu(P))$ time. As a corollary we obtain the first polynomial-time, constant-factor approximation algorithm for "fat" rectangular coverings.

1 Introduction

One of the main topics of computational geometry is how to decompose polygonal objects into simpler polygons, such as triangles, squares, rectangles, convex polygons and star-shaped polygons [15]. We distinguish between two types of decomposition, *partitions* and *coverings*. A decomposition is called a partition if the object is decomposed into non-overlapping pieces. If the pieces are allowed to overlap, then we call the decomposition a covering.

In this paper we consider the problem of covering polygons with squares. Given a polygon P, the *square covering* problem asks for a minimum number of (possibly overlapping) squares whose union is P.

Decomposition problems have many applications both within and outside computer science [7, 8]. When solving other problems for geometric figures, a common method is to decompose the figure into simpler parts, solve the problem on each component using a specialized algorithm, and then successively combine the partial solutions to solve the problem for larger and larger parts of the polygon [15].

O'Rourke and Supowit [14] showed that the problems of covering polygons with a minimum number of convex polygons, star-shaped polygons or spiral polygons are all NP-hard, if the polygon contains holes.

The related *rectilinear square covering* problem, i.e. when the polygons, as well as the squares, have sides that are vertical or horizontal, has been treated in several papers [3, 17, 13]. Some of the previous work on the rectilinear square cover problem used a bit-map representation for the input polygon [2, 1, 13, 17]. A *bit-map* representation of a polygon is a boolean (zero-one) matrix, where one (1) represents a point inside the polygon and a zero (0) - a point outside it. When using this representation, complexity is measured in terms of the number of points in the matrix, denoted p. Note that $p > \Omega(n)$, and for most practical applications $p \gg n$, where n is the number of edges of P. Bar-Yehuda and Ben-Hanoch argues that the representation of the polygon should be made as segment representation, not only theoretically, but also for most practical applications [3].

Scott and Iyenger [17] present an algorithm to find, in $O(n \log n)$ time, the maximal squares in the polygon, and then divide the rectangular portions of the polygon to cover them minimally.

[⋆] E-mail: christos@dna.lth.se

[⋆⋆] E-mail: joakim@dna.lth.se

However, their algorithm does not yield a globally minimum cover. It was shown by Aupperle, Conn, Keil and O'Rourke [2] that the rectilinear case is NP-hard for polygons containing holes. In the case where the image is hole-free, they provide an $O(p^{1.5})$ algorithm.

Recently, Bar-Yehuda and Ben-Hanoch [3] presented a linear time algorithm for covering simple (hole-free) rectilinear polygons with squares. Morita [13] developed a parallel algorithm, which finds a *minimal* (not minimum) square cover for bit-maps which may contain holes. The sequential running time of this algorithm is $O(p)$. A square cover is called *minimal* if it has no smaller subset that forms a cover. A square cover is called *minimum* if there is no smaller set that forms a cover.

The *general rectangle cover* problem has also been treated in several papers [4, 5, 7, 10, 12]. One application for this problem is the fabrication of VLSI chips. According to [7], the most common method for fabricating VLSI chips is the optical method of the automatic blockflasher, which exposes rectangles of almost any orientation. In order to minimize the cost of the fabrication it is desirable to cover the polygonal area of each layer of the circuit with as few rectangles as possible. For more information about the VLSI fabrication process see [7].

In [10] an algorithm was presented which covers a polygon P with $O(n \log n + \mu'(P))$ rectangles in $O(n \log n + \mu'(P))$ time, where $\mu'(P)$ is the minimum number of rectangles needed to cover P. In [5] a different heuristic was presented, guaranteeing an $O(\log n)$ approximation factor in polynomial time ($\Omega(n^6)$), provided that the vertices of the polygon have polynomially bounded integer coordinates. Even if the rectangles have bounded aspect ratio, i.e. the ratio between the longest and the shortest side of the rectangles is bounded by a constant, then, prior to this paper, no polynomial-time approximation algorithm was known to guarantee an approximation factor better than $O(\log n)$.

Thus, in this paper we show the first constant-factor approximation algorithm for rectangles with bounded aspect ratio running in time $O(n^2 + \mu^*(P))$, where $\mu^*(P)$ denotes the minimum number of (bounded ratio) rectangles needed to cover an arbitrary polygon P. We present two algorithms which cover an arbitrary polygon P, without any acute interior angles, with squares. In Section 2, we present an algorithm which covers an arbitrary polygon P with at most $11n + \mu(P)$ squares in time $O(n \log n + \mu(P))$. If the polygon is hole-free, then the algorithm produces a covering which is within an $O(\alpha(n))$ approximation factor in optimal time $O(n + \mu(P))$. The algorithm works by partitioning the polygon into smaller pieces, which are then covered independently. In Section 3 we present a square covering algorithm which guarantees a constant approximation factor running in $O(n^2 + \mu(P))$ time.

We obtain the same results for covering with fat rectangles, but the approximation factors increase by some constant, depending on how fat the rectangles are (Section 3.2).

2 A linear-time algorithm

In this section we will show a covering algorithm, H, that covers an arbitrary polygon P with at most $11n + \mu(P)$ squares in linear time, provided that the medial axis of the polygon is given. The algorithm works by partitioning the polygon P into cells which are then covered pairwise by squares lying within P. The first step of our algorithm is achieved by drawing the Voronoi diagram, $V(P)$, also called the polygonal skeleton or medial axis, of the polygon. In [9] the notion of the Voronoi diagram is generalized to include open line segments as well as points. Given an arbitrary polygon P, the part of the generalized Voronoi diagram lying within P is constructed, $V(P)$, which partitions P into $n + w$ faces (w is the number of concave vertices of P), where each segment and each concave vertex induces a face of the Voronoi diagram, Fig. 11. Every point lying in a cell induced by d, where d is either a segment or a concave vertex, lies at least as close to d as to any other point of the boundary of P. The diagram can be computed in $O(n \log n)$ time [9] for an arbitrary polygon and in linear time for a simple polygon [18]. The following fact has been shown:

Fact 1 *The number of edges in $V(P)$ is at most $3n$, where n is the number of vertices in the polygon P.*

Proof. Let $D(V(P))=(V_D,E_D)$ denote a dual of $V(P)=(V_V,E_V)$. Since $V(P))$ is a connected planar graph, its dual graph $D(V(P))$ is the graph obtained by the following procedure: (i) place a point on every edge and every concave vertex of P; these points are the vertices of $D(V(P))$; (ii) for each edge e of $V(P))$, draw a line joining the vertices in $D(V(P))$ corresponding to the two edges/vertices inducing e. These lines corresponds to the edges of $D(V(P))$, Fig. 1. Let n be the number of vertices of P and let w be the number of concave vertices of P.

For simplicity we start by calculating the number of Voronoi edges in $V(P)$, where P is a hole-free polygon. In this case the dual does not have any parallel edges. It is easily seen that $D(V(P))$ is a planar graph with all its $n+w$ vertices on its boundary, hence the maximal number of edges is $2(n+w)-3$. Note that if two edges in P are connected by a concave vertex, in P, then they are not connected in the dual of $V(P)$ to the vertex induced by the concave vertex, Fig. 1. Thus, we could for every vertex in $D(V(P))$ induced by a concave vertex of P add two edges to $D(V(P))$ and still preserve planarity of $D(V(P))$. Hence, $|E_V|=|E_D|\leq 2(n+w)-3-2w=2n-3$.

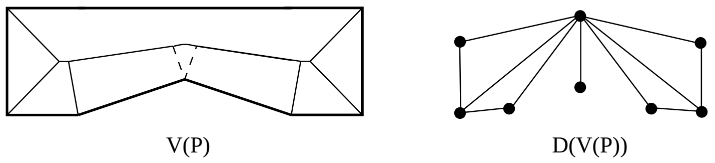

Fig. 1. A dual graph $D(V(P))$ of the generalized Voronoi diagram $V(P)$ of a polygon P.

In the general case there may be parallel edges in $D(V(P))$. (In this case P must contain at least one hole.) Let e_1' and e_2' be any two parallel edges in E_D, let t_1 and t_2 be their endpoints, and let $Q=(V_Q,E_Q)$ be the subgraph of $D(V(P))$ bounded by e_1' and e_2', including t_1 and t_2. Finally, let $S=(V_S,E_S)$, such that $V_S=\{V_Q-\{t_1,t_2\}\}$, that is E_S doesn't contain any edges with endpoints in t_1 or t_2.

Suppose Q includes r, $r\geq 0$, parallel edges, $e_1,\dots,e_r$, connecting t_1 and t_2 ordered from left to right. Then, S consists of $r+1$ disjoint planar subgraphs, $S_0,\dots,S_r$, without any parallel edges (since there can't be any holes within holes in P). Let $Q_i=(V_{Q_i},E_{Q_i})$, such that $V_{Q_i}=V_{S_i}\cup\{t_1,t_2\}$.

Let S_i be an arbitrary subgraph of S. Since S_i doesn't include any parallel edges, the above result $|E_{S_i}|\leq 2|V_{S_i}|-3$ holds for S_i. The vertices in V_{S_i} may also be connected to t_1 and t_2. The maximal number of edges connecting t_1 or t_2 to V_{S_i} is at most $|V_{S_i}|+2$. Hence, $|E_{Q_i}|=(2|V_{S_i}|-3)+(|V_{S_i}|+2)+2=3|V_{S_i}|+1=3|V_{Q_i}|-5$. If we put all the sets together we have that $|E_{Q_0}\cup\ldots\cup E_{Q_r}|=\sum_{i=0}^{r}(3|V_{Q_i}|-5)-r=3|V_Q|-1$, since $\sum_{i=0}^{r}|V_{Q_i}|=|V_Q|+2r$. Let w' denote the number of vertices in V_D-V_S induced by concave vertices of P. Note that the number of vertices on the boundary of $D(V(P))$ is $(n+w')-(|V_Q|-2)$. Now, we have that the number of edges in $D(V(P))$ is at most:

$$2((n+w')-(|V_Q|-2))-2w'-3+(3|V_Q|-1)=2n+|V_Q|<3n.$$

It is easily seen that this upper bound holds for an arbitrary number of holes in P.

Now, for every Voronoi face f, induced by an edge of P, let g be the segment of P which induces the face. The first step of our algorithm is achieved by partitioning each face into cells with only three or four vertices each, by drawing from every vertex of the face a segment connecting the vertex with its perpendicular projection on g, Fig. 2.

