On Approximating the Corner Cover Problem

Piotr Berman
Department of Computer Science & Engineering
The Pennsylvania State University
University Park, PA 16802
email: berman@cse.psu.edu

Bhaskar DasGupta
Department of Computer Science
Rutgers University
Camden, NJ 08102
email: bhaskar@crab.rutgers.edu

March 24, 1998

Abstract

The rectilinear polygon cover problem is one in which a certain class of features of a rectilinear polygon of n vertices has to be covered with the minimum number of rectangles included in the polygon. In particular, one can consider covering the entire interior, the boundary and the set of corners of the polygon. These problems have important applications in, for example, storing images and in the manufacture of integrated circuits. In this paper we consider covering the corners of the polygons, also known as the corner-cover problem. In [4], it was shown that the corner cover problem is NP-hard and a simple $O(n^{3/2})$-time approximation algorithm with performance ratio of 4 was presented. In this paper, we propose a non-trivial heuristic that approximates the corner cover problem for rectilinear polygons *without holes* with an improved performance ratio of 2 of the optimum in $O(n^2)$ time. The analysis of the heuristic involves a very careful amortization scheme and we believe similar technique may be useful for analysis of heuristics for other geometric problems as well.

Key words

Polygon, cover, rectangle, rectilinear, heuristics, approximation algorithms.

1 Introduction.

The rectilinear polygon cover problems are the ones in which one is required to cover a certain class of features of a rectilinear polygons with the minimum number of rectangles. Depending upon whether one wants to cover the interior, boundary or corners of the given polygon, the problem is termed as the *interior, boundary* or *corner* cover problem, respectively. Little progress has been made in finding efficient algorithms for covering arbitrary polygons with primitive shapes, and many such problems are known to be NP-hard [14]. Thus, the rectilinear polygon cover problems have received particular attention. These problems are also interesting because of their application in storing images [9], and in the manufacture of integrated circuits [10]. Also, an investigation of these problems has given rise to special kinds of perfect graphs of interest [11].

Masek [9] was the first to show that the interior cover problem is NP-complete for rectilinear polygons with holes. Conn and O'Rourke [2] later showed that the boundary cover problem is NP-complete for polygons with holes, even if the polygon is in general position. They also showed that the corner cover problem is NP-complete if we require each concave corner to be covered by two rectangles along both the perimeter segments defining the corner. For a long time the complexity of this problem was unknown for polygons without holes, until Culberson and Reckhow [3] showed the interior and boundary cover problems are NP-complete even if the polygon has no holes, and even if the polygon is required to be in general position. In [4] it is shown, among other results, that there is no polynomial-time approximation scheme for either the boundary or the interior cover problems unless $P = NP$.

Since the rectilinear cover problems are mostly NP-hard in general, there has been a lot of interest in finding exact solutions for special cases of these problems in polynomial time. Franzblau and Kleitman [6] gave a polynomial time algorithm for covering the interior of a vertically convex rectilinear polygon with the minimum number of rectangles, which improved a previous result of Chaiken et. al. [1]. Lubiw [7, 8] gave polynomial time algorithm for the interior cover problem for a somewhat larger class of polygons, called the plaid polygons. Conn and O'Rourke [2] gave polynomial time algorithm for covering the convex corners of a rectilinear polygon or horizontal perimeter segments of a rectilinear polygon in general position.

Regarding approximate solutions, Franzblau [5] analyzed a polynomial-time heuristic for the interior cover problem which approximates the optimum with a performance ratio of $O(\log\theta)$, where θ is the optimal cover size.

In contrast to the covering problem, the rectilinear polygon decomposition problem (when no overlapping of rectangles is allowed) has a polynomial time solution for polygons without degenerate holes [12, 15].

In this paper, we consider the corner cover problem. In [4], it was shown that the corner cover problem is NP-complete and a simple heuristic was proposed that runs in $O(n^{3/2})$ time with a performance ratio of 4. In this paper, we design and analyze a heuristic that approximates the corner cover problem for rectilinear polygons *without holes* within a ratio of 2 of the optimum in $O(n^2)$ time. The analysis of the heuristic involves a very careful amortization scheme and we believe similar technique may be useful for analysis of approximation heuristics for other geometric problems as well.

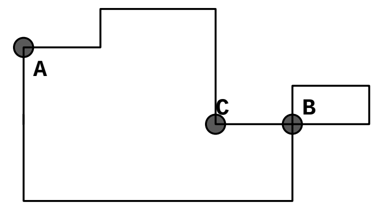

Figure 1: *In this rectilinear polygon A is a convex corner, C is a concave corner and B is a degenerate convex corner (the points are shown magnified)*

2 Preliminaries.

A *rectilinear polygon* is a polygon with its sides parallel to the coordinate axes. Such a polygon may or may not have holes, but if the holes are present they are also rectilinear.

We assume that the polygon is given as a sequence of its vertices such that the vertices of the polygon appear in a clockwise order and those of the holes appear in an anti-clockwise order. This ensures that the interior of the polygon is always on the right side of the boundary as we traverse the vertices in the given order. Unless stated otherwise explicitly, in all subsequent discussions we assume that the given polygon is *simple*, *i. e.,* no two non-consecutive edges of the polygon cross each other.

The corners of the given polygon can be classified into *convex*, *degenerate convex* and *concave* types (*fig.* 1). A *convex* corner is a corner produced by the intersection of two consecutive sides of the polygon which form a $90°$ angle inside the interior of the polygon. A *degenerate convex* corner is produced by the intersection of two pairs of edges forming two $90°$ angles. The remaining corners are the *concave* corners, produced by the intersection of two consecutive edges of the polygon which form a $270°$ angle inside the interior of the polygon.

The interior (*resp.* boundary, corner) cover problem for a rectilinear polygon is to find a set of rectangles (possibly overlapping) of minimum cardinality so that the union of these rectangles covers the interior (*resp.* boundary, corners) of the given polygon. For the corner cover, it is sufficient that each corner is on the boundary (possibly a corner) of one of the rectangles in the given set. Note that this differs from a similar problem as defined in Conn and O'Rourke[2] in which each concave corner of the given polygon has to be covered optimally by rectangles such that for some $\epsilon > 0$ every point on each of the two perimeter segment defining the concave corner, within distance ϵ of the concave corner, is covered by a rectangle. The corner cover problem is less demanding in the sense that it is sufficient for the above condition to hold for at least one of the perimeter segment defining the concave corner.

Although it is true that any cover of the interior also covers the boundary and any cover of the interior or boundary also covers the corners, these three cover sizes need not be the same. For example, in Fig. 5 of [1] the optimal corner cover size is 7 but the optimal boundary or interior cover size is 8, and in fig. 6 of [1] the optimal boundary cover size is 7 but the optimal interior cover size is 8. However, to our knowledge, there is no result in the existing literature which proves a *tight* non-trivial bound between the relative sizes of these three types of covers in general.

Initially, while describing our corner cover heuristic, we assume that the given polygon has no degenerate convex corner. Later, we show that the heuristic works with the same performance ratio

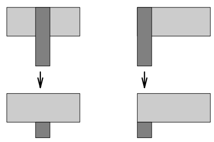

Figure 2: *Retraction of rectangles*

even if degenerate convex corners are allowed. Unless otherwise stated, in the rest of the paper, n denotes the number of vertices of the given rectilinear polygon without holes.

3 Approximation Algorithm for Corner Cover for Polygons Without Holes.

In this section we exhibit a heuristic that runs in $O(n^2)$ time and show that it achieves a performance ratio of 2 for the case when the given polygon has no holes. This improves the previously best known performance ratio of 4 for the case of the corner cover problem when the given polygon has no holes [4].

