Complexities of Efficient Solutions of Rectilinear Polygon Cover Problems[1]

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
University of Minnesota
Minneapolis, MN 55455-0159
email: dasgupta@cs.umn.edu

December 7, 1994

[1]A preliminary version of this result appeared in fourth Canadian Conference on Computational Geometry(1992)

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, we consider covering the entire interior, the boundary and the set of corners of the polygon. These problems have important applications in storing images and in the manufacture of integrated circuits. Unfortunately, most of these problems are known to be NP-complete. Hence it is necessary to develop efficient heuristics for these problems or to show that the design of efficient heuristics is impossible. In this paper we show:

(a) The corner cover problem is NP-complete.

(b) The boundary and the corner cover problem can be approximated within a ratio of 4 of the optimum in $O(n \log n)$ and $O(n^{1.5})$ time, respectively.

(c) No polynomial-time approximation scheme exists for the interior and the boundary cover problems, unless $P = NP$.

Key words

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

MAILING ADDRESS OF CORRESPONDING AUTHOR:

Bhaskar DasGupta
Department of Computer Science
4-192 EE/CSci Building
200 Union Street S.E.
University of Minnesota
Minneapolis, MN 55455-0159

1 Introduction.

In this paper we consider the rectilinear polygon cover problems, *i.e.*, the problem of covering 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 [18]. Thus, the rectilinear polygon cover problems have received particular attention. These problems are also interesting because of their application in storing images [14], and in the manufacture of integrated circuits [15]. Also, an investigation of these problems has given rise to special kinds of perfect graphs of interest [16].

Masek [14] was the first to show that the interior cover problem is NP-complete for rectilinear polygons with holes. Conn and O'Rourke [4] 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 [6] 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.

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 [9] 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. [3]. Lubiw [12, 13] gave polynomial time algorithm for the interior cover problem for a somewhat larger class of polygons, called the plaid polygons. Conn and O'Rourke[4] 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 [8] 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 [17, 19].

In this paper, we address the interior, boundary as well as the corner cover problems. These problems have important applications as mentioned before, hence we need to know about the complexities. Moreover, the motivation behind the study of the boundary or corner cover problems has been to better understand the covering problems by further exploring the boundary between those problems which can be exactly solved or efficiently approximated and those for which such exact or approximate solutions are unknown. [4, 6]. For example, although there is currently no heuristic known to us that solves the interior cover problem with constant performance ratio, we show that there exists polynomial-time algorithms with constant performance ratio for both the boundary and corner cover problems. This may suggest that even though all these problems are NP-complete, there may be differences in the complexities of efficient approximation algorithms of these problems.

The rest of the paper is organized as follows:

- In section 2 we state some basic definitions needed to study these cover problems. We also state the definitions of L-reductions, polynomial-time approximation schemes and the relationship between them.
- In section 3 we prove that the corner cover problem for rectilinear polygons is NP-complete.
- In section 4 we propose and analyze efficient heuristics for the boundary and corner cover problems for rectilinear polygons.
- In section 5 we prove that no polynomial-time approximation scheme is possible for the interior and boundary cover problems unless P=NP.
- We conclude in section 6 with some open problems which may be worth investigating further.

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. 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[4] 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 [3] the optimal corner cover size is 7 but the optimal boundary or interior cover size is 8, and in fig. 6 of [3] 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* bound between the relative sizes of these three types of covers in general.

An *anti-rectangle* set is a set of points inside the given rectilinear polygon such that no two of them can be covered together by a rectangle which does not contain a part of the exterior of the polygon (see *fig.* 1 for an example). Depending upon whether it is an interior, boundary or corner cover problem, these points can be placed only in the interior, boundary or corners of the given polygon, respectively. If θ is the size of a cover for one of these cover problems, and α is the size of an anti-rectangle set for this cover, then it is obvious that $\theta \geq \alpha$. When the cover size is minimum and the size of the anti-rectangle set is maximum, the equality $\theta = \alpha$ holds for some special cases of the cover problems. However, the equality is *not true* in general for either the interior, boundary or corner cover problems [3]. Erdös asked if the ratio θ/α is bounded for interior cover problem for arbitrary rectilinear polygons (mentioned by Chaiken et. al. [3]) and the answer is not known yet (the best known bound is $\frac{\theta}{\alpha} = O(\log \alpha)$ [8]).

Sometimes during the discussion of proofs in sections 3 and 5, we encounter rectilinear polygons which may have discontinuities in its outer boundary. We will sometime refer to these types of polygons as *open polygons*. Also, sometimes we will refer to these discontinuities as the *mouth* of the corresponding open polygon. Open polygons are very useful in the sense that they can be joined to each other through the discontinuities in their boundaries to form a single polygon whose boundary has no discontinuities.

In all of our discussions later, we assume that the given polygon has no degenerate convex vertex. This is not a problem for the interior and boundary cover problems, since otherwise the given problem can always be subdivided into two or more independent subproblems. The heuristic which we propose for the corner cover problem works with the same performance ratio even if degenerate convex vertices are allowed.

2.1 L-reductions and Polynomial-Time Approximation Schemes

For any optimization problem A let c_{opt} be the cost of the optimal solution and c_{approx} be the cost of an approximate solution produced by a heuristic. Let $\epsilon_n = \frac{|c_{opt} - c_{approx}|}{c_{opt}}$ be the relative error of the approximate solution for any input of size n. A *polynomial-time approximation scheme* (PTAS) for A is an algorithm that takes as input an instance of the problem and a constant $\epsilon > 0$, and produces a solution with error $\epsilon_n \leq \epsilon$ in time polynomial in n [5]. A more detailed discussion of the related concepts is available in [5, 10].

To assist in our discussion below, we formally state the definition of the *d-bounded-degree vertex cover* problem (abbreviated as the VC_d problem) as follows:

INSTANCE: An undirected connected graph $G = (V, E)$ in which every vertex has degree at most d for some constant integer d (and, hence, $|V| - 1 \leq |E| \leq \frac{d}{2}|V|$).

QUESTION: Find a subset V_{opt} of V of minimum cardinality such that such that for every edge $\{u, v\} \in E$ either $u \in V_{opt}$ or $v \in V_{opt}$ (note that, $c_1|V| \leq |V_{opt}| \leq c_2|V|$ for two positive constants c_1 and c_2).

It is well-known that the VC_d problem is NP-complete for $d \geq 4$ even if the graph is planar [10]. However, this does not exclude the possibility of a PTAS for the VC_d problem. For this purpose, we need two more related results as follows.

The class of *MAX-SNP* problems was defined by Papadimitriou and Yannakakis [20]. They also defined the concept of an *L-reduction* between two optimization problems. The definition stated below is a slightly modified but equivalent version of [20] (similar to the one used by Berman and

Schnitger [2]), and since in this paper we are concerned with minimization problems, we state the definition for minimization problems only.