Definition 1. *We call the edge of each cell which is at the boundary of P the* base *of the cell. We call the segments which are perpendicular to its base the* sides *of the cell, the edge opposite to its base is called the* top *of the cell, and finally, two cells sharing a Voronoi edge, e, are called a* proper pair of cells.

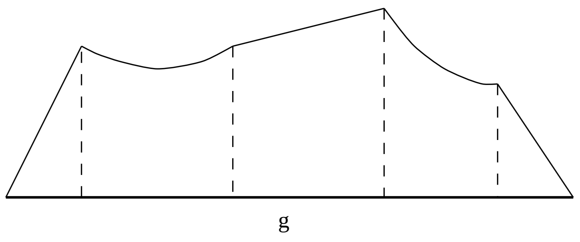

Fig. 2. A Voronoi face partitioned into cells.

Since there are at most $3n$ edges in $V(P)$, it remains to show that every proper pair of cells can be covered by a constant number of squares or that the number of squares produced to cover a (proper) pair of cells is optimal, to obtain the promised result. (The idea of processing Voronoi cells separately was also proposed in [10].) We will in the following section show that every pair of cells may be covered by at most 4 squares, except for a total of at most $\mu(P)$ additional squares, thus producing at most $4{\cdot}3n{+}\mu(P){=}12n{+}\mu(P)$ squares. In section 3.2 we will improve this result to $11n{+}\mu(P)$.

2.1 Covering the cells

The rest of the algorithm consists of independently processing each (proper) pair of cells and outputting a set of squares which covers them. Each pair is processed in time linear with respect to the number of squares produced in this step. The total number of edges is less than $3n$ and therefore the overall time performance of this step is $O(n{+}s)$, where s is the number of squares produced in this step. We will now describe how each pair of cells is processed.

There are four major cases: (1) the Voronoi edge between the two cells is a paraboloid, (2) when the two opposite cells are trapezoids, (3) when the two opposite cells are triangles, and (4) when the two opposite cells are both induced by concave vertices.

In the continuation of this text we will denote the cells in a proper pair C_l and C_t. If any of the cells are induced by an edge of P then we may assume without loss of generality that C_l is induced by an edge of P, C_l lies below C_t and that the base of C_l is horizontal and at the bottom.

Case 1. The Voronoi edge between C_l and C_t is a paraboloid, i.e. C_l is induced by an edge and C_t is induced by a concave vertex.

In this case the cells are covered by at most three squares. Let v be the concave vertex and let A, B, C and D be the corners of C_l in a counter-clockwise order, where A is the left endpoint of C_l's base, Fig. 3. There are two subcases:

Subcase 1.1 The concave vertex doesn't have a perpendicular projection on C_l's base, Fig. 3. In this case the cells are covered by at most three squares. We may assume w.l.o.g. that v lies to the right of BC. Note that $|AB|{\leq}|AD|$, since v lies to the right of BC, and that the interior angle $\angle DvC$, denoted α, is less than 180 degrees. We have two cases:

– $\alpha{\leq}90°$

Place a square s_1 on AB, such that s_1's base coincides with AB, and a square s_2 with center at D and bottom corner at A. C_l and the part of C_t to the left of the extension of BC are now entirely covered by these two squares. According to the definition of the medial axis we know that there exist two empty circles c_1 and c_2 within P, such that c_1 (c_2) has center at D (C) and passes through both A and v (B and v), Fig. 3a. Now it is easily seen that s_2 lies entirely inside c_1 and that s_1's perimeter, except its base, lies entirely inside c_1 and c_2. Hence, s_1 and s_2 lie within P.

It remains to cover the uncovered region of C_t, i.e. the uncovered region to the right of the extension of BC. Let q be a straight line splitting the angle α into two equal angles, and let E

be the intersection between q and the Voronoi edge between C and D, Fig. 3b. Place a square s_3 with center at E and right corner at v. Since $\alpha \leq 90°$, we have that s_3 covers the remaining uncovered region of C_t and, since we know that there exists an empty circle c_3 with center at E which passes through v, we have that s_3 lies entirely within P, Fig. 3b.

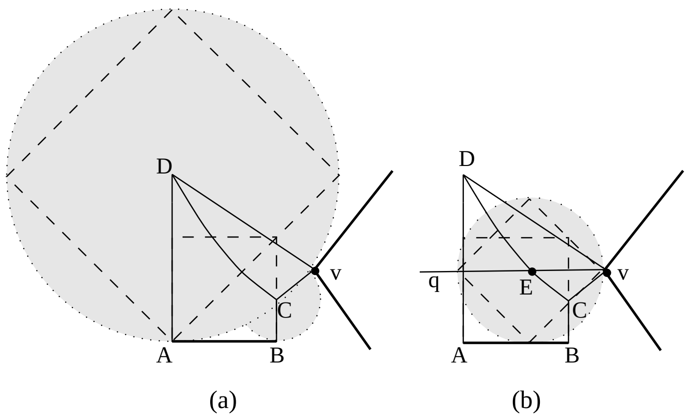

Fig. 3. Case 1.1.

- $\alpha > 90°$
  Place a square s_1 on AB, such that s_1's base coincides with AB, and place a square s_2 with center at D and bottom corner at A. C_l and the part of C_t to the left of the extension of BC are now entirely covered by these two squares, as described in the previous paragraph, and s_1 and s_2 lie within P. We know that the vertical distance between AB and v is between $|AD|$ and $|BC|$, since $\alpha > 90°$, thus place a square s_3 with right vertical side on v, bottom horizontal side on C and side length equal to $(|AB| - |BC|)$, Fig. 4. Since we have that the length of the sides of s_3 is $|AB| - |BC| > |vC|$ we have that s_3 covers the remaining uncovered region of C_t. C_l and C_t are now entirely covered by these three squares. Let c_1 and c_2 be the circles as described above. Note that c_1 includes the rectangular region to the left and above v, and to the right and below D, and that c_1 and c_2 includes the rectangular region to the left and below v, above C and to the right of AD. Thus it is easily seen that s_3 also lies entirely within P.

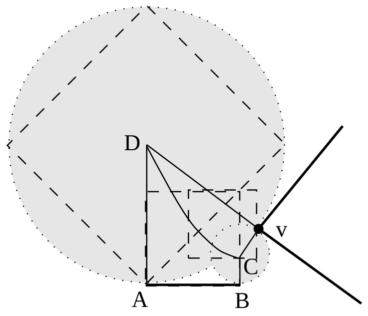

Fig. 4. Case 1.1.

Subcase 1.2 The concave vertex has a perpendicular projection on the C_l's base, Fig. 5.
In this case the two cells are covered by three squares. Let h be the distance between the concave

vertex and AB and let v's perpendicular projection on AB be denoted v_p. We may assume w.l.o.g. that $|AD| \geq |BC|$. We have two cases:

- $|AB| \leq h$
  Place a square s_1 on AB, such that s_1's base coincides with AB, and place an equally large square s_2 vertically above s_1, such that its horizontal top includes v. Note that s_1 and s_2 have the same x-coordinates. Finally, place a square s_3 with upper corner at v and lower diagonal corner at v_p. According to the definition of the medial axis there exist two empty circles c_1 and c_2 within P, such that c_1 (c_2) has center at D (C) and includes both A and v (B and v), Fig. 5a. Since c_1's and c_2's highest point both are vertically above s_2's upper left, respectively right, corner, it is easily seen that the three squares constructed in this case lie entirely within P and cover the two cells, Fig. 5a.

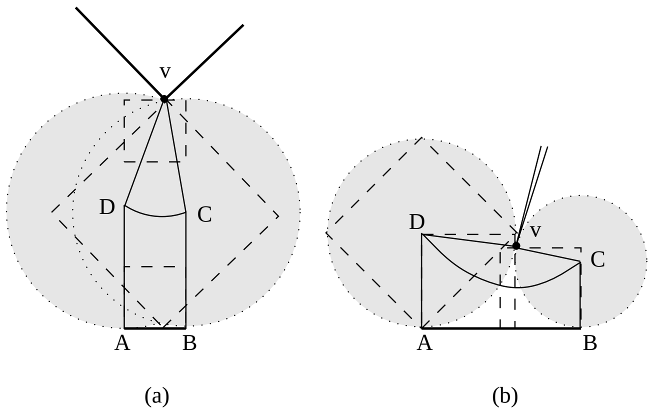

Fig. 5. Case 1.2.

- $h < |AB|$
  Note that $|AB| \leq 2 \cdot |AD|$, since $|AD| = |vD| \geq \frac{1}{2}|AB|$. According to the definition of the medial axis there exist two empty circles c_1 and c_2 within P, such that c_1 (c_2) has center at D (C) and includes A and v (B and v). Place a maximal square s_1 on AB, such that s_1's lower left corner coincides with A and s_1's right side includes v. Place a maximal square s_2 on AB, such that s_2's lower right corner coincides with B and s_2's upper side includes v. Note that s_1 and s_2 cover AB and that the perimeter of the region covered by s_1 and s_2, except the part on AB, lie entirely inside the two circles, thus the two squares lie entirely within P. Place a square s_3 with center at D and lower corner at A. Now we have that the three squares cover the two cells and that they lie entirely within P, Fig. 5b.

Case 2. C_l and C_t are both trapezoids.

In this case the cells are covered by at most four squares, unless they form a funnel (to be defined below). Let A, B, C and D be the corners of C_l in a counter-clockwise order, where A is the left endpoint of C_l's base, and let A' and B' be the left, respectively right, endpoint of C_t's base and thus, the symmetrical images of A and B, Fig. 6. We may assume w.l.o.g. that $|BC| \geq |AD|$. Let α be the interior angle $\angle ADC$. There are two subcases:

Subcase 2.1 $\alpha \geq 135°$ or $|AB| < |AD|$.
Place a square s_1 on AB, such that a side of s_1 coincides with AB, and a square s_2 with center at C and bottom corner at B. These two squares lie entirely within P and cover C_l, Fig. 6a. Cover C_t in the same way as C_l.