3.1 An Overview Of The Heuristic.

It is easy to preprocess a given rectilinear polygon of n vertices, in $O(n \log n)$ time, such that every corner of the given polygon has coordinates (x, y) with $x, y \in \{1, 2, \ldots, n\}$. After this step, an overview of the heuristic is as follows.

Input: A rectilinear polygon P without holes.
Output: A cover for the vertices of P.

Algorithm outline:

- The first phase partitions the polygon, if necessary, into smaller subpolygons.
- The second phase finds a cover of a part of the convex corners which also cover a part of the concave corners. The remaining uncovered corners are organized as a set of chains and loops.
- The third phase covers the chains and loops (in a rather greedy manner).

A more detailed description of the heuristic is given later in section 3.7.

The rectangles which are needed to cover the polygon can be classified into 5 types. A type$-i$ rectangle (for $0 \leq i \leq 4$) covers i convex corners. A *convex cover* is a corner cover that needs to cover convex corners only.

Proposition 3.1 *Rectangles in the convex cover can always be chosen so that either they overlap on interior of positive area or boundary of positive length, but not both.*

Proof. If two rectangles in our collection overlap on both interior of positive area and boundary of positive length, there can be essentially two different ways for them to do so as shown in *fig.* 2. We show how to avoid this in each case in *fig.* 2 by proper retraction of one of the rectangles so that this can be avoided. □

As a result of the above proposition, we know that each concave corner is covered by at most 2 rectangles of the rectangles and that each convex corner is covered by one rectangle in a convex cover, if the rectangles are retracted in the manner stated above.

We will term a cover which satisfies the criterion in Proposition 3.1 **and** also has the maximum number of type-3 and type-4 rectangles as a *proper cover.* One of the proper covers is also an optimal cover[1]. Henceforth, by an optimal cover we mean an optimal cover which is also a proper cover. So, a proper cover has the same number of type-3 and type-4 rectangles as an optimal cover. We redefine a *convex cover* to be a proper cover that needs to cover convex corners only. Henceforth, whenever we refer to a cover, we mean a proper cover only.

The following result is proved by Conn and O'Rourke[2].

Proposition 3.2 *For a given rectilinear polygon P without hole having n vertices, all the type-3 and type-4 rectangles of a proper cover can be found in $O(n)$ time.*

A collection of type-2 rectangles is called *maximal* if no two of all the remaining convex corners not covered by this collection of type-2 rectangles can be covered together by a single rectangle. Let c be the number of concave corners of P. Assume that a optimal convex cover has μ_i rectangles of type-i. We observe the following (see also [13]).

Observation 3.1 $c = \sum_{i=1}^{4} i\mu_i - 4$.

Let θ be the size of the optimal cover. We immediately observe the following.

Observation 3.2 $\sum_{i=1}^{4} \mu_i + \frac{c-\mu_3-2\mu_4}{2} \leq 2\theta - 2$.

Proof. The optimal cover must be of size at least as large as the optimal convex cover. This implies $\theta \geq \sum_{i=1}^{4} \mu_i$. On the other hand, $\frac{c-\mu_3-2\mu_4}{2} \leq \sum_{i=1}^{4} \mu_i - 2 \leq \theta - 2$ by Observation 3.1. □

Of course, our heuristic may have different number of type-i rectangles compared to an optimal cover. However, by choosing a proper cover we can guarantee that our cover has at least the *same number* of type-3 and type-4 rectangles as in an optimal cover. Also, as we will see, our heuristic will choose a *maximal* collection of type-2 rectangles.

Observation 3.3 *Assume that we have selected the type-3 and type-4 rectangles for a proper cover, and we have a maximal collection of n_2 type-2 rectangles, and let n_1 is the number of the remaining uncovered convex corners. Also, assume that μ_i is the number of type-i rectangles in the proper optimal cover. Then, $n_1 \leq \mu_1 + 2.(\mu_2 - n_2)$.*

Proof. Remember that no two rectangles in a proper cover cover the same convex vertex. Hence, $n_1 + 2.n_2 = \mu_1 + 2.\mu_2$. □

Observation 3.4 *The result of observation 3.3 holds even if we have a non-maximal collection of type-2 rectangles.*

[1] This follows by observing that if a type-3 (resp. type-4) rectangle is not in an optimal cover, then a rectangle covering one of the 3 (resp. 4) convex corners can be extended to cover the remaining 2 (resp. 3) convex corners

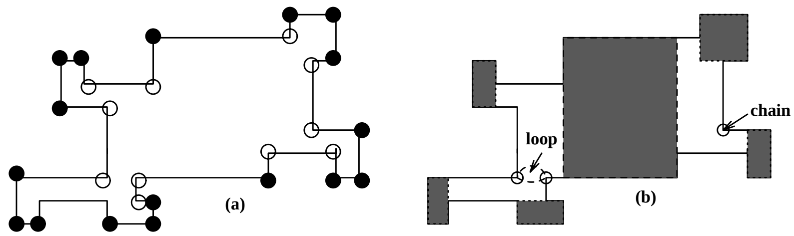

Figure 3: *Chain and loops*

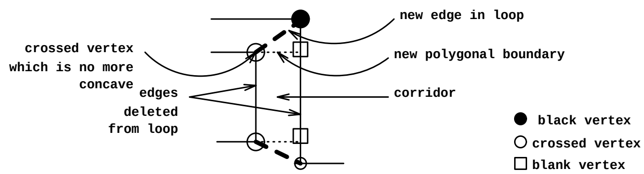

Figure 4: *Corridor decomposition creating two new loops from a single loop*

Proof. Assume that we have a non-maximal collection of type-2 rectangles. Compare this with a maximal collection of type-2 rectangles which includes our collection. Each rectangle not present in our collection will produce 2 more type-1 rectangles, and will decrease the number of type-2 rectangles by 1. □

Lemma 3.1 *Let n_i be the number of type-i rectangles used by a proper cover. Then, if $n_0 \leq \frac{c-2n_4-n_3-n_1}{2}$, this proper cover uses at most $2\theta - 2$ rectangles in the cover.*

Proof. Notice that, by Observation 3.1, $n_0 \leq n_2+n_3+n_4-2$. From the definition of a proper cover, $n_4 = \mu_4$ and $n_3 = \mu_3$. We also know that $\theta \geq \sum_{i=1}^{4} \mu_i$ and, by Observation 3.4, $n_1 \leq \mu_1+2.(\mu_2-n_2)$, and hence

$$\sum_{i=0}^{4} n_i \leq \mu_4 + \mu_3 + n_2 + \mu_1 + 2.(\mu_2 - n_2) + (n_2 + \mu_3 + \mu_4) - 2 = 2.(\mu_4 + \mu_3 + \mu_2) - 2 \leq 2\theta - 2$$

□

3.2 Corner Marking, Satisfactory Cover And Corridors.

We define a *chain* as follows. The vertices in the chain consist of the convex corners (marked as *black* corners) and the concave corners (marked as *crossed* corners) of the given polygon P, and edges between corners are introduced only if two corners can be covered by a single rectangle. A *loop* is a closed chain, i.e., with the *same* start and end corner.

Initially, the set of edges is the same as the boundary segments of the given polygon P, and corners of P from a single loop (*fig.* 3(a)). After the second phase of our heuristic, the uncovered concave corners will form a disjoint collection of chains and loops (*fig.* 3(b)).