Definition 2.1 *[2, 20] Let Π and Π' be two minimization problems. Then, Π L-reduces to Π' if there are three polynomial-time algorithms T_1, T_2, T_3 and two constants $\alpha, \beta > 0$ such that*

($\star$) *For each instance I of Π, algorithm T_1 produces an instance $I' = f(I)$ of Π', such that the optima of I and I', $OPT(I)$ and $OPT(I')$, respectively, satisfy $OPT(I') \leq \alpha OPT(I)$.*

($\star\star$) *Given any solution of an instance I' of Π' with cost c', algorithm T_2 produces another solution $c'' \leq c'$ of I', and algorithm T_3 produces a solution I of Π with cost c (possibly from the solution produced by T_2) satisfying $(c - OPT(I)) \leq \beta(c'' - OPT(I'))$.*

A minimization problem is MAX-SNP-complete if it is in MAX-SNP and any other problem in MAX-SNP can be L-reduced to this problem. The following result is proved in [20].

Theorem 2.1 *[20] The VC_d problem is MAX-SNP-complete for all constants $d \geq 4$.*

The importance of the class of MAX-SNP-complete problems comes from the following result proved in [1].

Theorem 2.2 *[1] No MAX-SNP-complete problem has a PTAS unless P=NP.*

Hence, from the above theorem it follows that to prove that the interior or the boundary cover problems has no PTAS unless P=NP, it is sufficient to show how to L-reduce the VC_d problem to the interior or the boundary cover problem. Note that the impossibility of a PTAS also imply that there exists a positive constant c such that no polynomial-time algorithm for these problems can have performance ratio better than c (unless, of course, P=NP).

3 NP-completeness of the Corner Cover Problem.

Conn and O'Rourke [4] proved a version of the corner cover problem to be NP-complete. We call this as the "notch cover" problem. In this problem, they require each concave corner to be covered by two rectangles along the two perimeter segments of the polygon defining the concave corner.

The idea of their proof is as follows. They transform the satisfiability problem to the "notch cover" problem. An arbitrary Boolean expression is simulated by a rectilinear polygon. The polygon is constructed from component polygons (which are open polygons with specific properties) that can be covered efficiently if and only if they perform like their Boolean expression components. A "true" value is represented by a 2×1 rectangle and a "false" value is represented by a 1×2 rectangle (see fig. 2(a)). If a rectangle is shared by two component polygons, it is counted as $\frac{1}{2}$ rectangle for each of the two component polygons (see *fig.* 2(b)).

In their proof, each component polygon may have discontinuities in its boundaries through which truth values are propagated (by joining different component polygons). Two types of discontinuities are distinguished: *input* discontinuities, which correspond to truth values of variables propagated into the polygon, and *output* discontinuities, which correspond to truth values propagated outside of the polygon. If a rectangle extends outside the interior of the polygon through a discontinuity of its boundary, the corresponding truth value of that discontinuity is TRUE; otherwise the corresponding truth value of that discontinuity is FALSE (see fig. 2(b)). For brevity, we refer to these discontinuities simply as *input* and *output*.

In particular, from their construction it follows that to prove the corner cover problem to be NP-complete, it is sufficient to show how to construct the following types of component open polygons in polynomial time:

(a) A length k "wire" open polygon which has one input, one output and which can be corner covered with k rectangles if and only if its input and output have the same truth values.

(b) A "not" open polygon which has one input and one output and which can be corner covered with $5\frac{1}{2}$ rectangles (resp. 6 rectangles) if and only if the truth values at its input and output are complimentary (resp. same).

(c) A "crossover" open polygon which has two inputs I_1 and I_2, two outputs O_1 and O_2 and which can be corner covered with 9 rectangles if and only if the truth values of I_1 and O_2 and the truth values of I_2 and O_1 are identical; otherwise it requires more than 9 rectangles for covering.

(d) A "switchback" open polygon with one input and one output which can be corner covered with 7 rectangles if and only if the truth values of the input and the output are identical; otherwise it requires more than 9 rectangles for covering.

(e) A "true" ("false") open polygon with one input which can be corner covered with 4 (resp. $3\frac{1}{2}$) rectangles if and only if the input has a truth value of TRUE (resp. FALSE); otherwise it requires more than 4 (resp. $3\frac{1}{2}$) rectangles for covering.

(f) A "generator" open polygon with one input and three inputs which can be corner covered with c_1 rectangles if and only if all its inputs and outputs have the same truth value; otherwise it requires more than c_1 rectangles for covering (here, c_1 is a positive constant).

(g) An "and/or" open polygon with two inputs and two outputs such that the polygon can be corner covered with c_2 rectangles provided the truth value of one output is the logical AND of the truth values of the two inputs and the truth value of the other output is the logical OR of the truth values of the two inputs; otherwise it requires more than c_2 rectangles for covering (here, c_2 is a positive constant).

Conn and O'Rourke shows how to construct component polygons of types (a)-(f) (with $c_1 = 11$) for the "notch cover" problem. An examination of their component figures (*fig.* 4 and 8 of Conn and O'Rourke[4]) shows that the *wire*, *not*, *true*, *false*, *crossover*, and *switchback* component polygons for their case work for us also, since optimal notch covers of these component polygons for various truth values depend on appropriate matching of the convex corners and hence correspond to optimal corner covers as well. However, their *generator* and *and/or* components fail to satisfy the requirements for corner cover. For example, in their generator component if rectangle A is used to cover the convex corner a then rectangle B is unnecessary, since the concave corner b is already covered from one side (*fig.* 2(c)).

The idea for modification of the generator component is to add a new pair of concave corners adjacent to each previous concave corner such that

(i) These new pairs of concave corners are not covered by any "essential" rectangles which must occur in any optimal corner cover.

(ii) Any rectangle that covers one of these new concave corners (and possibly some other corners) can always be extended or modified to cover both of these concave corners (along with the other covered corners) as well as the two original concave corners adjacent to them. For example, in *fig.* 2(d) any rectangle that covers corner x can always be extended to cover corners y, z and w.

The modified generator component is shown in *fig.* 2(d). Because of the properties (i) and (ii) above, any corner cover can easily be extended without increasing the number of rectangles into a corresponding "notch cover" and vice versa. Hence, the following lemma can be proved using an argument similar to the that used in Lemma 5.4 of Conn and O'Rourke [4].

Lemma 3.1 *The modified generator open polygon can be covered with* 28 *rectangles if and only if all of the wires connected to it have the same truth value (extra* 17 *rectangles come due to additional* 17 *"essential" rectangles).*

The modified And/Or open polygon is shown in *fig.* 3. In *fig.* 3(ii) the "essential" rectangles which must appear in any optimal corner cover are shown shaded. In this polygon if the convex corner a is covered together with the convex corner b, and the convex corner c is covered together with the convex corner d (corresponding to when X=Y=FALSE), then the concave corners of all the holes can be covered by 3 additional rectangles (*fig.* 3(b)(v)), and any other way of covering these concave corners will use more rectangles in the total cover. However, if the corner a(*resp.* c) is not covered with b(*resp.* d) (corresponding to the remaining 3 combinations of input truth values), then it is profitable to cover the concave corners ϵ, δ and ϕ (*resp.* α, β and γ) of the holes with the same rectangle which covers a(*resp.* c) (*fig.* 3(b)(iii)(iv)(vi)), since this allows us to save $\frac{1}{2}$ rectangles at one or both of the outputs. By an exhaustive case analysis for various input truth values and the corresponding size of optimal covers needed for them the following lemma can be verified (optimal covers for various input truth values are shown in *fig.* 3(iii)-(vi)).