Subcase 2.2 $\alpha < 135°$ and $|AD| \leq |AB|$. ("funnel")
Place a square s_1 with base on AB, such that its lower left corner coincides with A and its upper

left corner touches $A'B'$. Now, place a square s_2 with base on AB, such that its lower left corner coincides with s_1's lower right corner and its upper left corner touches $A'B'$. Proceed as described until AB is entirely included by the produced squares. These squares cover C_l. If these squares also cover C_t then we are finished, otherwise cover C_t in the same way as C_l, Fig. 6b. Since every edge has to be entirely covered in any optimal partition, it is easily seen that the total number of squares produced in this substep is not greater than $4k+\mu(P)$, where k is the number of pairs of cells belonging to this subcase.

Note that the number of squares needed to cover a funnel can be calculated easily in constant time.

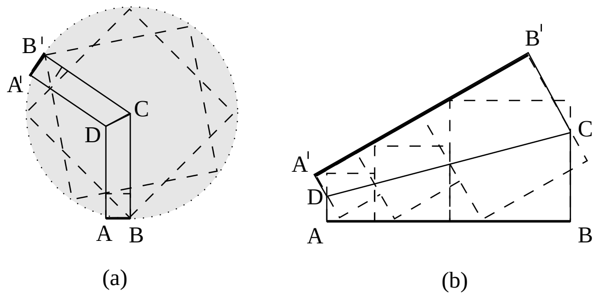

Fig. 6. Case 2.1 and 2.2.

Case 3. The two opposite cells are both triangles.

In this case the cells are covered by at most three squares. Let A, B and C be the corners of C_l in a counter-clockwise order, where A is the left endpoint of C_l's base, and let B' be the left endpoint of C_t's base, Fig. 7. Note that $45°\leq\angle BAC<90°$. Place two squares s_1 and s_2, such that s_1 coincides with AB and s_2 coincides with AB'. Finally, place a square, s_3, with bottom corner on A and with upper diagonal corner on C. It is easily seen that s_3 covers the remaining uncovered area, since the interior angle $\angle(BCA)$ is less than $90°$ and the crossing between s_3's sides and BC and $B'C$ are never above s_1's upper right corner respectively s_2's upper left corner, Fig 7. These three squares lie entirely within P, since we know that there exists a circle within P with center in C and perimeter on B and B', according to the definition of the medial axis.

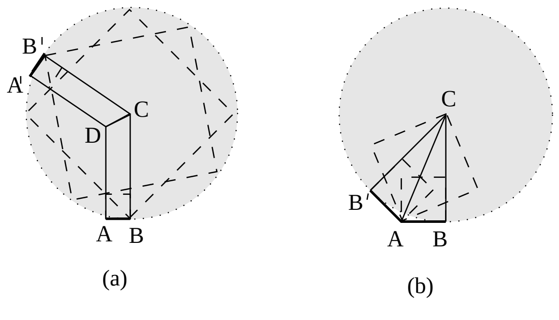

Fig. 7. Case 3.

Case 4. The two opposite cells are both induced by a concave vertex.

In this case the cells are covered by at most three squares. We may assume w.l.o.g. that the Voronoi edge, e, separating the two cells lies horizontal. Let v_t and v_l denote the concave vertex above respectively below e, let h denote the distance between v_l and v_t, and let c be the midpoint of e. We distinguish between two subcases:

Subcase 4.1 $|e| \leq h$
Place two squares s_1 and s_2 with sides of length e, such that s_1's base includes v_l, s_2's top includes v_t and s_1's and s_2's left sides have the same x-coordinate as e's left endpoint. If these squares cover the two cells then we are finished, Fig. 8a, otherwise, place a square s_3 with upper corner at v_t and lower diagonal corner at v_l. It is easily seen that these squares lie entirely within the polygon and cover the two cells, Fig. 8b.

Subcase 4.2 $|e| > h$
Place a square, s, such that its lower horizontal side includes v_l, its upper horizontal side includes v_t, and with center at c. Now place a square s_1, such that s_1's center is at e's left endpoint and s_1's right corner coincides with c and, finally, place a square s_2, such that s_2's center is at e's right endpoint and s_2's left corner coincides with c, Fig. 8b. According to the definition of the medial axis there exist two empty circles, c_1 and c_2, within P, such that c_1 (c_2) has center in e's left (right) endpoint and includes both v_l and v_t. Since the radius of c_1 and c_2 is greater than half the length of s_1's and s_2's diagonal, Fig. 8c, these three squares lie entirely within the polygon and cover the two cells.

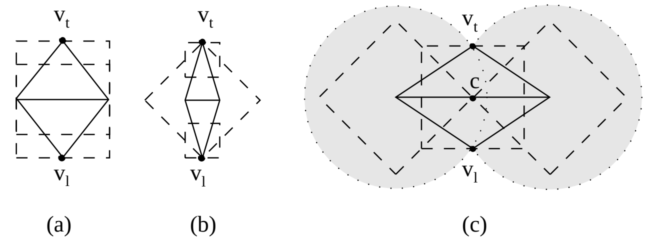

Fig. 8. Case 4.

From Fact 1 we have that P is partitioned into at most $3n$ pair of Voronoi faces, and in section 3.1 we have shown that every pair can be covered by at most four squares or, subcase 3.2, by an optimal number of squares. Hence, H will produce at most $12n+\mu(P)$ squares to cover P in $O(n \log n+\mu(P))$ time.

2.2 Improvements

As we can see from the above section there is just one subcase, namely 2.1, where we may need four squares to cover a pair of cells. We may improve our upper bound by noting that the maximum number of trapezoidal pair of cells is at most $2n$.

Observation 1 *For an arbitrary polygon P, with Voronoi diagram $V(P)$, it holds that the number of trapezoidal pair of cells, in $V(P)$, is at most $2n$.*

Proof. We know that for every convex vertex there exists a pair of triangles, and for every convex vertex there is a paraboloid Voronoi edge. Thus, there are at least n proper pair of cells that are not trapezoids. According to Fact 1 there are at most $3n$ edges, hence there are at most $2n$ trapezoidal pair of cells in $V(P)$.

By using this result it is not difficult to show that the algorithm will produce less than $4\times 2n+3\times n+\mu(P)$ squares to cover P, thus

$$H(P)\leq 11n+\mu(P).$$

2.3 A Tight Lower Bound on Optimal Coverings for Hole-Free Polygons

Theorem 1. *For all integers $n\geq 4$ and for every hole-free polygon P with n vertices it holds that $\mu(P)=\Omega(\frac{n}{\alpha(n)})$.*

Proof. Let $G=\{l_1,\ldots,l_{4\mu(P)}\}$ be a collection of m segments in the plane. Let P be the polygon bounded by the outer face in the partition of the plane induced by G. Y_G is a piecewise linear function, whose graph consists of subsegments of the segments l_i, $1\leq i\leq m$. Hart and Sharir [6] have shown that Y_G consists of at most $O(m\cdot\alpha(m))$ segments.

The number of edges of P is bounded by $O(m\cdot\alpha(m))$ [6]. Since the total number of segments bounding the squares in any minimum square covering of P is $\leq 4\cdot\mu(P)$ (four segments per square), we obtain, as above, that $n=O(\mu(P)\cdot\alpha(\mu(P)))$. Thus, $\mu(P)=\Omega(\frac{n}{\alpha(\mu(P))})$ and, hence $\mu(P)=\Omega(\frac{n}{\alpha(n)})$.

Theorem 2. *The lower bound shown in Theorem 1 cannot be generally improved, i.e. for each n there is a polygon P with $n\geq 10$ vertices such that $\mu(P)=O(\frac{n}{\alpha(n)})$.*

Proof. According to Wiernik [19] there exists a construction of a set M, of m straight-line segments, such that the lower envelope Y of M consists of $\Omega(m\cdot\alpha(m))$ subsegments.

Construct k non-vertical connected edges, $E=\{e_1,\ldots,e_k\}$, such that the edges in E are identical to a lower envelope produced by the m segments in [19]. Let e_i be the i:th left-most edge of E and let d denote the largest vertical distance between two points in E.

Expand E along the x-axis, by multiplying all x-coordinates such that the minimum difference along the x-axis between two incident vertices in E is $10d$.

Let e_0 and e_{k+1} be two horizontal edges of length $10d$, such that e_0 is appended to the left side of e_1 and e_{k+1} is appended to the right side of e_k. Let E' be the set of edges in E plus e_0 and e_{k+1}. Now, construct an upside-down histogram P, with one horizontal upper long-side, s_t, two vertical edges s_l and s_r, of length between $2|s_t|-d$ and $2|s_t|$ such that s_l connects s_t's left endpoint with e_0's left endpoint and s_r connects s_t's right endpoint with e_{k+1}'s right endpoint. The length of s_l and s_r guarantees that a square which includes an edge in E can't include any point on s_t. (See Fig. 9.)

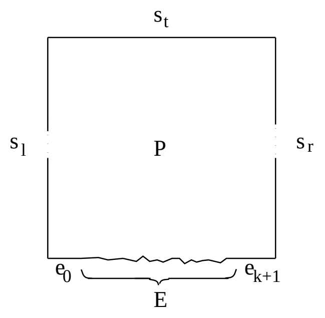

Fig. 9. It is possible to cover P with $O(n/\alpha(n))$ rectangles.

We prove that it is possible to cover P with $O(\frac{n}{\alpha(n)})$ squares. Let r_1 be a maximal square whose upper side coincides with s_t, and let r_2 be a maximal square with horizontal lower side of length $|s_t|$ that includes at least one point in E. The region of P covered by r_1 and r_2 is denoted

P_T and the uncovered area of P is denoted P_E. Let r_{e_0} and $r_{e_{k+1}}$ be two maximal squares, within P, such that r_{e_0} includes e_0, and $r_{e_{k+1}}$ includes e_{k+1}. Now, let R_E be a minimum set of maximal squares, within P, such that every edge of E coincides with at least one square in R_E. Let $R_{E'}$ be the set of squares in R_E plus r_{s_l} and r_{s_r}.

We will now show that the squares in R_E cover the remaining uncovered part of P_E, i.e. $(P_E-(r_{e_0}\cap P_E)-(r_{e_{k+1}}\cap P_E))$. Note the following facts:

1. the largest slope of an edge in E' is $9°$,
2. the shortest distance between two non-incident edges in E' is $10d$, and
3. the shortest distance between s_t and an edge in E is $2|s_t|-2d$.