To obtain a satisfactory cover, we will sometimes need to apply the *corridor decomposition* procedure. This procedure breaks a loop into a number of smaller loops by replacing some edges, and possibly introducing new vertices. Conceptually this means that we partition the given polygon into polygons with smaller number of corners. Such a partition is shown schematically in *fig.* 4. It may introduce new corners, which are always convex and **do not need** to be covered. We call these vertices as *blank* or *unmarked* corners. The types of rectangles (i.e., type-i rectangle for $1 \leq i \leq 4$)

inside each partitioned polygon is determined according to the *black corners only.* Also, after such decomposition some crossed corners may cease to be concave corners. For the purpose of corner cover, the heuristic will then consider the new loop or chain connecting this corner to its adjacent corners (see *fig.* 4). Hence, the heuristic will really be considering chains or loops, which may or may not follow the polygonal boundary of the polygons from the partition, but the partitioned polygons allow us to visualize the loops better. However, note that the additional corners or edges introduced satisfy the following:

- The total number of new corners or edges introduced (which are not one of the corners or edges of the given polygon) is at most $O(n)$.
- These new edges only serve the propose of decomposing the given polygon into smaller pieces.

Throughout the paper, in all the figures, we denote a black vertex by $\bullet$, a cross vertex by $\circ$ and a blank or unmarked vertex by $\square$. A marked corner is either a black or a crossed corner.

The initial marking of corners of P as black or cross will never be changed later, even if we partition the given polygon.

Assume inductively that we have a partition $P_1, P_2, \ldots, P_k$ of the polygon P (which may itself be a polygon created by the previous decomposition steps). The following invariants will be satisfied after each such partition.

Invariant 3.1 **(a)** *All the marked corners are on the boundaries of* $P_1, P_2, \ldots, P_k$.

(b) *The total number of type-*3 *and type-*4 *rectangles in all the partitioned polygons equal that of* P.

(c) *Marked corners on the boundary of each polygon* P_i *form a simple loop.*

(d) *The edges on each such loop (connecting adjacent vertices) are either segments of the boundary of* P *or they include a crossed vertex (fig. 4).*

(e) *Each partitioned polygon* P_i *has at least* 2 *marked vertices.*

(f) *Each concave corner of a partitioned polygon* P_i *is a concave corner of* P, *and each convex corner of* P *is a convex corner of some* P_i *(and hence all the crossed vertices are indeed concave corners of the original polygon* P*).*

Definition 3.1 *Let* n_i *be the number of type-i rectangles in a (proper) cover of a polygon* P *and let* m *be the number of crossed (concave) corners of* P. *Then, this proper cover is termed as a satisfactory cover provided* $n_0 \leq \lfloor \frac{m-2n_4-n_3-n_1}{2} \rfloor$.

Let n_{ij} be the number of type-i rectangles in a cover of the j^{th} decomposed polygon P_j and m_j be the number of crossed corners in P_j. By Definition 3.1, a proper cover for polygon P_j is *satisfactory* provided $n_{0j} \leq \lfloor \frac{m_j-2n_{4j}-n_{3j}-n_{1j}}{2} \rfloor$.

Lemma 3.2 *The union of satisfactory covers of* P_i*'s forms a satisfactory cover for* P. *Moreover, this satisfactory cover for* P *yields a performance ratio of* 2.

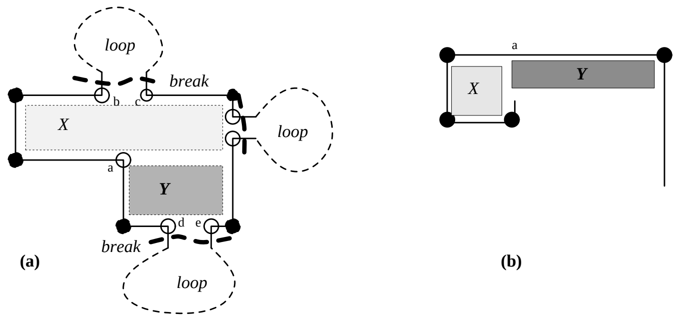

Figure 5: *(a)* **Local accounting**. *X earns 1 point and looses $\frac{1}{2}$ points at each of the corners b and c. Similarly, Y earns 1 point and looses $\frac{1}{2}$ points at each of the corners d and e. Both X and Y earn $\frac{1}{2}$ point at the corner a, but they do not loose any point there, since no additional break in the loop is created. (b) Context-sensitive accounting. Y gets back $\frac{1}{2}$ points for no break at corner a, since X overpaid it (the rectangles are shown slightly offset for clarity)*

Proof. Let n_j be the number of type-j rectangles in the union of satisfactory cover of the P_i's, and let m be the number of crossed corners of P. Hence, $n_j = \sum_{i=1}^{k} n_{ji}$ and (by Invariant 3.1(f)) $m = \sum_{i=1}^{k} m_i$.

The union of satisfactory covers is a satisfactory cover because

$$n_0 = \sum_{i=1}^{k} n_{0i} \leq \sum_{i=1}^{k} \left\lfloor \frac{m_i - 2n_{4i} - n_{3i} - n_{1i}}{2} \right\rfloor \leq \left\lfloor \frac{m - 2n_4 - n_3 - n_1}{2} \right\rfloor$$

The performance bound of 2 follows as a consequence of Lemma 3.1, and the fact that the total number of crossed vertices in all the decomposed polygons equals the number of concave corners c of the given polygon P. □

So, we will need to show that, for each polygon considered in our analysis, either we can produce a satisfactory cover for it, or we can decompose it into a number of smaller polygons each of which has a satisfactory cover. This will be done in the subsequent sections.

3.3 Credit assignment.

To assure that we construct a satisfactory cover, we will introduce a system of *credits* or *points* such that each type-i rectangle (for $1 \leq i \leq 4$) in the cover earns some number of points. Intuitively, this helps us to justify the fact that even though sometimes a cover of the convex corners of the polygon may not be able to cover many concave corners, it will not make the overall performance ratio worse.

We now describe our accounting scheme. Initially, each type-i rectangle (for $1 \leq i \leq 4$) has zero points. Two types of accounting, *local* and *context-sensitive*, are used. They are described below.

The following scheme is used in the *local* accounting of credits. By proposition 3.1 we know that two type-i (for $1 \leq i \leq 4$) rectangles do not simultaneously overlap on boundary of positive length and interior of positive area. Hence, no crossed vertex is covered by more than two such rectangles. If a crossed vertex is covered by j type-i polygons (for $1 \leq j \leq 2$), each of these rectangles earn $\frac{1}{j}$ points for this. However, covering crossed vertices in this manner sometimes may

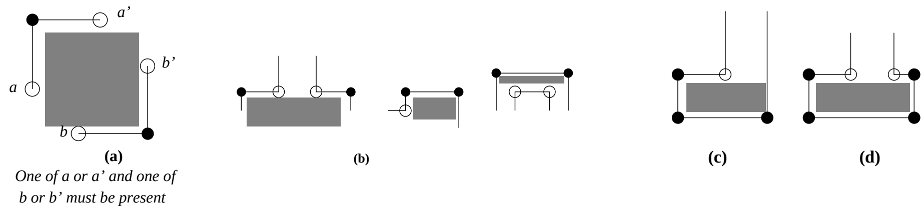

Figure 6: *(a) and (b) are not problematic, (c) and (d) are problematic cases.*

give rise to a break of the existing chain and the responsible rectangle looses 1 point for each such break. Note, however, that a break in the chain creates two new end-points for the broken chain and the responsible rectangle looses $\frac{1}{2}$ point at each end (i.e., it divides 1 point equally between the two end points in our scheme). See *fig.* 5(a) for an example.