Lemma 3.2 *The modified And/Or polygon can be covered with* 14 *rectangles if and only if the AND (resp. OR) output has the same truth value as the logical AND (resp. OR) of its input wires.*

Since now we have all the different types of component polygons available to us, the following theorem follows.

Theorem 3.1 *The problem "are there k rectangles that cover all the corners of an arbitrary rectilinear polygon?" is NP-complete.*

4 Heuristics for the Boundary and Corner Cover Problems.

The boundary cover problem was already proved to be NP-hard, even if the given polygon has no holes [6]. Also, we proved in the previous section that the corner cover problem is NP-complete. However, we show that it is possible to approximate both these problems with constant performance ratio.

In the following discussion, we assume, without loss of generality, that the vertices of the given polygon have integer coordinates (*i.e.,*placed on a grid).

4.1 Heuristic for the Boundary Cover Problem.

The following heuristic guarantees a performance ratio of 4.

Input: A rectilinear polygon P, possibly with holes.
Output: A set of rectangles which cover the boundary of P.
Algorithm:

> For each edge $e = (a, b)$ of P create an adjacent rectangle of width 1. The side containing e is termed as the *principal side* of the rectangle. First, extend this rectangle in the direction of the edge (*i. e.,* if this edge is horizontal (*resp.* vertical) extend in the horizontal (*resp.* vertical) direction) maximally until it hits the boundary. Then, extend the rectangle in the other direction maximally (see *fig.* 4(a)). Delete any repeated rectangles. The set S of the remaining rectangles constitutes an approximate cover.

Lemma 4.1 *The performance ratio of the above heuristic is* 4*. It runs in* $O(n \log n)$ *time.*

Proof. Rectangles in any optimal cover can be associated with the principal sides of the rectangles of the cover generated by the heuristic such that no principal side is associated to more than one rectangle of the optimal cover, each principal side is associated to some rectangle in the optimal cover, and each rectangle in the optimal cover is associated to at most 4 principal sides. Hence, we have a performance ratio of 4. The example in *fig.* 4(b) shows that this performance ratio is tight asymptotically. The rectangles needed by the heuristic can be constructed using a sweep-line algorithm in $O(n \log n)$ time.□

4.2 Corner cover for polygons with holes.

The following heuristic guarantees a performance ratio of 4.

Input: A rectilinear polygon P, possibly with holes.
Output: A set of rectangles which cover the corners of P.
Algorithm:

> Form the two sets S and T; S contains, for each horizontal segment e of P, an adjacent rectangle of width 1 of maximal horizontal extent (this is the *principal side* for this rectangle), and T contains, for each vertical segment e of P, an adjacent rectangle of width 1 of maximal vertical extent (the principal side is defined similarly). Now, form a bipartite graph $G = (S \cup T, E)$, where $E = \{(y, z) \in S \times T \mid$ principal sides of y and z share a corner $\}$ (see *fig.* 5 for an example). Construct a minimum vertex cover R of G using maximum matching. The set of rectangles R constitutes our approximate cover.

Lemma 4.2 *The above heuristic has a performance ratio of* 4*. It runs in* $O(n^{1.5})$ *time.*

Proof. Each edge of G is associated with a corner of P. Therefore, being a vertex cover of G is the same as being a corner cover of P. On the other hand, given a corner cover C, we may replace each rectangle x of C with at most four elements of $S \cup T$, so that each corner covered by x is covered by the principal sides of the replacement rectangles. This means that a vertex cover of G is no more than 4 times the size of the optimal cover. The running time is dominated by the time taken to find the vertex cover for a bipartite graph. The graph has at most $O(n)$ vertices and also at most $O(n)$ edges, since each edge of the graph corresponds to a corner of the given polygon P. Hence, it takes only $O(n^{1.5})$ time for the minimum vertex cover using the efficient matching algorithm for a bipartite graph [11]. The tightness of the performance ratio follows from the same example shown in *fig.* 4(c) for boundary cover. □

Remark: When the given polygon has no holes, it is possible to design a heuristic which runs in $O(n \log n)$ time and has a performance ratio of 2. However, the analysis of the heuristic is quite lengthy and interested readers are referred to [7] for further details.

5 Impossibility of Approximation Schemes.

Culberson and Reckhow [6] showed the NP-hardness of the interior and boundary cover problems for polygons without holes by reducing the 3-SAT problem to these problems. However, their reduction introduces quadratically many rectangles in the optimal solution and hence violates condition ($\star$) of an L-reduction as discussed in section 2.1. Similarly, it can be easily seen that condition ($\star\star$) of an L-reduction is also not maintained by their reduction. In this section we show that the VC_d problem can be L-reduced to the interior or boundary cover problems. Hence, from the discussions in section 2.1, it follows that a PTAS for these covering problems is impossible, unless $P = NP$.

We first consider the interior cover problem. The overall scheme of our approach is shown in *fig.* 6. We use a *gadget* for every vertex and for every edge. Beams (rectangles) coming out of a gadget indicate that this vertex participates in vertex cover. Using polygonal boundary segments, the beams are first translated, then permuted appropriately, again translated and finally enter the edge-gadgets so that the whole structure becomes a closed rectilinear polygon. Each edge gadget is coupled with two beams and represents an edge between the two vertices which correspond to the two beams. We need to show how to ensure that a vertex cover of the graph correspond to the interior cover of the constructed polygon and vice versa.

Next, we describe our construction in more details.

Beam machine: Each beam machine is an open polygon whose interior can be covered optimally with 6 rectangles with only one rectangle extending through the discontinuity of its boundary in horizontal or vertical direction (see *fig.* 7). Depending upon whether the rectangle extends out horizontally or vertically, we say that the beam machine is using its horizontal or vertical beam, respectively. In each of the two optimal covers there is an uncovered square adjacent to its boundary (shown in *fig.* 7 by thick lines). Any other cover of the beam machine must necessarily use more than 6 rectangles. This is the same open polygon used by Culberson and Reckhow [6].

Vertex gadget: The vertex gadget for a vertex v is shown in full details in *fig.* 8 and *fig.* 9. It consists of d beam machines when d is the degree of v ($d = 3$ in the figure). There is one additional beam machine at bottom right, and a notch at the extreme left bottom, which forces this addition beam machine to use its horizontal beam. The background of this gadget can be covered optimally with $2d + 1$ rectangles, thus leaving out d uncovered squares (*fig.* 9). These squares can be covered by the horizontal beams of the d beam machines producing an optimal cover of the vertex gadget, or by one more additional rectangle producing a non-optimal cover of the vertex gadget. One motivation of covering the interior of the vertex gadget with a non-optimal cover is that this allows the beam machines in the vertex gadget to use their vertical beams, which may be used to cover parts of other polygonal regions which are connected to the vertex gadget. In particular, this gadget has the following properties:

(a) There is an optimal cover of this gadget with $8d + 7$ rectangles when each beam machine uses its horizontal beam. This corresponds to the case when this vertex does not participate in a vertex cover.

(b) If any one of the beam machines uses its vertical beam then $8d+8$ rectangles are necessary and sufficient to cover this gadget. The same property holds when more than one beam machine uses their vertical beams. This corresponds to the case when this vertex participates in a vertex cover.