It is easily seen that every point in P_E with a perpendicular projection on any edge of E' is covered by the squares in $R_{E'}$, Fig. 10a. Note that every remaining uncovered region lies closer than d to a concave vertex of E'. Thus it remains to show that every point in P_E, closer than d to a concave vertex of E', is covered by the squares in $R_{E'}$.

Let v be a concave vertex connecting e_i and e_{i+1}, $0\leq i\leq k$, in E', and let r_{e_i} and $r_{e_{i+1}}$ be the two squares in $R_{E'}$ whose bases include e_i, respectively e_{i+1}, Fig. 10b. According to the above facts, we have that r_{e_i} and $r_{e_{i+1}}$ extend at least $10d$ to the right, respectively to the left of v, thus we have that r_{e_i} and $r_{e_{i+1}}$ cover the part of P_E closer than d to v. Hence $(R_E\cup r_t\cup r_{s_l}\cup r_{s_k})$ covers P_E and $\mu(P)=O(\#R_E)$.

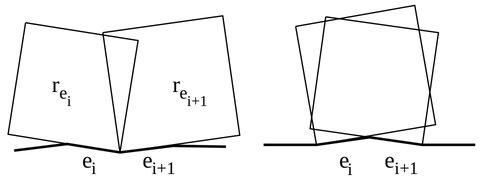

Fig. 10.

According to [19] we have that the number of segments in E is $\Omega(\#R_E\cdot\alpha(\#R_E))=k(=n-5)$. Let $Y(R_E)$ be the lower envelope of R_E. Since $\#Y(R_E)=\#E$, according to the definition of E, we have that the number of squares in R_E is at most $O(\frac{\#E}{\alpha(\#E)})=O(\frac{n}{\alpha(n)})$. Hence $\mu(P)=O(\frac{n}{\alpha(n)})$.

Summarizing the main result of this section we obtain the following theorem.

Theorem 3. *For any polygon P with n vertices, Algorithm H produces a square covering of P with at most $11n+\mu(P)$ squares in $O(n\log n+\mu(P))$ time. In the hole-free case the algorithm runs in linear time and produces a cover which is within an $O(\alpha(n))$ approximation factor of the optimal. The number of squares needed to cover P by using the algorithm H can be calculated in $O(n\log n)$ time ($O(n)$ time in the hole-free case).*

2.4 Parallelization of the Algorithm

The algorithm presented above partitions the polygon P into $O(n)$ Voronoi cells, which are then processed independently. Every cell can be processed in constant or optimal time, thus proposing a natural parallel algorithm.

Rajasekaran *et al.* [16] showed in 1994 that the medial axis of a polygon P can be constructed in parallel by $O(n)$ processors in $O(\log n)$ time in the CRCW PRAM model by using random sampling. Hence, our algorithm can easily be parallelized by first constructing the medial axis of P and then covering the cells independently.

All pair of cells, except funnels, can be processed in constant time by one processor. A funnel f can be processed in constant time using $O(\mu(f))$ processors, where $\mu(f)$ is the minimum number

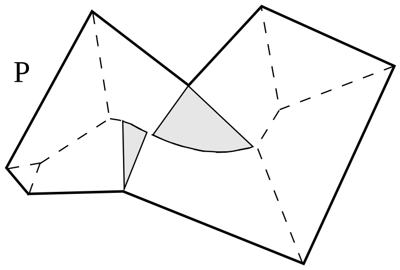

Fig. 11. Medial axis (dashed). Faces induced by concave vertices are shadowed.

of squares needed to cover f. We obtain a parallel algorithm which runs in $O(\log n)$ randomized time using $O(\max[\mu(P), n])$ processors in the CRCW PRAM model. Since we may calculate the number of squares needed to cover a funnel of P in constant time, we have that the number of squares needed by H to cover P can be calculated in parallel by $O(n)$ processors using $O(\log n)$ time.

3 Achieving a Constant Approximation Factor

In this section we will present a simple algorithm, H, which covers an arbitrary polygon P with at most $24 \times \mu(P)$ squares in quadratic time. We also show the modifications that are needed to improve the approximation factor from 24 to 14 (Section 3.1). Let $H(P)$ denote the number of squares produced by H. Let $H_1, \ldots, H_5$ denote the five different steps of H, and let $H_2(P), H_5(P)$ denote the number of squares produced in steps 2 and 5.

Step 1: Generate a list of Voronoi faces (H_1)
The first step of our algorithm is to construct a list of all faces of the generalized Voronoi diagram of P. Each face is represented by a list of its edges in clockwise order. In [9] the notion of the Voronoi diagram, also called the polygonal skeleton or medial axis, is generalized to include open line segments as well as points. Given an arbitrary polygon P, the part of the generalized Voronoi diagram lying within P is constructed which partitions P into $n+w$ faces, Fig. 11, where each segment and each concave vertex induces a face of the Voronoi diagram (w is the number of concave vertices of P). Every point lying in a face induced by d, where d is either a segment or a concave vertex, lies at least as close to d as to any other point of the boundary of P. By using Kirkpatrick's algorithm the time complexity of H_1 is $O(n \log n)$.
Step 2: Detect and cover all funnels (H_2)
In this step we need some previous results and definitions.

Definition 2. *[10] Let P be any hole-free polygon. A* funnel cell *of P is a trapezoidal piece of a Voronoi face in P having the following properties. Let A, B, C and D be the vertices of the trapezoid in counter-clockwise order, such that AD is parallel to and shorter than BC, and the interior angle (A, D, C) is greater than or equal to 90 and less than 135 degrees, Fig. 12. The segment AB lies on an edge of P, say e, and the segment CD is a Voronoi edge bounding the Voronoi face induced by e in P [9].*

By this, and by the definition of generalized Voronoi diagrams, it follows that the mirror image of a funnel cell with respect to the Voronoi edge bounding the cell is also a funnel cell. Such a pair of funnel cells with a Voronoi edge separating them is called a funnel*, Fig. 12. Let A' and B' be the symmetric image of A respectively B.*

Fact 2 *Let F denote the set of all funnels in an arbitrary polygon P.*
a) It holds that $\mu(P)=O(n+\sum_{t \in F} \mu(t))$, where $\mu(t)$ is the minimum number of squares needed to

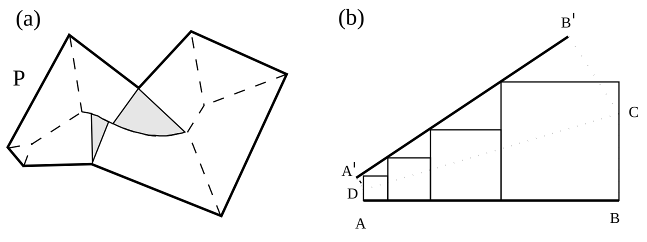

Fig. 12. A funnel.

cover a funnel t.
b) All funnels of P, if there are any, can be detected in $O(n \log n)$ time [11].

Since we can find all the funnels in P in $O(n \log n)$ time, we will start the covering procedure (Steps 2 and 5) by covering all funnels. Let t be any funnel of F, and let A, B, A' and B' be the vertices of the funnel in counter-clockwise order, Fig. 12. We may assume w.l.o.g. that AB is horizontal and at the bottom and that $|BB'| \geq |AA'|$. We start by placing a square s_1 on AB with lower left corner at A and upper left corner at $A'B'$, then place a square s_2 on AB with lower left corner at s_1's lower right corner and upper left corner on $A'B'$. Continue as described until AB is entirely included by the produced set of squares. Let S denote the set of squares, $s_1, \ldots, s_m$, produced to include AB. If $A'B'$ is also included in S, i.e. if the interior angle $\angle ADC$ is 90°, then we are finished, otherwise we do the corresponding procedure on $A'B'$.

Since the squares in a covering of P have to include all the edges of P, it is easily seen that the number of squares produced in this step is optimal. Now, since every square can be produced in constant time, the time-complexity for H_2 is $O(n \log n + \mu(F))$ and the number of squares produced is $H_2(P) = \mu(F)$, where $\mu(F)$ is the minimum number of squares needed to cover the set of funnels in P.

Step 3: Partitioning the edges into groups (H_3)
Partition the remaining uncovered edges of P into groups, $g_1, \ldots, g_r$, such that all the edges in a group can be included in a line with the same slope as the edges in the sequence. Partitioning the edges into $O(n)$ groups using an ordinary sorting-algorithm takes $O(n \log n)$ time.

Step 4: Partitioning the groups into sequences (H_4)
Now partition the groups into sequences such that all the edges in a sequence may be included in a non-empty rectangle. For each group g_i we do the following: let l_i be the line that includes all the edges in a group g_i. We check for every edge of P, whether the edge crosses l_i. If some edge crosses the line between two edges in g_i, the group is partitioned into two smaller groups. If no edge of P crosses l_i then the group is said to be a *sequence*. Continue as described until all the groups are partitioned into sequences. The sequences are denoted $s_1, \ldots, s_p$, and note that a sequence may consist of one single edge. The time-complexity of H_4 is $n \cdot \sum_i O(\log |g_i|) = O(n^2)$, since we have to go through all the edges for every group.

Step 5: Cover the remaining uncovered Voronoi faces (H_5)
This step of the algorithm consists of independently processing each sequence and outputting a set of squares which covers the Voronoi faces induced by the edges in the sequence, and the concave faces induced by the concave vertices of the edges in the sequence. Let E be an arbitrary sequence of P. Rotate the polygon, in such a way that the edges in E are horizontal and at the bottom, and let $e_1, e_2, \ldots, e_k$ be the set of edges in E, ordered from left to right. Recall that every edge, e, of P induces a Voronoi face, $V(e)$, in P, Fig 11.

We will now produce a set of squares, S, that covers the Voronoi-cells, $V(e_1),\ldots,V(e_k)$, and the faces induced by the concave vertices in E, and then prove that the number of squares in S is at most $24\times\mu(P)$.

A square in a polygon is said to be *maximal* if it is not contained by any larger square lying in the polygon.

The covering is done in three simple steps.

1. Place the largest possible square on the sequence with lower left corner at e_1's left endpoint. If the square isn't maximal extend it to the left. Denote this square s_1. If s_1's lower right corner lies between two edges e_i and e_{i+1} then place the largest possible square on the sequence with lower left corner at e_{i+1}'s left endpoint. If the square isn't maximal extend it to the left. This square is denoted s_2. Otherwise, if s_1's lower right corner lies on an edge $e_i \in E$ place the largest possible square on the sequence with lower left corner at s_1's lower right corner, Fig. 13. If the square isn't maximal extend it to the left. This square is denoted s_3. Continue to cover the edges as described until all the edges in E are entirely included by $S=\{s_1,\ldots,s_q\}$.