In the *context-sensitive* accounting of points, a type-1 rectangle gets back $\frac{1}{2}$ points for not making a new break in a chain, provided that the break was already created and paid for by another rectangle (*fig.* 5(b)). This accounting will allow us to select type-1 rectangles rapidly in a simple greedy manner.

Proposition 3.3 *Including breakup of existing chains, if at all, a rectangle earns a total of* **at least** $\frac{1}{2}$ *points at each of the crossed corners it covers.*

Proof. A crossed corner can be covered by at most two rectangles in our cover. If it is covered by one rectangle R, then R gets 1 point on this corner and looses $\frac{1}{2}$ points, thereby earning a total of $\frac{1}{2}$ points. If this corner is shared by two rectangles R_1 and R_2, then there is no break of the chain at this corner, and R_1 and R_1 share 1 point equally between them. □

Proposition 3.4 *Let n_i be the number of type-i rectangles in any proper cover and assume that each type-i rectangle (for $1 \leq i \leq 4$) earns, in the credit assignment scheme described above,* **at least** *t_i points. Let a be the number of covered crossed corners, and b be the number of disjoint chains formed from the remaining uncovered crossed corners. Then, $a - b \geq \sum_{i=1}^{4} n_i.t_i$.*

Proof. This follows directly from the credit assignment scheme, since each covered crossed corner contributes $+1$ to the sum $\sum_{i=1}^{4} n_i.t_i$ and each break in chain contributes -1 to the sum $\sum_{i=1}^{4} n_i.t_i$. □

Lemma 3.3 *A proper cover of any partitioned polygon is satisfactory if $t_1 = 1$, $t_2 = 0$, $t_3 = 1$, and $t_4 = 2$.*

Proof. The proof is deferred until Section 3.5. □

We investigate further on the lower bounds on the values of t_i for $1 \leq i \leq 4$. We make the following observations.

Observation 3.5 **(a)** *If a type-4 rectangle covers more than 2 crossed corners (and, hence, at least 4 crossed corners), then $t_4 \geq 2$.*

(b) *If a type-3 rectangle covers more than 1 crossed corner (and, hence at least 3 crossed corners), then* $t_3 \geq \frac{3}{2}$.

(c) $t_2 \geq \frac{1}{2}$. *This is seen as follows. Consider a type-2 rectangle produced by convex corners which are not horizontally or vertically aligned. It will always cover at least two crossed corners, thus* $t_2 \geq 1$ *(fig. 6(a)). A type-2 rectangle produced by convex corners which are either horizontally or vertically aligned also satisfies* $t_2 \geq \frac{1}{2}$ *(fig. 6(b)).*

Since Lemma 3.3 requires $t_2 \geq 0$ *for a satisfactory cover, each type-2 rectangle carries a* **surplus** *of* $\frac{1}{2}$ *points which can be taken by another rectangle using context-sensitive accounting. In particular, we will see in Section 3.6 how to use these surplus points to choose appropriate type-1 rectangles.*

So, the remaining following three types of rectangles may be *problematic*:

(a) Type-3 rectangles covering exactly 1 crossed corner (*fig.* 6(c)).

(b) Type-4 rectangles covering exactly 2 crossed corners (*fig.* 6(d)).

(c) Type-1 rectangles if they do not earn at least 1 point.

In each of the above cases, we are below our required lower bound by 1 point.

We will show subsequently that, for each of the above problematic cases above, either the case does not arise or we can apply our corridor decomposition procedure to guarantee a satisfactory cover. In particular, the problematic cases (a) and (b) are discussed in Section 3.4, and the problematic case (c) is discussed in Section 3.6.

3.4 More Of Corridor Decomposition

To handle one or more of the problematic cases listed in the previous section, we will sometimes need to apply a procedure termed as the *corridor decomposition*. The idea is as follows. One or more of the above problematic cases create break in the loops and, unfortunately, not enough crossed vertices are covered to justify it by our accounting procedure. This decomposition procedure will make sure that we can either *repair* the break, or we have a *local surplus of credits* not claimed by any other rectangles.

We define a *corridor* as a part of the polygon which joins two portions of the polygon with the following properties. (*fig.* 4):

(a) Type-3 and type-4 rectangles do not cross the corridor.

(b) the edges joining the two parts are loop edges.

(In the figures and proofs, we show corridors **only in the vertical direction**, but all the proofs and procedures can easily be extended if the corridor is in the horizontal direction).

As a result of the above conditions we can apply the corridor decomposition procedure to partition the polygon, since we satisfy all the parts (a)-(e) of invariant 3.1 required for the permissibility of the application of any partition procedure.

Note that the problematic type-3 and type-4 rectangles define a corridor. On one side of the corridor is the type-3 or type-4 polygon; on the other side is the remaining polygon. We will later observe that some cases of problematic type-1 rectangles also define a corridor.

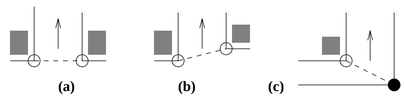

Figure 7: *Funnel and bend type corridors. (a) and (b) are funnel types, whereas (c) is the bend type. A dotted line indicates the first new edge (which repairs the lower part of the corridor) and a vertical arrow (↑) indicates direction in which we proceed for further repair. The exterior is shown shaded*

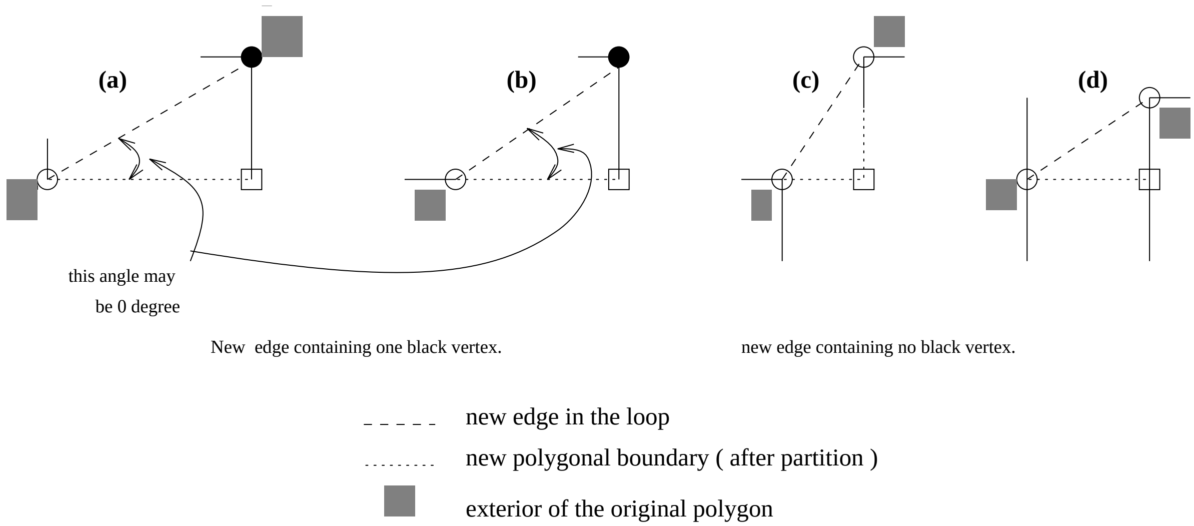

Figure 8: *New types of edges possible due to corridor repair*

The corridor decomposition procedure should be able to do the following. A break in the loop containing the corridor is created due to the covering corners on one side of the corridor (for the case of problematic type-3 or type-4 rectangles this is created due to the covering of crossed vertices by the type-3 or type-4 rectangle, respectively). The corridor *repair* procedure should be able to do one or both of the following:

(a) Rejoin the disconnected loop (which in effect means that it partitions the polygon into smaller polygons) on both sides of the corridor so that no chain is created, but the number of loops may increase. Each new loop created with 3 or more vertices must have at least 2 crossed vertices,

(b) **A local surplus** of at least 1 point is obtained which is not claimed by any other rectangle in our cover. This additional 1 point pays for the problematic rectangles which define the corridor.