For each of the covers described above, it is also possible to place an equal number of *anti-rectangle* points in the interior of the polygon.

Edge gadget: This gadget is shown in *fig.* 10. If either of its two input beams are used then it can be covered optimally with 4 rectangles, otherwise it requires at least 5 rectangles. This is same as the *inverter* structure of [6].

Translation stage: It consists of e pairs of beam machines as shown in *fig.* 11, where e is the number of edges in the graph. A *joint* rectangle is the horizontal beam which is shared by an aligned pair of beam machines. For optimal cover, if the "incoming" vertical beam for the left-side beam machine of the aligned pair is present (*i. e.,* the corresponding vertex participates in the vertex cover), then the "outgoing" vertical beam from the right-side beam machine of the aligned pair should be used for optimal cover; otherwise the common horizontal beam (*i.e.,* the joint rectangle) between must be used by the aligned pair for an optimal cover of the interior of this open polygon. We know, from our discussion of beam machines before, that each beam machine has an uncovered square near its mouth depending upon whether its horizontal or vertical beam has been used. Hence, the unique optimal cover for the "background" of this stage (covering the staircases with the uncovered squares at the mouth of the beam machines) requires $2e$ rectangles. This is a slightly simplified version of the joint open polygon used by Culberson and Reckhow [6]. There are two translation stages, one connecting the vertex gadgets to the rest of the polygon, and the other one connecting the edge gadgets to the rest of the polygon. They are necessary so that the optimal covering of the rest of the polygon does not affect the covering of the vertex gadgets and the edge gadgets in any significant way.

Permutation stages: These open polygons are needed because the order in which the beams come out of the vertex gadgets is not necessarily the same as they should arrive at the edge gadgets (*i.e.,* we need to permute the beams). For example, consider a graph G with vertices v_1, v_2, v_3 and edges $\{v_1, v_3\}$, $\{v_2, v_3\}$. There is one beam α out of v_1, one beam β out of v_2, and two beams γ and δ out of v_3 (see *fig.* 12). The beams start in the order $\alpha, \beta, \gamma, \delta$ and must be permuted to produce the sequence $\alpha, \gamma, \beta, \delta$, so that the edges can be realized (see *fig. 12*). Unfortunately, the component polygons used by Culberson and Reckhow [6] cannot be used without violating the constraints of L-reduction. Hence, we use a different approach. In essence, we mimic the following algorithm to generate the required permutation $Y = (y_1, y_2, \ldots, y_n) = X_\sigma = (x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(n)})$ of the input sequence $(x_1, x_2, \ldots, x_n)$:

for $i = 1$ to n do

$y_i = x_{\sigma(i)}$

More precisely, there are at most $p \leq 2e$ permutation stages when e is the number of edges in the graph. Each stage consists of staircases and holes as shown in *fig.* 13. In the i^{th} permutation stage we put the i^{th} beam from left "in the required permutation" in its correct place (i.e., do what the i^{th} iteration of the above algorithm does). The right-side hole is placed appropriately to make it impossible for the right-staircases of all the previous permutation or translation stages to be covered by any rectangle which covers optimally the background this or later stages. If an input beam is present and covers one notch of the left-side hole, the vertical beam of the beam machine at the right should be used for the optimal cover (see *fig.* 14), otherwise, for optimal cover, the horizontal beam of this right-side beam machine covers the notch of the right-side hole (and, hence, its vertical beam cannot be used in an optimal cover). In all we need 8 rectangles to cover each stage, because, there are 8 anti-rectangle points, and none of these anti-rectangle points can be covered by rectangles covering those of previous or later stages. The details of the properties of this stage are given in Lemma 5.3 below.

First, we state the Lemmas which state the properties of the various stages discussed above.

Lemma 5.1 *A vertex gadget can be covered optimally only if all of its vertical beams are not "used".*

Proof. Follows from the above discussion of the vertex gadget. □

Lemma 5.2 *For each translation stage the following are true:*

(a) *It requires 2e rectangles to cover its staircases along with the uncovered squares of its beam machines (i. e., its background).*

(b) *If the incoming vertical beam to a left-side beam machine is present, then for an optimal cover the outgoing beam vertical of the corresponding aligned right-side beam machine should be present. However, if the incoming beam is not present, then for optimal cover the common horizontal beam of these aligned pair of beam machines must be used.*

(c) *The beam machines can always be covered optimally, if the rule stated in part (b) above is followed.*

(d) *The optimal cover of the background of this stage cannot be affected by any rectangle that participates in an optimal cover of vertex or edge gadgets or other stages.*

Proof.

(a), (b), (c) Follows from the discussion in the description of the translation stage.

(d) No rectangle that covers part of vertex gadget can cover the right-upper staircases of the first translation stage. Same result is true about rectangles of other stages and those in the translation stages also. Similarly, no rectangle of edge gadget or other stages can cover the left-lower staircases of the last translation stage.□

Lemma 5.3 *Consider the i^{th} permutation stage (stage i). Then the following are true (fig. 15):*

(a) *Stage i has 8 anti-rectangle points (for its background cover) which cannot be covered together with those of any other stage. Also, 8 rectangles are sufficient to cover the background of stage i.*

(b) *Let the two notches of the left hole be n_1 and n_2 and that of right hole be n_3 (fig. 15). Then,*

- **(i)** *if n_1 is covered by the incoming beam, then for optimal cover n_2, n_3 and the three corners of the left staircases must be covered by rectangles which cover the 5 corners of the right staircases and hence the right-side beam machine can use its vertical beam.*
- **(ii)** *if n_1 is not covered by the incoming beam, then for optimal cover notches n_1, n_2 and the three corners of the left staircases must be covered by the rectangles which cover the five corners of the right staircases and n_3 is covered by the horizontal beam of the beam machine (and hence the outgoing vertical beam is absent).*

(c) *Anti-rectangle points for covering the background of stage i cannot be covered together with those of any other stage.*

(d) *The stage i places the i^{th} beam from the left in the required permutation into its correct position.*

Proof.

(a) 5 anti-rectangle points placed in 5 corners of the right staircases are not visible to any other stage (since the right-side hole of stage $i+1$ prevents their visibilities to remaining stages). The leftmost anti-rectangle point is placed between the left hole and the boundary directly above the left staircase of the next stage and hence cannot be covered with similar other points of other stages. Similar argument holds for the placement of the rightmost anti-rectangle point also. The middle anti-rectangle point is always placed between the two holes of this stage below the right-side hole of the previous stage, except for stage 1, which guarantees its isolation from other permutation stages. The middle anti-rectangle point for stage 1 is placed between the two holes anywhere.

(b) **(i)** We show by case analysis that it is true.

Case 1. One or more of the 3 left staircases are not covered by right staircases. Then, we need at least 6 rectangles.

Case 2. One or both of n_2 or n_3 not covered. Then, we need at least 6 rectangles.

(ii) We show by case analysis that it is true.

Case 1. 5 right staircases cover 3 left staircases, n_1 and n_2; n_3 is covered by the rightside beam machine. Then we need 5 rectangles (optimal case).

Case 2. One or more of 3 left staircases are not covered by right staircases. Then, we need 6 rectangles.

Case 3. Either of n_1 or n_2 is not covered by right staircases. Then, we need at least 6 rectangles.