2. Let $R(s_i)$, $1\leq i\leq q$, be the Voronoi region or regions that are induced by the edges in the sequence, such that every point in $R(s_i)$ has a perpendicular projection on s_i's base. If $R(s_1)$ isn't entirely covered by s_1 then let p be that uncovered point in $R(s_1)$ with longest perpendicular projection on s_1's base. Now, place a square s_{q+1} with center in p and with bottom corner at p's projection on s_1's base, Fig. 13. It is easily seen that s_{q+1} covers the remaining uncovered region of $R(s_1)$ and, that s_{q+1} lies entirely within P. Continue to cover the remaining uncovered regions of $R(s_2),\ldots,R(s_q)$.

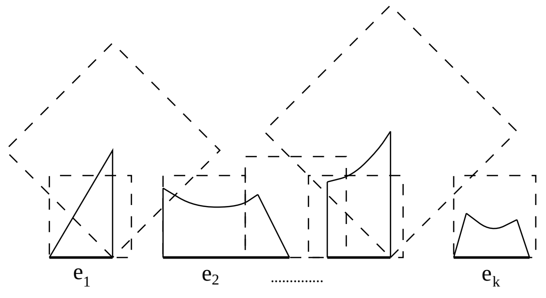

Fig. 13. The Voronoi faces induced by $e_1,\ldots,e_k$ are entirely covered by the squares (dashed) produced in Steps 1-2.

3. It remains to cover the faces induced by the concave vertices of P. Let C be a concave vertex of P. Split the concave angle into two equal angles. The two resulting regions, c_1 and c_2, are denoted *concave cells*, Fig. 14.

Definition 3. *If a concave cell, c, shares a Voronoi edge with any of the Voronoi faces induced by the edges of a sequence, E, then c is said to* belong *to E.*

A concave cell can be covered by two squares within P as follows. The Voronoi edge of the cell with one endpoint in the concave vertex W is called the *arm* of the cell. Partition the cell into two subcells by splitting the angle at W into two equal angles. Calculate the point, p_1 (p_2) in each subcell with largest distance to W. Construct two squares such that they both have one corner in W and center in p_1, respectively p_2. It is easily seen that these two squares cover the cell and, according to the definition of the medial axis, they lie entirely inside the polygon.

Since the edges in E are horizontal and at the bottom, it holds that every concave cell that belongs to E lies entirely above, (i.e. has larger y-coordinates than), the edges in E. Thus a square that includes two or more edges of E covers all the concave cells that belong to E, between these edges. Now, cover every uncovered cell that belongs to E.

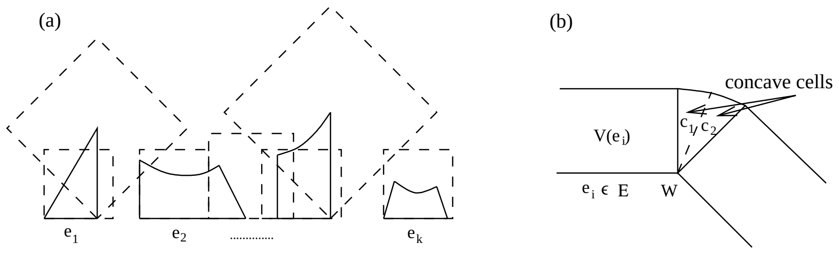

Fig. 14. The concave cell c_1 belongs to E.

Let OPT be a set of $\mu(P)$ squares that covers P and let $OPT(E)$ be a minimum set of squares that covers $V(e_1), \ldots, V(e_k)$. It is obvious that $|OPT(E)|$ is greater than or equal to q (the number of squares produced in step 5.1).

There are at most two uncovered concave cells for every square produced in step 5.1, thus the number of squares in S is at most 6 (=2 squares + (2 concave cells $\times$ 2 squares)) times greater than the number of squares in $OPT(E)$. A square in OPT may include edges or part of edges at all its four sides, and since the squares produced by H_5 cover one sequence at a time it's possible that every square in S only includes edges on one side. Thus the number of squares produced in this step is at most 2 squares +(2 concave cells $\times$2 squares)$\times$4 sides=24 times the number of squares in OPT.

Constructing a square takes $O(n)$ time and there are at most $O(n)$ squares, according to Fact 2, constructed in this step, since the funnels are covered by H_2. Hence, the time-complexity for H_5 is $O(n^2)$.

Steps 1 and 3 take $O(n \log n)$ time, Step 2 takes $O(n \log n + \mu(P))$ time and, finally, Steps 4 and 5 take $O(n^2)$ time. Thus the total time-complexity for H is $O(n^2 + \mu(P))$.

Summarizing the main result of this section we obtain the following theorem.

Theorem 4. *For any polygon P, without any acute interior angles, with n vertices, Algorithm H produces a covering of P with at most $24 \times \mu(P)$ squares in $O(n^2 + \mu(P))$ time.*

3.1 Improvements

In this section we will show how to improve the approximation factor, from 24 to 14, by producing each square within P that includes edges or parts of edges at three or four of its sides.

Definition 4. *A square r is said to be a 3-square in P iff r includes edges or parts of edges (of length > 0) of P on at least three of its sides.*

The new algorithm is denoted H', and it's an extended version of H. The only difference between Steps 1-4 of H and H' is that an extra step is added between Step 3 and 4 in H', which produces all 3-squares and then marks the edges in the sequences included in these squares.

To find all 3-squares in P we first group the edges in P into a minimum number of groups, $S_1, \ldots, S_p$, such that the slope of the edges in S_i, $1 \leq i \leq p$, differ by exactly a multiple of 90°. This grouping can easily be done in $O(n \log n)$ time by first sorting the slopes, and then scanning the sorted list.

Next, for each group we find its 3-squares by first rotating the polygon (or the coordinate system), such that the edges in the group are vertical or horizontal, and then constructing the generalized Voronoi diagram for these edges in the L_∞-metric, Fig. 15. Every vertex and isothetic edge in the Voronoi diagram corresponds to an isothetic rectangle. For every Voronoi vertex the corresponding (isothetic) rectangle is the maximal square, whose center lies at the vertex. Thus, every Voronoi vertex connecting three or four Voronoi edges corresponds to a square, whose center lies at the vertex.

Then each square is examined to decide whether it is a 3-square within P. The total number of squares examined according to the above procedure is $O(n)$, and for each one of them it takes $O(n)$ time to decide whether it is a 3-square within P. Thus, the total time to find all valid 3-squares in P is $O(n^2)$.

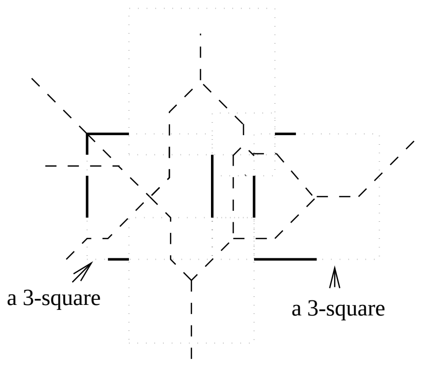

Fig. 15. The Voronoi diagram (dashed) in L_∞-metric.

The edges or parts of edges that are included in the boundaries of the 3-squares are marked in the groups produced in H_3' before H_4'. Now, all the 3-squares in an optimal covering are produced and we will now prove that the total number of squares produced by H' is at most $14 \times \mu(P)$.

To improve the approximation factor we have to modify the step of H' corresponding to H_5 (Section 3), denoted H_5'.

Step $\boldsymbol{H_5'}$ Let $\mathcal{E}$ be an arbitrary sequence of P, and let R be the set of 3-squares produced by H', such that every square in R includes edges or parts of edges of $\mathcal{E}$. Rotate P and all the produced squares, such that $\mathcal{E}$ is horizontal and at the bottom. Order the squares in R from left to right $r_1, \ldots, r_s$. Let E_1 be the subset of $\mathcal{E}$ to the left of r_1, let E_{s+1} be the subset of $\mathcal{E}$ to the right of r_s and let E_i, $2 \le i \le s$, be the subset of $\mathcal{E}$ between r_{i-1} and r_i. The subset E_i, $1 \le i \le s+1$, is denoted a *subsequence* of $\mathcal{E}$, Fig. 18. Note that a subsequence may be an empty set. The edges or parts of edges in E_i are denoted $e_{i_1}, \ldots, e_{i_{q_i}}$ (ordered from left to right).

Now, for every subsequence of $\mathcal{E}$ we will cover the cells that are *associated* to E_i (to be defined below), $1 \le i \le s+1$, in three different ways. Then we will select the covering which produces the smallest number of squares. The three different coverings are computed as follows:

Covering 1. Place the largest possible square on the sequence with lower left corner at e_{i_1}'s left endpoint. If the square isn't maximal extend it to the left. Denote this square $s_1^1(E_i)$. If $s_1^1(E_i)$'s lower right corner lies between two edges e_{i_j} and $e_{i_{j+1}}$, $1 \le j < q_i$, then place the largest possible square on the subsequence with lower left corner at $e_{i_{j+1}}$'s left endpoint. If the square isn't maximal extend it to the left. This square is denoted $s_2^1(E_i)$. Otherwise, if $s_1^1(E_i)$'s lower right corner lies on an edge e_{i_j} place the largest possible square on the sequence with lower left corner at $s_1^1(E_i)$'s lower right corner, Fig. 13. If the square isn't maximal extend it to the left. This square is then denoted $s_2^1(E_i)$. Continue to cover the edges as described until all the edges in E_i are entirely included by $SEQ_1'(E_i)=\{s_1^1(E_i), \ldots, s_{k_1}^1(E_i)\}$. Note that the number of squares in $SEQ_1'(E_i)$ is equal to the minimum number of squares needed to include the edges or parts of edges in E_i.

Covering 2. Instead of placing the first square on e_{i_1}'s left endpoint, we place the largest possible square, s^2_1, on the sequence with lower left corner at r_{i-1}'s bottom right corner. Now, continue as described in the previous paragraph until all the edges in E_i are entirely included by $SEQ'_2(E_i)=\{s^2_1(E_i),\ldots,s^2_{k_2}(E_i)\}$. Note that this covering is not done for E_1.