In Lemma 3.9 in the next section, we will see why increasing number of loops is not harmful as long as we do not introduce any new corners in the loops due to repair.

In addition to maintaining the invariants, the corridor decomposition procedure will also satisfy the following conditions:

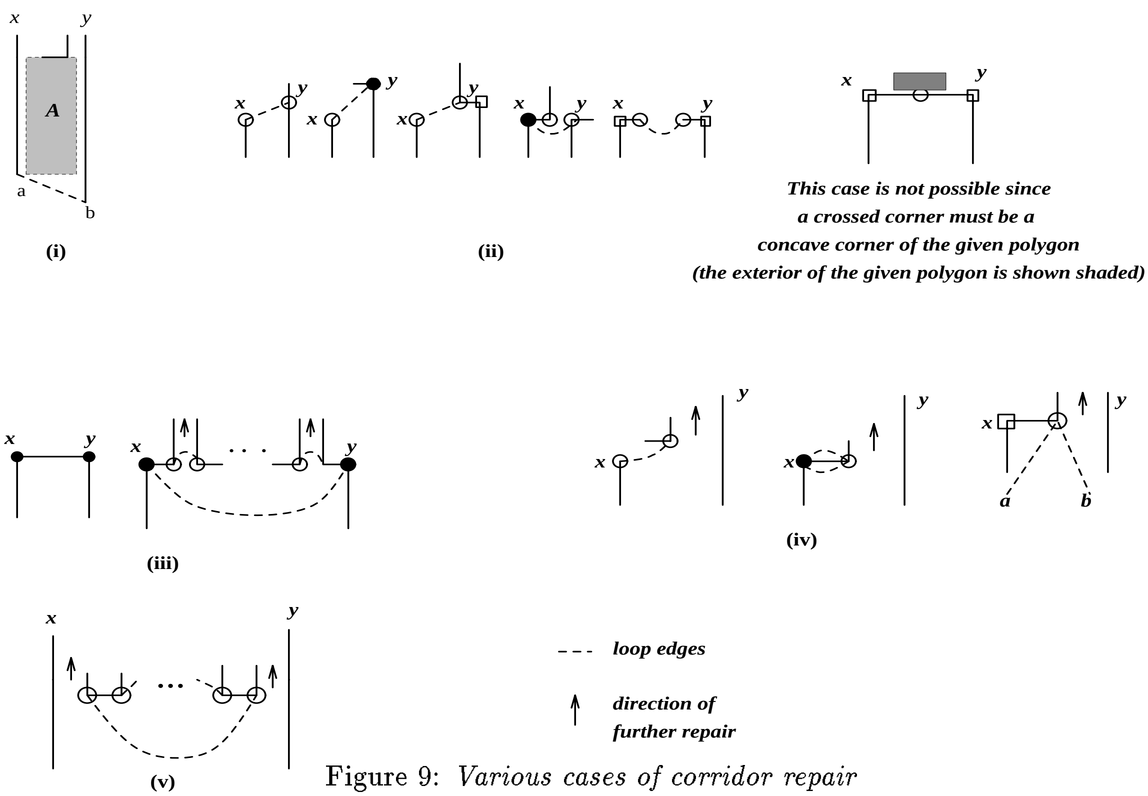

Figure 9: *Various cases of corridor repair*

(I) The corridor procedure is invoked in only the three types of situations as shown in *fig.* 7. The first two of them are termed as the *funnel* type (*fig.* 7(a),(b)), and the third type as the *bend* type (*fig.* 7(c)).

(II) Whenever a corridor creates a new edge in a loop which may be subject to further decomposition, this edge contains at least one crossed vertex. New edges containing one black and one crossed vertex are restricted to the types as shown in *fig.* 8(a),(b), and new edges containing two crossed vertices are restricted to the types as shown in *fig.* 8(c),(d). The corners and the position of the exterior of the original polygon is shown shaded in *fig.* 8.

Note that the problematic type-3 and type-4 rectangles define a corridor of funnel type. Problematic type-1 rectangles may sometimes define a bend type of corridor. Also, we will *always* apply a corridor procedure for the bend type corridor, whenever such a corridor is found.

Lemma 3.4 *It is always possible to have a repair procedure for a corridor, and this procedure, if necessary, creates new edges which satisfy property (II) above.*

Proof. Refer to *fig.* 9. We repair part of the corridor in the following manner:

- If the corridor is of funnel type, join the loop via the edge ab. This repairs the lower portion of the corridor.
- If the corridor is of bend type, create a new loop containing a and b. Hence, two sides of the corridor still need to be repaired.

Due to symmetry, it is sufficient to show how to repair the corridor on one side, since, for bend type corridors, a similar procedure can be applied to the other side of the corridor. We push up the rectangle A having the two sides of the corridor as its two vertical sides until it is stopped by some side of the polygon (*fig.* 9(i)). Let x and y be the next two corners of the polygon following a and b, respectively. Depending upon the relative positions and types of a and b, we have the following cases:

Case 1. Both x and y are on the boundary of A.

Case 1.1. At least one of x and y is not black. Depending upon the types of x and y, we have the various cases and corresponding repairs needed as shown in *fig.* 9(ii). We know that covering each crossed corner gives $\frac{1}{2}$ point to each rectangle, and if just one rectangle covers a crossed corner, it gets 1 point for this. Hence, it is not difficult to see that a surplus of 1 point is obtained in each case of repair.

Case 1.2. Both of x and y are black. We create a new loop of black and cross corners, and continue with further application of the corridor procedure as shown in *fig.* 9(iii). Notice that, as per our requirement, this new loop with 3 or more corners has at least 2 crossed corners.

Case 2. x is on the boundary of A, but y is not. The various cases and repairs needed are shown in *fig.* 9(iv). Again, a surplus of 1 point is obtained in each case.

Case 3. y is on the boundary of A, but x is not. Similar to case 2 above.

Case 4. None of x or y are on the boundary of A. The only case and needed repair is shown in *fig.* 9(v). □

Remark 3.1 *Assume that we are given a polygon in the form a loop of k corners. Then, first we can tentatively place all the type-3 and type-4 rectangles in $O(k)$ time using Proposition 3.2 and determine all problematic funnel type corridors. Now, it is easy to design an algorithm running in $O(k^2)$ time decomposing all bend type and problematic funnel type corridors. (our algorithm must repair all bend type corridors first before repairing other types of corridors).*

Proposition 3.5 *After any necessary corridor decompositions, each type-3 and type-4 rectangles earn at least 1 point.*

Proof. Follows directly from Lemma 3.4. □.

3.5 Covering chains and loops.

In this section we will give an estimate of the maximum number of rectangles needed to cover chains and loops. We define a *simple* loop (resp. *simple* chain) as a loop (resp. chain) such that no two of its edges cross each other. We need the following important property for chains and loops.