(c) Follows from the fact that none of the anti-rectangle points in the optimal cover can be seen by any other stage.

(d) Follows from the discussion in the description of permutation stages.□

Lemma 5.4 *Consider the edge gadgets and their connection to the last translation stage (fig. 16). The background of the edge gadget needs $2e-1$ rectangle to cover optimally independent of any other stage (where e is the number of edge gadgets). This optimal cover of the background of the edge gadgets is not influenced by the rectangles in the optimal cover of the backgrounds of the previous stages. If an edge gadget gets one or two incoming beams, it needs only 4 more rectangles to cover it.*

Proof. Rectangles covering vertex gadgets or stages 1 to p cannot extend to the edge gadgets. The beam rectangles in the last translation stage can extend to cover the edge gadgets (see *fig.* 16), but these rectangles are not part of the optimal cover of the background of this last translation stage. $2e-1$ rectangles are necessary and sufficient to cover the background of the edge gadget (the corresponding anti-rectangle points are placed as shown in *fig.* 16). □

Now that we have discussed each component polygon individually, we need to show how to connect them appropriately to produce a closed polygon. We have already shown how to connect successive permutation stages and how to connect the last translation stage to the edge gadgets. Connecting the first translation stage to the first permutation stage and connecting the last permutation stage to the last translation stage is essentially similar to connecting successive permutation stages. What remains to be shown is how to connect the vertex gadgets to the first translation stage. This is discussed in the following Lemma.

Lemma 5.5 *The part of the polygon which connects the vertex gadgets to the first translation stage needs v rectangles to cover it independently of the optimal cover of the background of any other stage or gadgets, where v is the number of vertices of the graph. These rectangles do not influence the optimal cover of the remaining part of the complete polygon.*

Proof. We can place anti-rectangle points at the joints of the vertex gadgets to the first translation stage, and none of these anti-rectangle points can be covered together with those in the covers of the vertex gadgets, the translation stage or any other stages (see *fig.* 17).□

We show, in *fig.* 18, the complete polygon for a simple graph with two vertices.

Lemma 5.6 *Any covering of such a constructed polygon, corresponding to a given graph, can be transformed in polynomial time (in the size of the given graph) to another cover with the following properties without increasing the number of covering rectangles:*

(a) *Either all the vertical beams of a vertex gadget are used or none is used.*

(b) *Each edge gadget uses atleast one of its two incoming vertical beams from the previous translation stage (and hence needs 4 additional rectangles for its optimal cover).*

(c) *The background of the vertex gadgets, translation stages, permutation stages and edge gadgets are covered with the optimal number of rectangles.*

Proof.

(a) If only some vertical beams are used for a vertex gadget we can as well use all the vertical beam rectangles without increasing the number of covering rectangles for the vertex gadget. Then, we keep moving successively from one stage to another till we reach the edge gadgets, turning off the common horizontal rectangles of the corresponding aligned pair of beam machines for this beam at each stage off and using the incoming and outgoing beam rectangles. This enables us not to use more rectangles than before. Finally, surely we do not use more rectangles at the edge gadget (we will use one less if it had no incoming vertical beam rectangles).

(b) Assume this is not the case. We choose arbitrarily one of the two beams of the edge gadget to use. Hence, we save one rectangle at the edge gadget. Now, we keep moving from the edge gadget through successive stages towards the vertex gadgets in a manner similar to as described in part (a) above. We may use one more rectangle when we reach the vertex gadgets if it was not using its beams, but the gain of one rectangle at the corresponding edge gadget compensates.

(c) Once we have done the transformations needed for parts (a) and (b) above, we can select the necessary background covers depending on whether the incoming vertical beams of a particular stage are used or not as outlined in Lemma 5.5, Lemma 5.2(part (a)), Lemma 5.3 and Lemma 5.4, since the optimal covers of these parts of the polygon are independent of each other and depends only on the presence or absence of incoming vertical beams.

It is obvious that the transformations in parts (a), (b) or (c) above takes polynomial time.□

Lemma 5.7 *Let $G = (V, E)$ be a connected graph in which the degree of any vertex is at most a constant $d > 0$ and P be the polygon constructed from it by the above procedure. Let p be the number of permutation stages used be ($0 \leq p \leq d\cdot \mid V \mid$). Then, there exist four positive constants $\alpha_1, \alpha_2, \alpha_3$ and α_4 such that the following two results hold.*

(a) *Any vertex cover of G of size m corresponds to a interior cover of P of size $\alpha_1\cdot \mid V \mid + \alpha_2 + m$.*

(b) *Given an interior cover of P of size θ_1, we can transform this cover in polynomial time to another cover of size $\theta \leq \theta_1$ such that this transformed interior cover of P of size θ corresponds to a vertex cover of size $\theta - (\alpha_3 \cdot |V| + \alpha_4)$.*

Proof.

For the purpose of counting we always associate the outgoing beams for a particular stage with that stage. That is, for example, the outgoing beams for the vertex gadgets are counted in the total number of rectangles needed for the vertex gadgets.

(a) We need

- **(i)** $\mid V \mid$ polygons to cover the joint of vertex gadgets stage 0, by Lemma 5.5,
- **(ii)** $2\cdot \mid E \mid + 11\cdot \mid E \mid = 13\cdot \mid E \mid$ polygons to cover background and polygons of translation stage 0, due to the properties of beam machines and Lemma 5.2,
- **(iii)** $8 \cdot p + 6 \cdot p = 14 \cdot p$ rectangles to cover backgrounds and polygons of permutation stages 1 to p due to the properties of beam machines and Lemma 5.3,

(iv) $2\cdot \mid E \mid +11\cdot \mid E \mid = 13\cdot \mid E \mid$ polygons to cover translation stage $p+1$, due to the properties of beam machines and Lemma 5.2,

(v) $6\cdot \mid E \mid -1$ polygons to cover the edge gadgets and its background, due to the properties of edge gadgets, Lemma 5.4 and Lemma 5.6(part(b)),

(vi) Each vertex participating in the vertex cover needs at most $(7\cdot d+8)$ rectangles, others need at most $(7\cdot d+7)$ rectangles, due to the properties of vertex gadgets. Hence, the total count for the polygons in the vertex gadgets is at most $m\cdot(7\cdot d+8)+(\mid V \mid -m)\cdot(7\cdot d+7)$.

The result now follows by adding up all of the above, and noting that $p \leq d \cdot |V|$ and $|E| \leq \frac{d\cdot|V|}{2}$.

(b) The Proof is similar to (a) above. We need to select a vertex in the vertex cover if and only if the corresponding vertex gadget in the polygon uses all its vertical beams.□

Theorem 5.1 *The VC_d problem can be L-reduced to the interior cover problem for rectilinear polygons.*

Proof. Given a graph $G=(V,E)$ as an instance I of the VC_d problem, we need to exhibit three polynomial-time algorithms T_1, T_2, T_3 and two constants $\alpha, \beta > 0$ such that conditions $(\star)$ and $(\star\star)$ of an L-reduction is satisfied. Let T_1 be the transformation to construct the polygon I' from G as outlined in this section.