Covering 3. In the last covering we cover the subsequence from right to left, thus we do exactly as in the previous covering, but instead we place the largest possible square on the subsequence with lower right corner at r_i's bottom left corner. Continue as described in Covering 1 (but now from right to left) until all the edges in E_i are entirely included by $SEQ'_3(E_i)=\{s^3_1(E_i),\ldots,s^3_{k_3}(E_i)\}$. Note that this covering is not done for E_{s+1}.

The edges in E_i are now included in three different coverings. Before we compare the three different coverings, we have to cover the remaining uncovered regions of the cells that are *associated* to the subsequence. We define them below.

Let E_i be an arbitrary subsequence of $\mathcal{E}$, let $C(E_i)=\{c_1(E_i),\ldots,c_{p_i}(E_i)\}$ be the concave cells that belong (see Definition 3) to E_i ordered from left to right and, let e_{r_i} be the edges or parts of edges in $\mathcal{E}$ included in r_i. The left and right endpoints of e_{r_i} may be concave vertices that induce concave cells. If the left endpoint of e_{r_i} induces two concave cells then the concave cell entirely above $\mathcal{E}$ is denoted $c_{p_{i-1}+1}(E_{i-1})$; this cell is said to be *associated* to E_{i-1}. If the right endpoint of e_{r_i} induces two concave cells then the concave cell entirely above $\mathcal{E}$, $c_0(E_i)$, is *associated* to E_i. The concave cells $c_1(E_i),\ldots,c_{q_i}(E_i)$ and $c_0(E_i)$ and/or $c_{q_i+1}(E_i)$, if they exist, plus the Voronoi cells induced by the edges in E_i are said to be *associated* to E_i.

The remaining uncovered cells associated to E_i are covered as described in Step 2 and 3 of H_5. The three resulting sets of squares are denoted $SEQ_1(E_i)$, $SEQ_2(E_i)$ and $SEQ_3(E_i)$. Now all the cells associated to the subsequence are covered by three different coverings, and we can easily compare the number of squares in each of them, and then select a covering which produces the smallest number of squares. The smallest set is denoted $SEQ(E_i)$. This is done for all the subsequences of P.

Definition 5. *Let r be an arbitrary 3-square in P. If r overlaps with edges of P on all its four sides then let any of the four sides be r's* base. *Otherwise, one side of r doesn't overlap with any edges of P, and the opposite side of r is then called the* base *of r.*

There might still be uncovered concave cells that belong to the edges in $\mathcal{E}$. Each such concave cell has to be induced by some concave vertex lying on the base of a 3-square. Let r be an arbitrary 3-square in R that includes edges or parts of edges on three of its sides. We may assume that r's base is horizontal and at the bottom. There may be uncovered concave cells, belonging to the edges or parts of edges lying on r's base, that sticks out above the top of r. To cover these uncovered cells let p_r be the point (with a perpendicular projection on r's base) in the uncovered cells with largest distance to r's base and let p'_r be the perpendicular projection of p_r on r's base. Place a maximal square r' with bottom corner on p'_r and center in p_r. It is easily seen that all the remaining uncovered concave cells that belongs to the edges or parts of edges lying on r's base is covered by r'. We say that r' is the *companion* of r.

Analysis of H' Let OPT be a set of $\mu(P)$ squares that covers P and let OPT' be the set of 3-squares in OPT.

We start by calculating the number of 3-squares and companions produced by H'. To do this we need some definitions and observations.

Definition 6. *The set of edges or parts of edges of P lying on the base of a 3-square r is called the* base-set *of r. Assume that r's base lies horizontal and at the bottom. Let r_l and r_r denote the set of edges or parts of edges of P lying on r's left respectively right side, and let p_l and p_r denote the point in r_l respectively r_r with longest distance to r's base. If p_l lies closer to r's base than p_r then r_l is called the* side-set *of r, otherwise r_r is said to be the* side-set.

Definition 7. *Let r be a 3-square. We may assume that r's base is horizontal and at the bottom. A square S in OPT is said to be a* sponsor *to r iff at least one of the following two conditions holds:*

1. *S overlaps part of the base-set of r, or*
2. *S overlaps a part of r's side-set, of length > 0, that includes the point of the side-set with shortest distance to r's base.*

If r has a companion, r', then S also sponsors r'

Observation 2 *A square S in OPT sponsors at most four 3-squares and companions in P.*

Proof. Note that if a square S in OPT sponsors a 3-square r, and r and S aren't identical then one side of S lies entirely inside r and doesn't overlap any edges of P.

Rotate S such that the sides of S are vertical or horizontal. Now, S is either a 3-square or an "ordinary" square.

If S is a 3-square then three of its sides overlap edges or parts of edges of P, and S sponsors the identical 3-square produced by H' and its companion. Hence, if S is a 3-square there is only one side of S that may be entirely overlapped by a different 3-square. Below we will prove that S sponsors at most two other squares produced by H'.

In the continuation of this proof we will assume w.l.o.g. that S sponsors a 3-square r_1, S overlaps r_1's right side and, at least partially, bottom side, and that S and r_1 aren't identical, Fig. 16. Since all squares in OPT are maximal and since r_1 and S aren't identical, S's bottom side can't lie entirely within a 3-square.

If S isn't a 3-square then let r be a square, not identical to S, that overlaps S's entire top side. Note that, if S sponsors r then r's bottom side has to overlap the edges on S's bottom side, or r's right side has to overlap the edges on S's right side.

> If r overlaps the edges on S's bottom side then it also has to overlap S's entire left or right side to be a 3-square, since r has to be larger than S.
> If r doesn't overlap the edges on S's bottom side it has to overlap the edges lying on S's right side. Note that r is bounded from above by the edges lying on r_1's top or right side. From this we get that both r's bottom and left side lies entirely inside S and r_1 and don't overlap any edges of P. In this case r can't be a 3-square.

Hence, if S isn't a 3-square, we have that every 3-square that overlaps S's entire top side also has to overlap either S's left or right side. That is, if S isn't a 3-square, it remains to study the 3-squares that may overlap S's entire left or right side.

For S's left side there are two possible cases:

1. S partially overlaps r_1's base-set.
   In this case there may exist a 3-square r_2 that overlaps the left side of r_1 and r_2's bottom side overlaps part of S's bottom side. It holds that the left side of r_2 has to be r_2's base. Since r_2 is a 3-square and since r_2's side-set lies to the left of r_1, we have that S can't overlap any part of r_2's side-set. Hence S doesn't sponsor r_2 or its companion.
2. S partially or entirely overlaps r_1's side-set, Fig. 16b.
   Assume that r_1's bottom side partially or entirely overlaps the side-set, and that the right side of r_1 is the base of r_1. There may exist a 3-square r_2 that overlaps the left side of r_1 and the entire left side of S. We have three cases:
   (a) r_2 is smaller than r_1, Fig. 16b.
   It holds that the left side of r_2 has to be r_2's base, thus the upper side of r_2 has to include the side-set of r_2. Hence S doesn't sponsor r_2 or its companion.
   (b) r_2 is larger than r_1, Fig. 16c.
   In this case we have that: (1) the left endpoint of the edges lying on r_1's upper side lies to the left of the left endpoint of the edges lying on r_1's bottom side, since the edges lying on r_1's bottom side have to be the side-set of r_1, and (2) r_2's right side lies to the left of the

left endpoint of the edges lying on r_1's upper side, since the edges lying on r_2's bottom side have to be the side-set of r_2. Hence, S's bottom side can't overlap any edges lying on r_2's bottom side. That is, S doesn't sponsor r_2 or its companion.

(c) r_2 and r_1 have equal size, Fig. 16d.
In this case S may sponsor both r_1 and r_2. Since the left side of r_2 overlaps the base-set of r_2, we have, according to *case 1*, that S doesn't sponsor any 3-squares that overlap r_2's left side. Note that the boundary of $r_1 \cup r_2$ describes a rectangle which includes edges or parts of edges on all four sides. Thus, it is easily seen that neither r_1 nor r_2 have any companion.

Since there are edges of P lying on r_2's left side we have that every 3-square to the right of r_2 has to be smaller than r_2. Hence, according to Case 2a, these 3-squares aren't sponsored by S. The case when the left side of r_1 is the base of r_1 is symmetrical to the above case, just switch r_1 and r_2.

We have established that, if S is a 3-square then there is at most one side of S that can be entirely overlapped by a 3-square that isn't identical to S. If S isn't a 3-square then it remains to study the 3-squares that overlap S's entire right side. Hence, we have that if S's right side is included in a 3-square, r_3, then this case is symmetrical to *case* 1 and 2 above, and, thus, we will have the same symmetrical cases.

If S is a 3-square then S sponsors at most three squares. Otherwise, if S isn't a 3-square, then S sponsors at most four of the squares produced by H'.

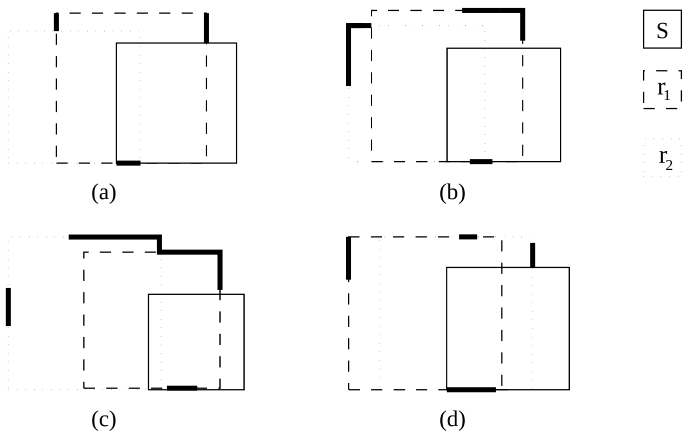

Fig. 16. A square $S \in OPT$ sponsors at most two 3-squares.

Observation 3 *For every 3-square r, $r \notin OPT$, there are at least two squares in OPT that sponsor r and its companion.*

Proof. We may assume that r's base is horizontal and at the bottom. It is easily seen that if $r \notin OPT$, then there must exist squares in OPT that overlap the base- and side-set of r. We may assume that a square $S_1 \in OPT$ overlaps part of the base-set of r, Fig. 17a. Since S_1 and r aren't identical and since S_1 is maximal, we have that S_1 can't overlap the segments lying on r's left- or right side. Hence, there must exist another square $S_2 \in OPT$ that includes the part of the side-set of r with shortest distance to r's base. Thus, both S_1 and S_2 have to sponsor r and its companion.