Lemma 3.5 *All the chains and loops constructed by our algorithm will be simple.*

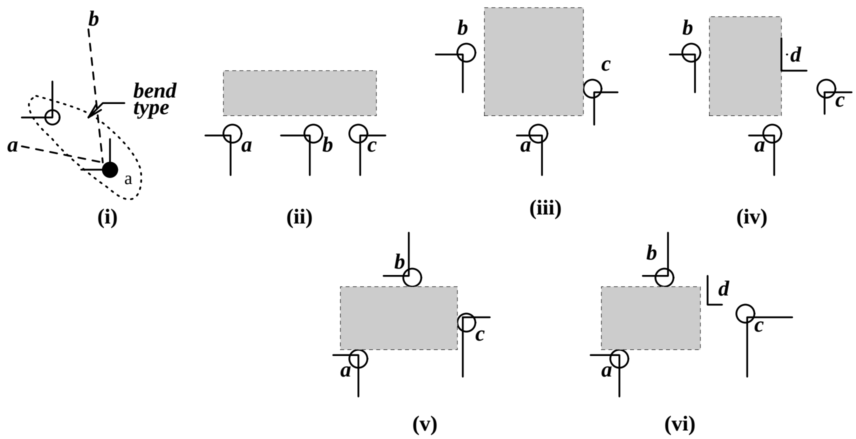

Figure 10: *Finding* 3 *corners in a loop.*

Proof. We obviously satisfy this property at the beginning before our heuristic starts. Breaking an existing chain or loop can never violate this property. We will sometimes need to repair a chain or loop by the *corridor repair* procedure described in the previous section. However, an examination of this procedure (proof of Lemma 3.4) clearly shows that we never violate this property by repairing the corridor. □

Lemma 3.6 *Let A be the number of disjoint chains, and B be the total number of corners in them. Then, we can cover all the vertices in the chains with at most $\lfloor \frac{A+B}{2} \rfloor$ rectangles in $O(B)$ time.*

Proof. A single chain can be covered greedily by covering sequences of pairs of adjacent corners in the chain, except possibly the last corner if there are odd number or corners. Let there be x corners in a single chain. If x is odd, the cover needs $\frac{x+1}{2}$ rectangles. If x is even, the cover needs $\frac{x}{2}$ rectangles. In other words, in either case, the chain needs $\lfloor \frac{x+1}{2} \rfloor$ rectangles.

Let $x_1, x_2, \ldots, x_A$ be the number of corners in the A disjoint chains ($x_1 + x_2 + \ldots + x_A = B$). The total number of rectangles needed are at most $\sum_{i=1}^{A} \lfloor \frac{x_i+1}{2} \rfloor \leq \lfloor \frac{A+B}{2} \rfloor$ □

Corollary 3.7 *A single chain of k corners can be covered with $\frac{k}{2}$ rectangles if k is even.*

We can do somewhat better while covering loops, which is *very crucial* for the analysis of the performance of our heuristic (as noted by the corridor repair procedure). For this we need the following lemma.

Lemma 3.8 *In a simple loop of* 3 *or more corners, we can always find a set of* 3 *corners which can be covered together by a single rectangle.*

Proof. Let a be a corner in the loop with the *minimum* y-coordinate. Let b and c be the neighbors of a in the loop (*fig.* 10). If a, b and c can be covered together by a rectangle, we are already done. So, assume that this is not the case. Now, the proof involves a detailed case analysis.

Case 1. One of a, b or c, say a, is a black vertex. Since a, b and c cannot be covered together, there must be a bend type corridor (*fig.* 10(i)), and we need to do corridor decomposition.

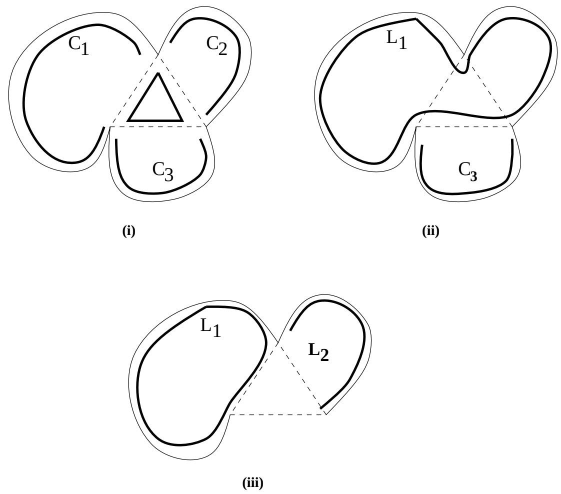

Figure 11: *Covering a loop. k_i is the number of corners in chain C_i. (i) k_1, k_2, k_3 all even. (ii) k_1 and k_2 odd, $k_3 > 0$ even. (iii) k_1 and k_2 odd, $k_3 = 0$.*

Case 2. All of a, b and c are crossed corners.

Case 2.1. a, b and c are collinear. We are done (*fig.* 10(ii)).

Case 2.2. a, b and c are not collinear.

Case 2.2.1. b is vertically above a.

Case 2.2.1.1. b is to the left of a.

Case 2.2.1.1.1. a, b and c can be covered by a rectangle (*fig.* 10(iii)). We are done.

Case 2.2.1.1.2. Otherwise, let d be the leftmost corner on the right of a which prevents this from happening (*fig.* 10(iv)). Assume that d is not a corner of this loop. Since the polygon has **no holes**, there has to be a loop edge passing under d, but this violates the simplicity of the loop. Hence, d must be on the loop, and we can cover a, b and d by a rectangle.

Case 2.2.1.2. b is to the left of a. Similar argument as in case 2.1.1 (*fig.* 10(v),(vi)).

Case 2.2.2. c is vertically above a. Similar to case 2.1. □

Remark 3.2 *If the loop has k vertices, then the set of such 3 vertices needed for the above lemma can be found trivially in $O(k)$ time.*

Lemma 3.9 *Let X be the total number of corners in all the loops in a proper cover of a polygon. Then, we can cover the corners in all the loops with at most $\lfloor \frac{X}{2} \rfloor$ rectangles.*

Proof. First, consider the case when there is just one loop L. Let k be the number of corners in L. We will prove the lemma for a single loop by induction on k. We know that $k > 1$, because we always maintain Invariant 3.1(e).

Basis: For $k = 2$, the claim is trivially true. For $k = 3$, the claim follows directly from Lemma 3.8. *Inductive Step:* Let $k > 3$ and assume that the claim holds for all loops with at least 2 and at most $k - 1$ corners. We have the following cases.

Case 1. k is even. The claim follows from the Corollary 3.7.

Case 2. k is odd. By Lemma 3.8, we can find 3 corners which can be covered by a single rectangle. Let k_1, k_2 and k_3 be the number of corners of L, *excluding the corners on this triangle*, which are on the three chains C_1, C_2 and C_3, respectively, passing through the three sides of this triangle (*fig.* 11). Note that $k = k_1 + k_2 + k_3 + 3$. Since k is odd, the only possibilities are either none or exactly two of k_1, k_2, k_3 are odd.

Case 2.1. None of k_1, k_2 and k_3 is odd. Partition the loop into three chains C_1, C_2 and C_3 each having even number of corners, and cover the remaining 3 corners on the triangle by a single rectangle (*fig.* 11(i)). By Corollary 3.7, the total number of rectangles needed is $\frac{k_1+k_2+k_3}{2} + 1 = \frac{k-1}{2} = \lfloor \frac{k}{2} \rfloor$.

Case 2.2. Two of k_1, k_2 and k_3, say k_1 and k_2, are odd. There are now two cases.