First, consider condition $(\star)$. From the discussion in section 2.1, we know that $c_1 \cdot |V| \leq OPT(I) \leq c_2 \cdot |V|$ for two positive constants c_1 and c_2. It is trivial to see that $OPT(I') \geq c_3 \cdot |V|$ for some positive constant c_3 (just consider covering the vertex gadgets). Hence, using Lemma 5.7, it follows that there exists two constants c_3 and c_4 such that $c_3 \cdot |V| \leq OPT(I') \leq c_4 \cdot |V|$. Hence, there exists a constant $\alpha > 0$ such that $OPT(I') \leq \alpha OPT(I)$.

Next, we consider condition $(\star\star)$. Given any solution of the interior cover problem of I' of size θ_1, let T_2 be the transformation needed to satisfy conditions (a), (b) and (c) of Lemma 5.6. Let $\theta \leq \theta_1$ be the size of this transformed solution. Hence, by part (b) of Lemma 5.7, we have an algorithm T_3 to find a vertex cover of size $m = \theta - (\alpha_3 \cdot |V| + \alpha_4)$ for two constants α_3 and α_4. Hence,

$$\begin{aligned} m - OPT(I) &= \theta - \alpha_3 \cdot |V| - \alpha_4 - OPT(I) \\ &= (\theta - OPT(I)) - \alpha_3 \cdot |V| - \alpha_4 \\ &\leq c_5 \cdot (\theta - OPT(I')) - \alpha_3 \cdot |V| - \alpha_4 \quad \text{for some constant } c_5 \\ &\leq \beta(\theta - OPT(I')) \end{aligned}$$

Hence, we have proved the following result.

Theorem 5.2 *No polynomial-time approximation scheme exists for the interior cover problem for rectilinear polygons, unless P=NP.*

A careful examination of our construction shows that all the results hold for boundary cover also, in particular we can always place all the anti-rectangle points on the boundary of the polygon. Hence, we also prove the following result.

Theorem 5.3 *No polynomial-time approximation scheme exists for the boundary cover problem for rectilinear polygons, unless P=NP.*

Unfortunately, the above reduction does not hold for the corner cover problems, some of the component polygons do not have the necessary covering properties when the corner cover problem is considered.

6 Conclusion and Open Problems.

We have proposed efficient heuristics for the boundary and corner cover problems for rectilinear polygons. We have also shown that the corner cover problem for rectilinear polygon is NP-complete, and it is impossible to obtain a polynomial-time approximation scheme for the interior and boundary cover problems for rectilinear polygons, unless P=NP. However, the last result does not exclude the possibility of designing heuristics with constant performance ratio for the interior cover problem. Franzblau[8] 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:

- Let α and θ be the sizes of the maximum-cardinality anti-rectangle set and minimum cardinality cover size for the interior cover of an arbitrary rectilinear polygon with holes. Is it true that $\frac{\theta}{\alpha} \leq c$ for some positive constant c?
- 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?

7 Acknowledgments.

We wish to thank H. E. Conn and J. O'Rourke for sending us their result [4], J. Culberson for sending their detailed construction [6] and G. Schnitger for suggesting the problem of non-approximability of interior cover problem and helpful discussions. We would also like to thank the anonymous reviewer for helpful comments which greatly improved the readability of this paper. This research was partially supported by NSF Grant CCR-9114545.

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy. Proof Verification and Hardness of Approximation Problems. *Proc. 33rd IEEE Symp. Found. of Comp. Sc.(FOCS)* (1992), pp. 14-23.

[2] P. Berman and G. Schnitger. On the Complexity of Approximating the Independent Set Problem. *Information and Computation*, 96 (1992), pp. 77-94.

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

[4] 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.

[5] T. H. Corman, C. E. Leiserson and R. L. Rivest. *Introduction to Algorithms.* The MIT Press, Cambridge, Massachusetts, 1990.

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

[7] P. Berman and B. DasGupta. Results on Approximation of the Rectilinear Polygon Cover Problems. Tech. Rep. # CS-92-06, The Pennsylvania State University, June, 1992.

[8] 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.

[9] 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.

[10] M. R. Garey and D. S. Johnson. *Computers and Intractability - A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., 1979.

[11] J. E. Hopcroft and R. M. Karp. An $n^{\frac{5}{2}}$ Algorithm for Maximum Matching in Bipartite Graphs. *SIAM J. Computing*, 2 (1979), pp. 225-231.

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

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

[14] W. J. Masek. Some NP-complete Set Covering Problems. MIT, Cambridge, MA, unpublished manuscript. Referenced in [10].

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

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

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

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

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

[20] C. H. Papadimtriou and M. Yannakakis. Optimization, Approximation and Complexity Classes. *Journal of Computer & System Sciences,* 43 (1991), pp. 425-440.

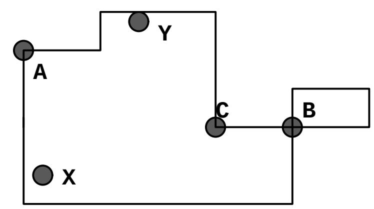

Figure 1: *In this rectilinear polygon A is a convex corner, C is a concave corner, B is a degenerate convex corner and (X, Y) is an anti-rectangle set (the points are shown magnified)*

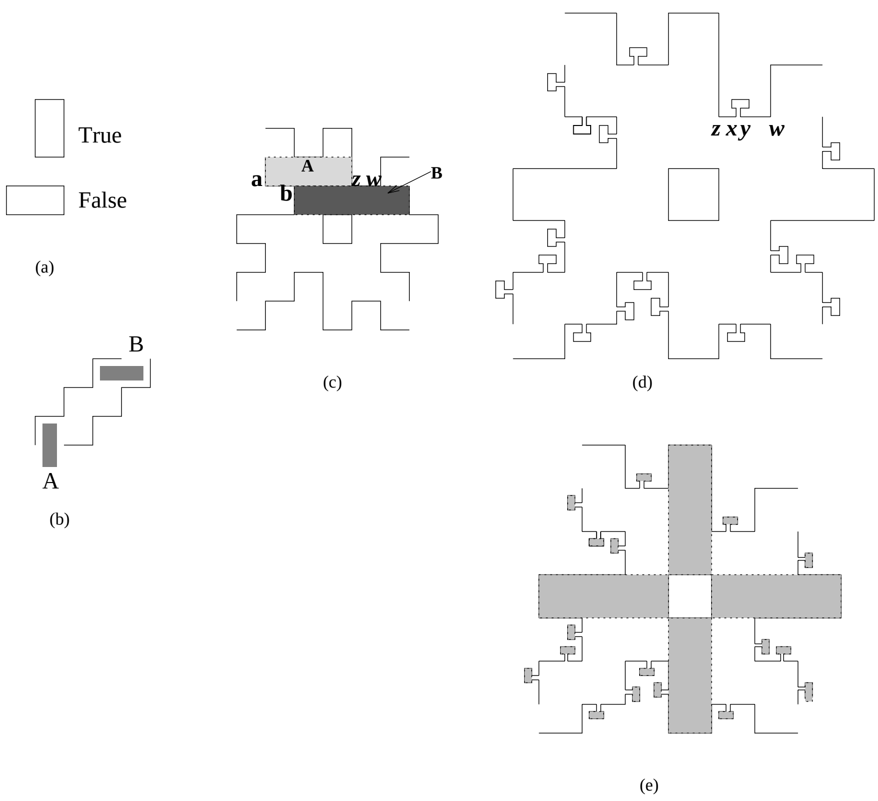

Figure 2: *(a) Rectangles corresponding to TRUE and FALSE values. (b) The rectangle B is counted as 1 rectangle, but the rectangle A is counted as $\frac{1}{2}$ rectangle in the cover of this component. (c) The original generator component of Conn and O'Rourke. (d) The modified generator component. (e) "Essential" rectangles in this modified component are shown shaded.*