By using the above observations we can now prove the following Lemma.

Lemma 1. *The number of 3-squares and companions produced by H' is at most*

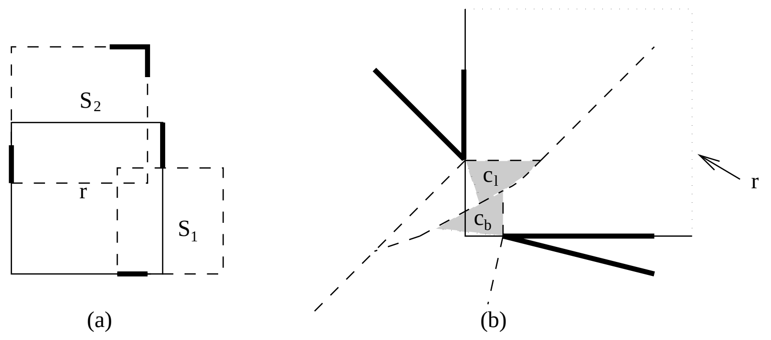

Fig. 17. (a) There are at least two squares in OPT that sponsor every 3-square $r \notin OPT$. (b) At most one of the concave cells associated to r's lower left corner may stick out to the left or below r.

$$3|OPT'|+2(|OPT|-|OPT'|).$$

Proof. To simplify the proof we say that each 3-square and companion produced by H' costs exactly one credit to construct. We will in this proof show that the squares in OPT can give one credit to each 3-square and companion and that the total number of credits given to the produced squares won't exceed $3|OPT'|+2(|OPT|-|OPT'|)$. For every square $S \in OPT$ two cases may occur. Recall that S sponsors at most four 3-squares or companions.

- If S is a 3-square then S gives one credit to the 3-square produced by H' that is identical to S and one credit to its companion.
  If S also sponsors two other 3-squares, r_1 and r_2, then S gives half a credit each to r_1 and r_2. According to Observation 2 *Case 2c*, r_1 and r_2 don't have any companions. Thus, in this case, S gives at most three credits to the squares produced by H'.
  Otherwise, if S just sponsors one other 3-square, r, then S gives half a credit to r and half a credit to its companion. Thus S gives at most 3 credits to the squares produced by H'.
- If S isn't a 3-square then S gives half a credit to each 3-square it's sponsoring and half a credit to each of the companions of the 3-squares. Thus, S will give at most two credits to the squares produced by H'.

Since every 3-square r, $r \notin OPT$, has two squares in OPT as sponsors, according to Observation 3, we have that r will get at least one credit from the squares in OPT, and every 3-square that is identical to a square S in OPT gets one credit from S. Thus, all the 3-squares produced by H' get one credit each from the squares in OPT and the total number of credits given to the produced 3-squares and their companions is at most $3|OPT'|+2(|OPT|-|OPT'|)=|OPT'|+2|OPT|$.

As described in the previous section, there may be concave vertices lying on the perimeter of a 3-square r in OPT' that induce concave cells, that aren't covered by r and r'. Assume that the sides of r are horizontal or vertical and denote by r_b the set of edges or parts of edges lying on the bottom side of r. The left and right endpoint of r_b may be concave vertices that induce Voronoicells. These cells may stick out to the left respectively to the right of r. In the following fact we bound the number of cells which "stick out".

Fact 3 *For each 3-square r in OPT' there are at most four concave cells induced by concave vertices lying on the perimeter of r that are not completely covered by r or its companion r'.*

Proof. Let r be an arbitrary 3-square in P. Rotate r and P such that the sides of r become horizontal or vertical. Let r_b and r_l be the edges or parts of edges lying on r's bottom respectively left side. If the left endpoint of r_b and/or the lowest bottom endpoint of r_l induces concave Voronoicells, then we associate these cells to the lower left corner of r.

We may assume that the left endpoint of r_b and the bottom endpoint of r_l are concave vertices that induce the Voronoicells c_b and c_l. Otherwise, if any of these endpoints don't induce a cell, then it is easily seen that there is at most one cell associated to r's lower left corner that sticks out to the left of r's left side or below r' bottom side.

Recall that a Voronoicell, c, induced by a concave vertex is partitioned into two concave cells, c_1 and c_2, Fig. 14. That is we will only consider the concave cell of c_b entirely above r_b and the concave cell c_l entirely to the right of r_l. These concave cells are denoted c_{b_1} and c_{l_1}. Since c_{l_1} lies to the right of r_l, c_{b_1} lies above r_b and since two Voronoicells can't overlap, it is easily seen that at most one of the concave cells, c_{l_1} and c_{b_1}, associated to r's lower left corner may stick out outside r's left side or below r's bottom side, Fig. 17b. By symmetry the same holds for each corner of r.

Now we are ready to calculate the total number of squares produced by H'. Let $OPT(\mathcal{E})$ be the subset of OPT, such that every square in $OPT(\mathcal{E})$ includes edges or parts of edges of $\mathcal{E}$ and let $OPT(E_j)$, $1 \leq j \leq s+1$, be the subset of $OPT(\mathcal{E})$, such that every square in $OPT(E_j)$ includes edges or parts of edges of E_j. We may assume w.l.o.g. that every square in OPT is maximal. Note that $OPT(E_j) \cap OPT(E_{j+1}) = \varnothing$.

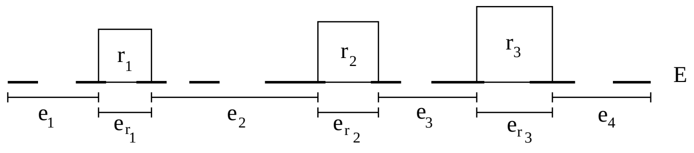

Fig. 18. A sequence E divided into subsequences.

We will in the remaining proof show that the number of squares produced by H'_5 to cover the Voronoi cells associated to $E_1, \ldots, E_{s+1}$ is at most $6 \times |OPT(\mathcal{E})|$. This is done by giving to each subsequence an amount of credits (from the squares in $OPT(\mathcal{E})$), such that the total amount doesn't exceed $6 \times |OPT(\mathcal{E})|$. It then remains to prove that the number of squares produced by H'_5 is at most equal to the amount of credits given to the subsequences.

- If a 3-square r_i belongs to OPT' then, if e_{r_i} induces a concave cell that belongs to E_i then $r_i \in OPT$ gives two credits to E_i. Symmetrically the same is done for E_{i+1}, if e_{r_i} induces a concave cell that belongs to E_{i+1}. According to Fact 3 each 3-square in OPT will at most give the sequences of P a total of eight credits.
- If $i=1$ or $i=s+1$ then there are two cases. We may assume w.l.o.g. that $i=1$.
  - The squares in $OPT(E_1)$ don't overlap r_1. Then $OPT(E_1)$ gives $6 \times |OPT(E_1)|$ credits to E_1.
  - The squares in $OPT(E_1)$ overlap r_1. Then $OPT(E_1)$ gives $6 \times |OPT(E_1)| - 2$ credits to E_1 and two credits to E_2.
- The squares in $OPT(E_i)$ don't overlap r_{i-1} nor r_i, Fig. 19a. Then $OPT(E_i)$ gives $6 \times |OPT(E_i)|$ credits to E_i.
- The squares in $OPT(E_i)$ overlap r_{i-1} or r_i but not both, Fig. 19b. If they overlap r_{i-1} then $OPT(E_i)$ gives $6 \times |OPT(E_i)| - 2$ credits to E_i and two credits to E_{i-1}. (The other case is symmetrical.)
- The squares in $OPT(E_i)$ overlap both r_{i-1} and r_i, Fig. 19c. Then $OPT(E_i)$ gives $6 \times |OPT(E_i)| - 4$ credits to E_i, two credits to E_{i-1} and two to E_{i+1}.

Recall that all the concave cells that are associated to a sequence lie entirely above the edges in the sequence, see Fig. 14. Thus, a square that includes two edges in $\mathcal{E}$ covers the concave cells that are associated to $\mathcal{E}$ between the two edges.

Let $SEQ(\mathcal{E})$ be the set of squares produced by H'_5 to cover the cells associated to the sequences $E_1 \cup \ldots \cup E_{s+1}$ and let $SEQ'(E_i)$ be the set of squares in $SEQ(E_i)$ that includes edges or parts of edges of E_i. It remains to show that the number of squares produced by H'_5 to cover the cells

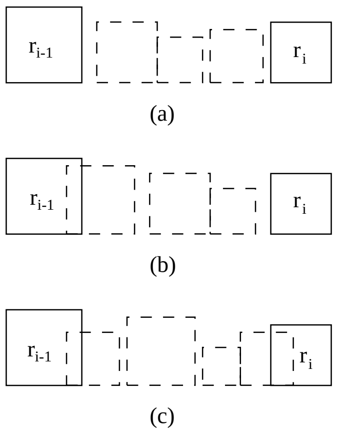

Fig. 19. The squares in $OPT'(E_i)$ may overlap r_{i-1} and/or r_i.

associated to each subsequence, $E_1, \ldots, E_{s+1}$, is at most equal to the amount of credits given to E_i by $OPT(E_{i-1})$, $OPT(E_i)$, $OPT(E_{i+1})$ and possibly by $OPT'(\mathcal{E})$ (if r_i or r_{i+1} belongs to OPT').

Note that $SEQ(E_i) \leq SEQ_k(E_i)$, where $1 \leq k \leq 3$.