If $k_3 > 0$, then we partition the loop L into chain C_3 with an even number $k_3 > 0$ of corners, and a loop L_1 with odd number $3 < k - k_3 < k$ of corners (*fig.* 11(ii)). By Corollary 3.7, C_3 can be covered with $\frac{k_3}{2}$ rectangles. By induction hypothesis, the loop L_1 can be covered by $\lfloor \frac{k-k_3}{2} \rfloor$ rectangles. Hence, we use $\frac{k_3}{2} + \lfloor \frac{k-k_3}{2} \rfloor = \lfloor \frac{k}{2} \rfloor$ rectangles.

Otherwise, $k_3 = 0$. Then, we partition L to two parts:

- A chain L_2 consisting of the corners on C_2 and one corner of the triangle (see *fig.* 11(iii)). L_2 has $k_2 + 1$ corners. Since $k_2 + 1$ is even, by Corollary 3.7, L_2 can be covered by $\frac{k_2+1}{2}$ rectangles.
- A loop L_1 consisting of the remaining $3 \leq k_1 + 2 < k$ corners. By induction hypothesis, the loop L_1 can be covered by $\lfloor \frac{k_1+2}{2} \rfloor$ rectangles.

Hence, we use $\frac{k_2+1}{2} + \lfloor \frac{k_1+2}{2} \rfloor = \lfloor \frac{k}{2} \rfloor$ rectangles.

Finally, consider the case when we have $p > 1$ loops $L_1, L_2, \ldots, L_p$. Let k_i be the number of corners in loop L_i $(k_1 + k_2 + \ldots + k_p = X)$. Each loop L_i can be covered by $\lfloor \frac{k_i}{2} \rfloor$ rectangles. Hence, the total number of rectangles needed is $\sum_{i=1}^{p} \lfloor \frac{k_i}{2} \rfloor \leq \lfloor \frac{X}{2} \rfloor$. □

Remark 3.3 *Notice that the number of rectangles needed to cover all the loops is independent of the number of loops (unlike chains).*

Remark 3.4 *A loop of k vertices can be covered as outlined in the above lemma in $O(k^2)$ time. The idea is follows. If k is even, the loop can be covered in $O(k)$ time using Lemma 3.6. Otherwise, we can find, in $O(k)$ time, the three vertices of the loop that can be covered together and decompose the loop into at most three disjoint loops (excluding these* 3 *vertices). We need to repeat such a decomposition process at most $O(k)$ times.*

Remark 3.3 states that breaking a loop into two or more loops does not make the performance ratio worse as long as the total number of corners in the partitioned loops *do not increase*. However, the same does not hold for chains due to Lemma 3.6. Hence, as we noted in the previous section, the corridor decomposition procedure is not allowed to break chains, but it may break a loop into two or more smaller loops.
Finally, we are ready to give a proof of Lemma 3.3.

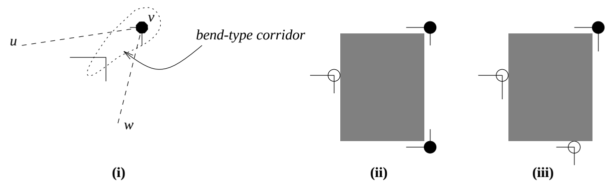

Figure 12: *Appropriate selection of a type-*1 *rectangle.*

Proof of Lemma 3.3 of Section 3.3. Let t be the total number of crossed corners, a be the number of covered crossed corners, d be the number of uncovered crossed corners, $0 \leq e \leq d$ be the number of uncovered crossed corners in all the loops, and b be the number of disjoint chains produced ($t = a + d$). By Proposition 3.4, $b + d = t - (a - b) \leq t - n_1 - n_3 - 2n_4$. By Lemma 3.6, we need at most $\lfloor \frac{b+(d-e)}{2} \rfloor$ type-0 rectangles to cover the uncovered corners in the chains. By Lemma 3.9, we need at most $\lfloor \frac{e}{2} \rfloor$ type-0 rectangles to cover the uncovered corners in all the loops. Hence, the total number n_0 of type-0 rectangles satisfy

$$n_0 \leq \left\lfloor \frac{b + (d - e)}{2} \right\rfloor + \left\lfloor \frac{e}{2} \right\rfloor \leq \left\lfloor \frac{b + d}{2} \right\rfloor \leq \left\lfloor \frac{t - n_1 - n_3 - 2n_4}{2} \right\rfloor$$

This value of n_0 satisfies Definition 3.1 of a satisfactory cover. □

3.6 Selecting Appropriate Type-1 Rectangles

The only major part that still remains is to show how we can choose type-1 rectangles appropriately so that either each of them earns *at least* 1 point, or they define a corridor which allows the repair procedure to work. For this purpose, we will sometime use our context-sensitive accounting as defined in Section 3.3 to enable us to get *extra points.*

Lemma 3.10 *Assume that we are given a polygon P as a loop of crossed and black corners* [2]. *Assume also that all the type-*3 *and type-*4 *rectangles and an arbitrary collection of type-*2 *rectangles*[3], *corresponding to some proper cover, have been placed. Furthermore, assume that P has no bend type corridors*[4]. *Then, it is possible to select type-*1 *rectangles appropriately such that each of them earns at least* 1 *point.*

Proof. Consider selecting a particular type-1 rectangle. Let the type-1 rectangle be defined by the black corner v. Therefore, v is a black vertex on a loop of size at least 2 and v is not covered as yet. If v is in a loop of exactly 2 vertices including v, we are already done. This is because if the other vertex w is a crossed vertex, then the type-1 rectangle covering v and w gets 1 point, since it does not have to pay for any break (and hence gets 0.5 point by context-sensitive accounting), and covers w (and hence gets 0.5 point by local accounting). If w is a black vertex, then this

[2] This polygon may be the result of partitioning a larger polygon. However, note that blank vertices (corresponding to additional convex corners introduced by partitioning) need not be covered and hence are never part of the loop. Also, note that the loop has at least 2 corners by Invariant 3.1(e).

[3] As noted in Observation 3.5(c), a type-2 rectangle, however placed, always earns $\frac{1}{2}$ points and hence always has a surplus of $\frac{1}{2}$ points.

[4] In other words, P has no bend type corridors, and since the given cover is satisfactory, all problematic funnel type corridors have been decomposed.

type-2 rectangle covering v and w earns 0 point, since it does not have to pay for any break (and, 0 points are sufficient for type-2 rectangles). So, assume that v has 2 neighbors in the loop. Let the neighbors be u and w.

Case 1. Both of u and w are black. Then, since u, v and w cannot be covered together, there must be a bend type corridor not decomposed (*fig.* 12(i)). This is a contradiction.

Case 2. One of u and w, say w, is black.

Case 2.1. u, v and w can be covered together. Then, there are three cases to consider.

Case 2.1.1. w is not covered by any other rectangle. We have a type-2 rectangle covering w and v which earns 0 point (*fig.* 12(ii)).

Case 2.1.2. w is covered by some other type-2 rectangle R. Then, the type-1 rectangle covering u and v gets $\frac{1}{2}$ points from R (by context-sensitive accounting and Observation 3.5(c)) and gets $\frac{1}{2}$ points for covering u.

Case 2.1.3. w is covered by some other type-3 or type-4 rectangle R. R cannot cover u and hence the type-1 rectangle covering u and v gets 1 point for covering u.

Case 2.2. u, v and w cannot be covered together. Then, we must have a bend type corridor not decomposed (*fig.* 12(i)). This is a contradiction.

Case 3. Both u and w are crossed vertices.

Case 3.1. u, v and w can be covered together. Then, we have a type-1 rectangle which earns 1 point by local accounting (*fig.* 12(iii)).