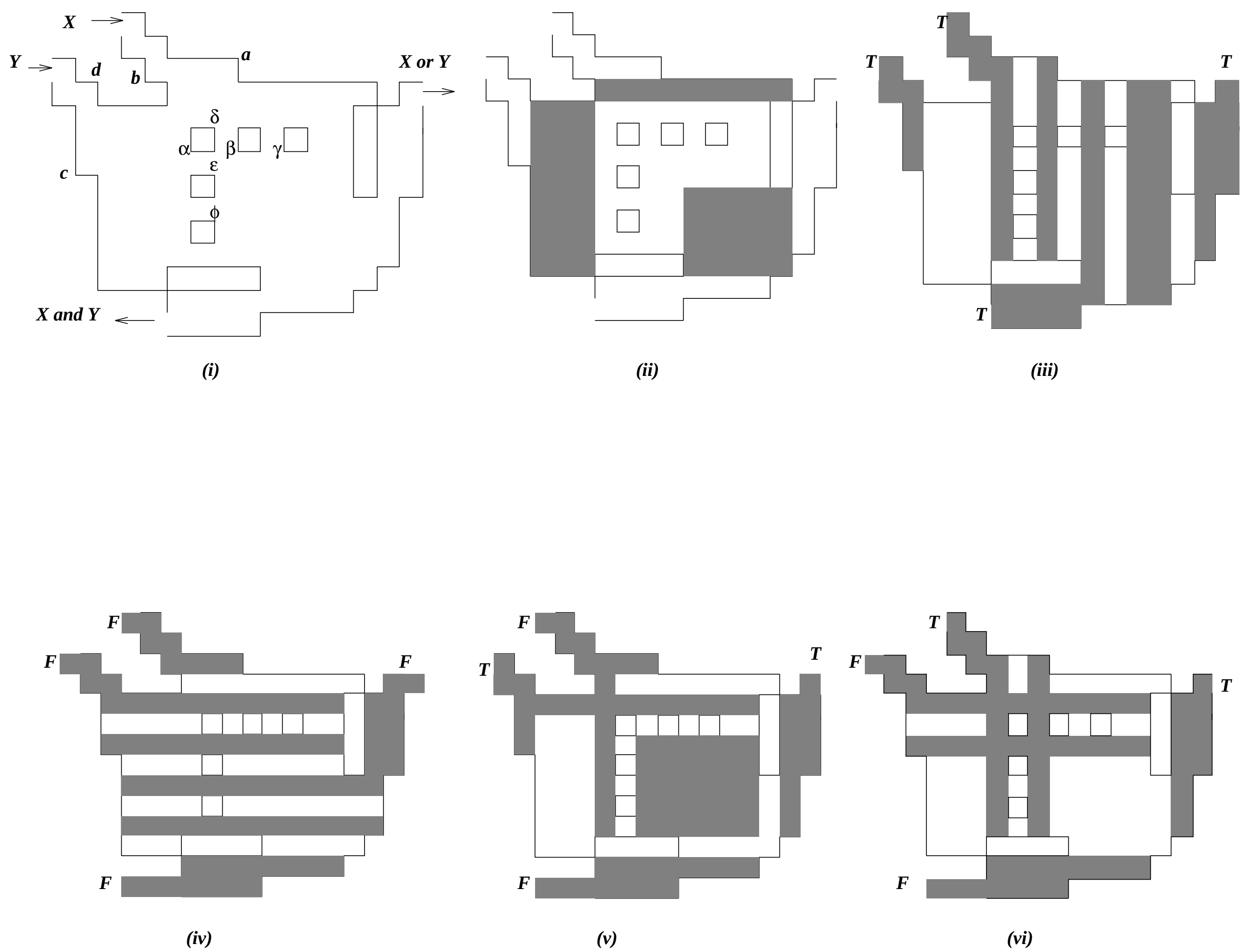

Figure 3: *(i) The modified And/Or component (ii) The "essential" rectangles (shown shaded) (iii)-(vi) "Non-essential" rectangles in the optimal covers of the And/Or component for various truth values of their inputs (T=TRUE, F=FALSE)*

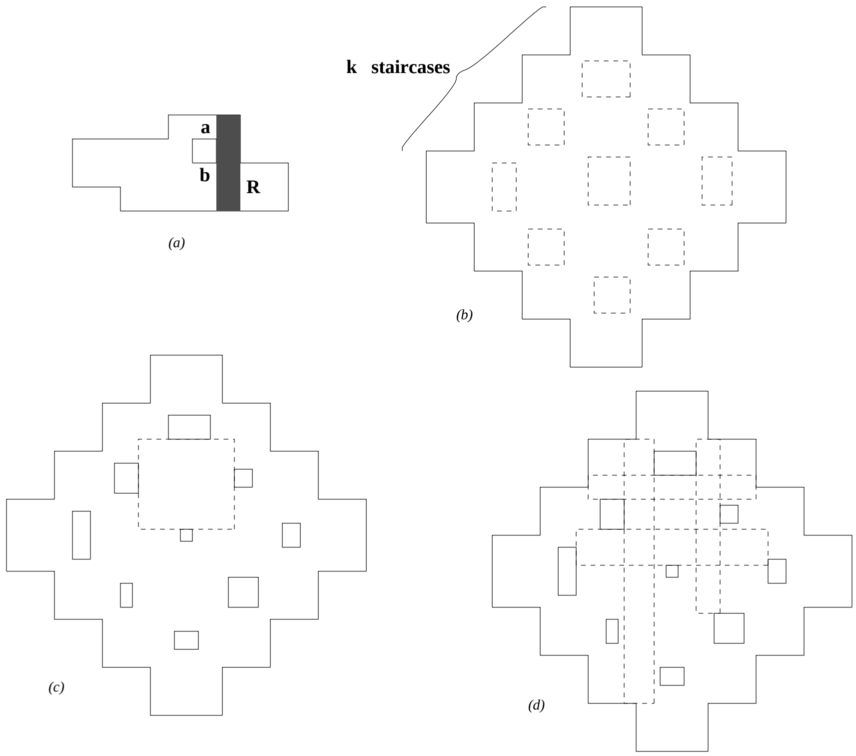

Figure 4: *(a)* `ab` *is the principal side of the rectangle* `R` *(b) A polygon with* $4k$ *staircases, in which* k^2 *holes are placed in the* k^2 *dotted square regions shown such that no two sides of the polygon (including both the outer boundary and boundary of the holes) are vertically or horizontally aligned. (c) The polygon is shown for* $k = 3$*, together with a rectangle (shown dotted) in its optimal boundary cover. (d) Four polygons in our heuristic which will be generated from the four principal sides of the optimal rectangle shown in (c). The optimal cover uses* $(k+1)^2$ *rectangles, whereas the heuristic uses* $4k^2 + \Theta(k)$ *rectangles.*