- If $i=1$ or $i=s+1$ then there are two cases. By symmetry it suffices to treat the two cases for $i=1$.
  Case 1. *The squares in $OPT(E_1)$ don't overlap r_1.*
  Then $SEQ'_1(E_1)$ may or may not overlap r_1 and it holds that $|SEQ'(E_1)| \leq |OPT(E_1)|$. The number of squares in $SEQ(E_1)$ ($\leq |SEQ_1(E_1)|$) is at most $6 \times |OPT(E_1)|$ plus at most two needed to cover $c_{p_1+1}(E_1)$, if $c_{p_1+1}(E_1)$ exists. This is exactly the number of credits given to E_1 by $OPT(E_1)$ and $OPT(E_2)$ or OPT', since either $OPT(E_2)$ has to overlap r_1 or, otherwise, r_1 belongs to OPT'.
  Case 2. *The squares in $OPT(E_1)$ overlap r_1.*
  It holds that $SEQ'_3(E_1)$ also overlaps r_1 and $|SEQ'(E_1)| \leq |SEQ'_3(E_1)| \leq |OPT(E_1)|$. Since, in $SEQ_3(E_i)$, the concave cells between e_{r_1} and $e_{1_{q_1}}$ are covered by $s^3_{k_3}(E_1)$ and r_1, it holds that there are at most $|SEQ_3(E_i)| \leq 6 \times |OPT(E_1)| - 2$ squares in $SEQ(E_1)$, which is exactly the number of credits given to E_1 by $OPT(E_1)$.
- If the squares in $OPT(E_i)$ don't overlap r_{i-1} nor r_i, Fig. 19a, then $SEQ'_1(E_i)$ may or may not overlap r_{i-1} and/or r_i, and $|SEQ'(E_i)| \leq |SEQ'_1(E_i)| \leq |OPT(E_i)|$. Thus the number of squares in $SEQ(E_i)$ ($\leq |SEQ_1(E_i)|$) is at most $6 \times |OPT(E_i)|$ plus, if $c_0(E_i)$ and/or $c_{p_i+1}(E_i)$ exist, at most two to cover $c_0(E_i)$ respectively $c_{p_i+1}(E_i)$. This is exactly the amount of credits given to E_i by $OPT(E_i)$, $OPT(E_{i+1})$, $OPT(E_{i-1})$ and OPT', since (1) either $OPT(E_{i-1})$ has to overlap r_{i-1} or $r_{i-1} \in OPT'$, and (2) either $OPT(E_{i+1})$ has to overlap r_i or $r_i \in OPT'$.
- If the squares in $OPT(E_i)$ overlap either r_{i-1} or r_i but not both, Fig. 19b, then we may assume by symmetry that the squares in $OPT(E_i)$ overlap r_{i-1}. We have that $SEQ'_2(E_i)$ overlaps r_{i-1} (and maybe also r_i), and $|SEQ'_2(E_i)| \leq |OPT(E_i)|$. Thus the number of squares in $SEQ(E_i)$ ($\leq |SEQ_2(E_i)|$) is at most $6 \times |OPT(E_i)|$ minus two for $c_1(E_i)$, since this cell is already covered by the leftmost square in $SEQ'_2(E_i)$ and r_{i-1} (if there exists a concave cell $c_0(E_i)$ then this cell is also covered by these two squares), and plus two to cover $c_{p_i+1}(E_i)$, if $c_{p_i+1}(E_i)$ exists. This is exactly the amount of credits given to E_i by $OPT(E_i)$, $OPT(E_{i+1})$ and OPT', since $OPT(E_{i+1})$ has to overlap r_i or $r_i \in OPT'$.
- If the squares in $OPT(E_i)$ overlap both r_{i-1} and r_i, Fig. 19c, then we have two possible cases:
  Case 1. $|SEQ'(E_i)| = |OPT(E_i)|$.
  Then $SEQ'_2(E_i)$ or $SEQ'_3(E_i)$ overlaps both r_{i-1} and r_i. We may assume that $SEQ'_2(E_i)$ overlaps both r_{i-1} and r_i. Thus, the concave cells $c_1(E_i)$ and $c_{q_i}(E_i)$ are covered by r_{i-1} and $s^2_1(E_i)$ respectively r_i and $s^2_{k_i}(E_i)$ (if $c_0(E_i)$ and/or $c_{p_i+1}(E_i)$ exist then they are also

covered by these squares). Thus, it holds that the number of squares in $SEQ(E_i)$ is at most $SEQ_2(E_i){\le}6{\times}|OPT(E_i)|{-}4$, which is exactly the amount of credits given to E_i by $OPT(E_i)$.

Case 2. $|SEQ'|{<}|OPT(E_i)|$.

Then we argue that $|SEQ_2(E_i)| \le 6 \times |OPT(E_i)| - 6$. To prove this we first note that $|SEQ'_2(E_i)| \le |OPT(E_i)| - 1$, and that the square $s_1^2(E_i)$ in $SEQ'_2(E_i)$ touches r_i. Thus, $|SEQ_2(E_i)|$ is at most $6 \times (|OPT(E_i)| - 1)$ minus two for the concave cell induced by the left endpoint of e_{i_1}, since this is already covered by r_{i-1} and $s_1^2(E_i)$, and possibly (if $c_{p_i+1}(E_i)$ exists) plus two needed to cover $c_{p_i+1}(E_i)$. This is less than the amount of credits given to E_i by $OPT(E_i)$.

The number of concave cells induced by the edges in e_{r_i}, where r_i is a 3-square in OPT' is at most four, according to Fact 3. Thus, the total number of squares produced by H'_5 to cover these concave cells in P is at most $8{\times}|OPT'|$. We have that all the subsequences and their concave cells have been covered by $SEQ(E_i)$, $1{\le}i{\le}s{+}1$, and the number of squares in $SEQ(E_i)$ doesn't exceed the number of credits given to the subsequences. Thus $|SEQ(E_1)|{+}\ldots{+}|SEQ(E_{s+1})|{\le}6{\times}|OPT(\mathcal{E})|$.

Recall that every square in $OPT{-}OPT'$ may include edges on two of its sides, hence it may give credits to two subsequences. By using the result from (1) together with the above results we obtain that the total number of squares produced by H' is at most

$$(8{+}1){\times}|OPT'|{+}2{\times}|OPT|{+}(2{\times}6){\times}(|OPT|{-}|OPT'|){\le}14{\times}|OPT|.$$

3.2 Fat Rectangular Coverings and Similar Problems

We may use the two algorithms proposed in this paper for covering polygons with other types of polygons, denoted *q-polygons*, provided that the following two properties hold. Let k_1 and k_2 be two arbitrary, but fixed, integers.

1. A q-polygon is coverable by k_1 squares, and
2. A square is coverable by k_2 q-polygons.

q-polygons that fulfill these two properties are, for example, rectangles having bounded aspect ratio.

Now, let $\mu^*(P)$ denote the minimum number of q-polygons needed to cover an arbitrary polygon P. We have that $\mu(P){\le}k_1{\cdot}\mu^*(P)$. From the above properties it easily seen that by using the two algorithms proposed in this paper, for the q-polygon cover problem, we will get the following two results:

– An algorithm corresponding to the one presented in section 3 will produce at most $14 \cdot k_1 \cdot k_2 \cdot \mu^*(P)$ q-polygons to cover a polygon P in $O(n^2 + \mu^*(P))$ time.
– An algorithm corresponding to the one presented in section 2 will produce at most $11n \cdot k_2 + k_1 \cdot k_2 \cdot \mu^*(P)$ q-polygons to cover P in $O(n \log n + \mu^*(P))$ time.

References

1. M. Albertson and C. J. O'Keefe. Covering regions with squares. *SIAM Journal on Discrete Mathematics*, 2(3):240–243, 1981.
2. L.J. Aupperle, H.E. Conn, J.M. Keil, and J. O'Rourke. Covering orthogonal polygons with squares. In *Proceedings of the 26th Annual Allerton Conference on Communication, Control and Computation*, pages 97–106, 1988.
3. R. Bar-Yehuda and E. Ben-Hanoch. A linear-time algorithm for covering simple polygons with similar rectangles. *International Journal of Computational Geometry & Applications*, 6(1):79–102, 1996.
4. B.M. Chazelle. *Computational Geometry and Convexity.* PhD thesis, Department of Computer Science, Yale University, New Haven, CT, 1979. Carnegie-Mellon University Report CS-80-150.
5. J. Gudmundsson and C. Levcopoulos. Close approximations of minimum rectangular coverings. In *FST & TCS'96*, volume 1180 of *LNCS*, pages 135–146, 1996.
6. S. Hart and M. Sharir. Nonlinearity of Davenport-Schinzel sequences and of generalized path compression schemes. *Combinatorica*, 6:313–319, 1986.
7. A. Hegedus. Algorithms for covering polygons by rectangles. *Computer Aided Design*, 14(5), 1982.
8. J.M. Keil and J.-R. Sack. Minimum decompositions of polygonal objects. In G.T. Toussaint, editor, *Computational Geometry*, pages 197–215. North-Holland, Amsterdam, Holland, 1985.
9. D.G. Kirkpatrick. Efficient computation of continuous skeletons. In *Proceedings of the 20th Annual IEEE Symposium on Foundation of Computer Science*, pages 18–27, 1979.
10. C. Levcopoulos. A fast heuristic for covering polygons by rectangles. In *Proceedings of Fundamentals of Computation Theory*, volume 199 of *LNCS*, 1985.
11. C. Levcopoulos. Improved bounds for covering general polygons with rectangles. In *Proceedings of the 7th Foundations of Software Technology and Theoretical Computer Science*, volume 287 of *LNCS*, 1987.
12. C. Levcopoulos and A. Lingas. Covering polygons with minimum numbers of rectangles. In *Proceedings of the 1st Symposium on Theoretical Aspects in Computer Science*, volume 166 of *LNCS*, 1984.
13. D. Morita. Finding a minimal cover for binary images: an optimal parallel algorithm. Technical Report 88-946, Deptartment of Computer Science, Cornell University, 1988.
14. J. O'Rourke and K.J. Supowit. Some NP-hard polygon decomposition problems. *IEEE Transactions on Information Theory*, IT-29:181–190, 1983.
15. F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, New York, NY, 1985.
16. S. Rajasekaran and S. Ramaswami. Optimal parallel randomized algorithms for the voronoi diagram of line segments in the plane and related problems. In *Proceedings of the 10th ACM Symposium on Computational Geometry*, pages 57–66, 1994.
17. D.S. Scott and S.S. Iyengar. TID: a translation invariant data structure for storing images. *Communications of the ACM*, 29(5), 1986.
18. J. Snoeyink, C.A. Wang, and F. Chin. Finding the medial axis of a simple polygon in linear-time. In *Proceedings of the 6th Annual International Symposium Algorithms Comput.*, volume 1004 of *LNCS*, pages 382–391, 1995.
19. A. Wiernik and M. Sharir. Planar realization of nonlinear Davenport-Schinzel sequences by segments. *Discrete Computational Geometry*, 3:15–47, 1988.