Case 3.2. u, v and w cannot be covered together. Then, we must have a bend type corridor not decomposed (*fig.* 12(i)). This is a contradiction. □

Remark 3.5 *Assuming the loop is given as a sequence of its vertices, obviously each type-1 rectangle can be found greedily as outlined in the above lemma in $O(1)$ time by traversing the corners of the loops in order.*

3.7 The Heuristic, Its Running Time And Performance Ratio

We are now all set to describe the heuristic in details.

INPUT : A rectilinear polygon P without holes.

OUTPUT: An approximate cover of the corners of P.

ALGORITHM:

0. Preprocess P such that every corner has coordinates (x, y) with $x, y \in \{1, 2, \ldots, n\}$.

1. Let L be a single loop of corners representing P. Repair all bend type corridors of L (using the proof of Lemma 3.4). Let $L_1, L_2, \ldots, L_p$ be the disjoint loops resulting from L.

2. **Tentatively** place all the type-3 and type-4 rectangles. Note all the corridors defined by problematic type-3 and type-4 rectangles (Section 3.3 and *fig.* 6). Now, repair all these (funnel type) corridors (using the proof of Lemma 3.4). Let $L_1, L_2, \ldots, L_q$ be the new set of disjoint loops (representing the partitioned polygons).

3. **For** $i = 1, 2, \ldots, q$ **do**

   (a) Place all type-3 and type-4 rectangles in a proper cover of L_i (Proposition 3.2).

   (b) Cover the remaining uncovered black and crossed corners of L_i, represented as a set of chains and loops, greedily as outlined in Lemma 3.6 and Lemma 3.9 (note that each type-1 rectangle will always gain at least 1 point by Lemma 3.10, since we have already applied corridor decomposition procedure for bend type corridors).

   Endfor

Lemma 3.11 *The heuristic runs in $O(n^2)$ time.*

Proof. Step 0 takes $O(n \log n)$ time. Step 1 can be implemented in $O(n^2)$ time (Remark 3.1). In Step 2, type-3 and type-4 rectangles can be placed tentatively in $O(n)$ time by Proposition 3.2, and we can repair all the funnel type corridors in $O(n^2)$ time (Remark 3.1). Since the loops $L_1, L_2, \ldots, L_q$ are disjoint and there are in all $O(n)$ corners in all of them, Step 3(a) takes $O(n)$ time by Proposition 3.2, Step 3(b) takes $O(n^2)$ by Lemma 3.6, Remark 3.4 and Remark 3.5. □

Finally, we mention how to handle the case of degenerate convex corners, and still have a performance ratio of 2.

Theorem 3.1 *The heuristic uses at most $2\theta - 2$ rectangles, when θ is the optimal cover size.*

Proof. If the polygon has no degenerate convex corners this follows already from the fact that we have a satisfactory cover. Otherwise, assume that the polygon has β degenerate convex corners. We partition the polygon into $\beta + 1$ disjoint polygons, and apply our heuristic on each of them. Let θ_i be the size of optimal cover of the i^{th} polygon. Then, since we attain a satisfactory cover for this polygon, we use at most $2\theta_i - 2$ rectangles for this polygon. The optimal cover size θ for the entire polygon must use at least $\sum_{i=1}^{\beta} \theta_i - (\beta - 1)$ rectangles. We use at most $2\sum_{i=1}^{\beta} \theta_i - 2\beta$ rectangles. The tightness of the performance ratio follows from the example shown in *fig.* 13. □

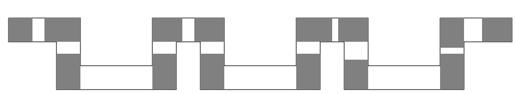

Figure 13: *Worst case example for the heuristic for corner cover without holes. The polygon has some odd number k of bends (shown for $k = 13$). The optimal cover uses $\frac{k+1}{2}$ rectangles whereas the heuristic uses $k - 1$ rectangles. The rectangles used by the heuristic are shown shaded*

Remark 3.6 *It is possible to modify the heuristic so that the worst case example as shown in fig. 13 is solved optimally. However, this does not guarantee an improvement of the worst case performance ratio of our heuristic.*

4 Conclusion and Open Problems.

We have proposed efficient heuristic for the corner cover problems for rectilinear polygons without holes. Currently, to the best of our knowledge, it is still open if one could design an approximation algorithm with constant performance ratio for the interior cover problem. Franzblau[5] proposes a sweep-line heuristic that guarantees a constant performance ratio if the polygon has no holes, but the upper bound for the performance ratio proved there is $O(\log\theta)$ (where θ is the optimal cover size) when the polygon has holes. The following problems still remain open and may be worth investigating further:

- Can we generalize this heuristic to work with the same performance ratio when the given polygon may have holes?
- Can we prove a better upper bound of the performance ratio for the sweep-line heuristic for the interior cover problem when the given polygon may have holes?

References

[1] S. Chaiken, D. J. Kleitman, M. Saks and J. Shearer. Covering Regions by Rectangles. *SIAM J. Algebraic Discrete methods*, 2 (1981), pp. 394-410.

[2] H. E. Conn, and J. O'Rourke. Some restricted rectangle covering problems. Tech Rep. JHU-87/13, The Johns Hopkins University, Baltimore, MD, 1987, also appeared in *Proc. of the 1987 Allerton Conference* (1987), pp. 898-907.

[3] J. C. Culberson and R. A. Reckhow. Covering Polygon is Hard. *Journal of Algorithms,* 17 (1994), pp. 2-44.

[4] P. Berman and B. DasGupta. On the Complexities of Efficient Solutions of the Rectilinear Polygon Cover Problems. *Algorithmica*, 17 (1997), pp. 331-356.

[5] D. S. Franzblau. Performance Guarantees on a Sweep-line Heuristic for Covering Rectilinear Polygons with Rectangles. *SIAM J. Disc. Math.*, 2 (1989), pp. 307-321.

[6] D. S. Franzblau and D. J. Kleitman. An Algorithm for Covering Polygons with Rectangles. *Information and Control*, Vol. 63 No. 3 (1984), pp. 164-189.

[7] A. Lubiw. Ordering and some Combinatorial Optimization Problems with some Geometric Applications. Ph. D. Thesis, University of Waterloo, Ontario, Canada, 1985.

[8] A. Lubiw. The Boolean Basis Problem and how to Cover Polygons by Rectangles. University of Waterloo, Ontario, Canada, Manuscript, 1988. Referenced in [5].

[9] W. J. Masek. Some NP-complete Set Covering Problems. MIT, Cambridge, MA, unpublished manuscript. Referenced in M. R. Garey and D. S. Johnson. *Computers and Intractability - A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., 1979.

[10] C. Mead and L. Conway. *Introduction to VLSI Systems.* Addision-Wesley, Reading, MA., 1980, Chapters 2 and 4.

[11] R. Motwani, A. Raghunathan and H. Saran. Perfect Graphs and Orthogonally Convex Covers. *SIAM Journal on Discrete Mathematics,* 2 (1989), pp. 371-392.

[12] T. Ohtsuki. Minimum Dissection of Rectilinear Polygons. *Proc. IEEE Symp. on Circuits and Systems* (1982), pp. 1210-1213.

[13] J. O'Rourke. *The Art Gallery Theorem.* Oxford University Press, 1987.

[14] J. O'Rourke and K. J. Supowit. Some NP-hard Polygon Decomposition Problems. *IEEE Trans. Inform. Theory*, 29 (1983), pp. 181-190.

[15] L. Pagli, E. Lodi, F. Luccio, C. Mugnai and W. Lipski. On Two Dimensional Data Organization 2. *Fund. Inform.*, 2 (1979), pp. 211-226.