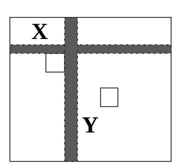

Figure 5: X *and* Y *are two vertices of the graph and* (X, Y) *is an edge*

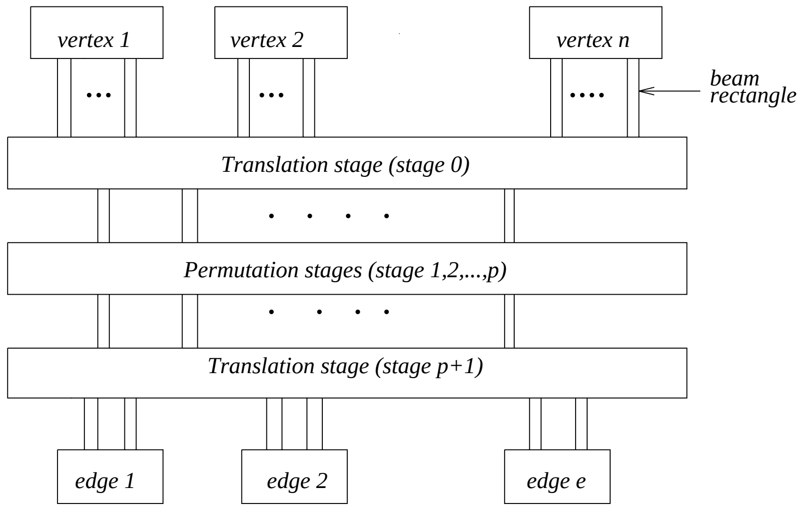

Figure 6: *The overall scheme*

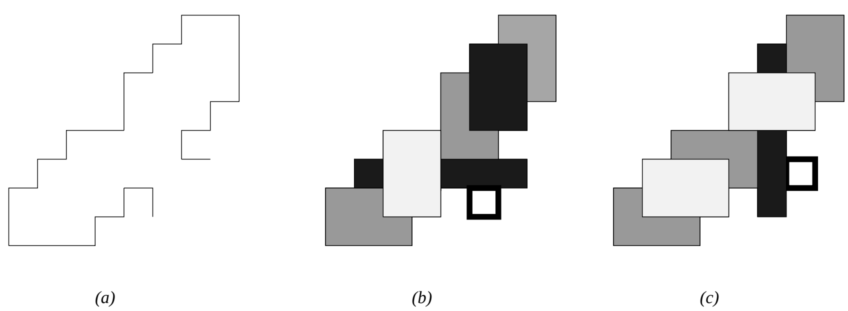

Figure 7: *A beam machine and its two optimal covers. The uncovered square near its mouth is shown by thick lines. In the optimal cover of (ii) (resp. (iii)) the beam is coming out horizontally (resp. vertically)*

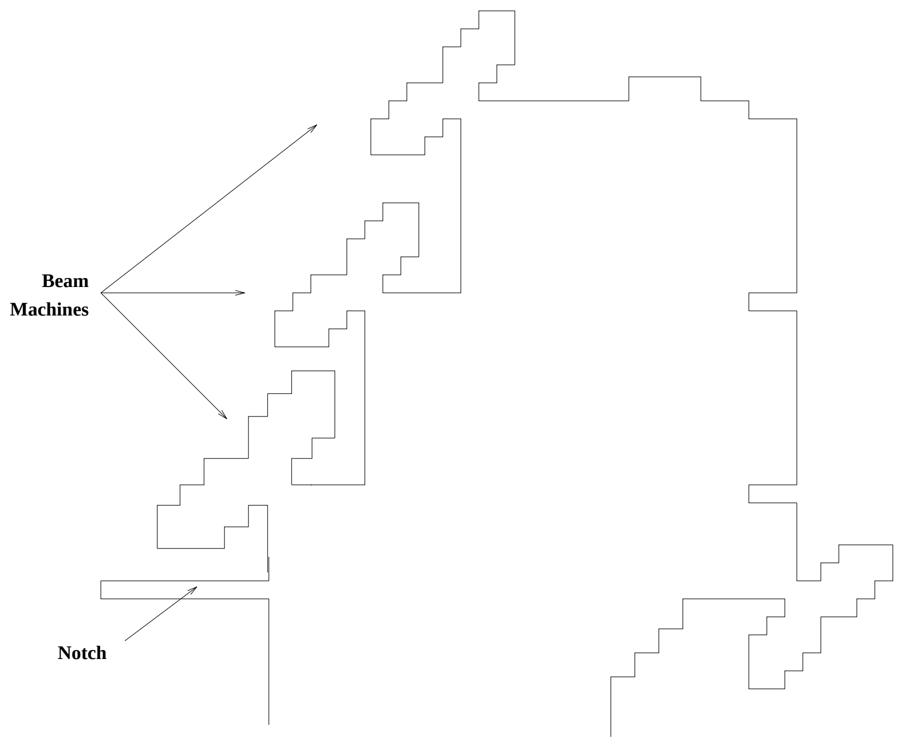

Figure 8: *(a) A vertex gadget for a vertex of degree d (the figure illustrates the case of $d = 3$).*

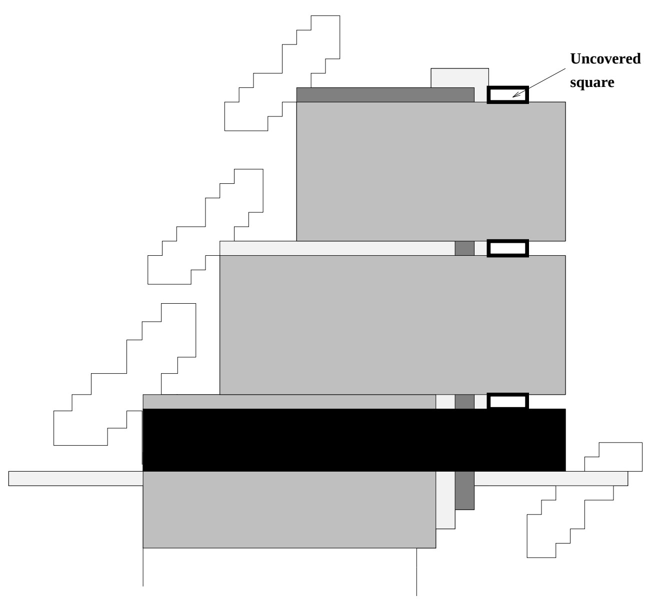

Figure 9: *(a) Optimal rectangle cover of the background (i.e., the part of the polygon outside the beam machines) of the vertex gadgets. The squares shown by thick lines are the "uncovered" squares.*

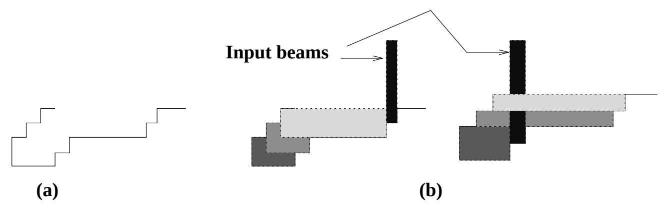

Figure 10: *(a) An edge gadget. (b) Its two optimal covers.*

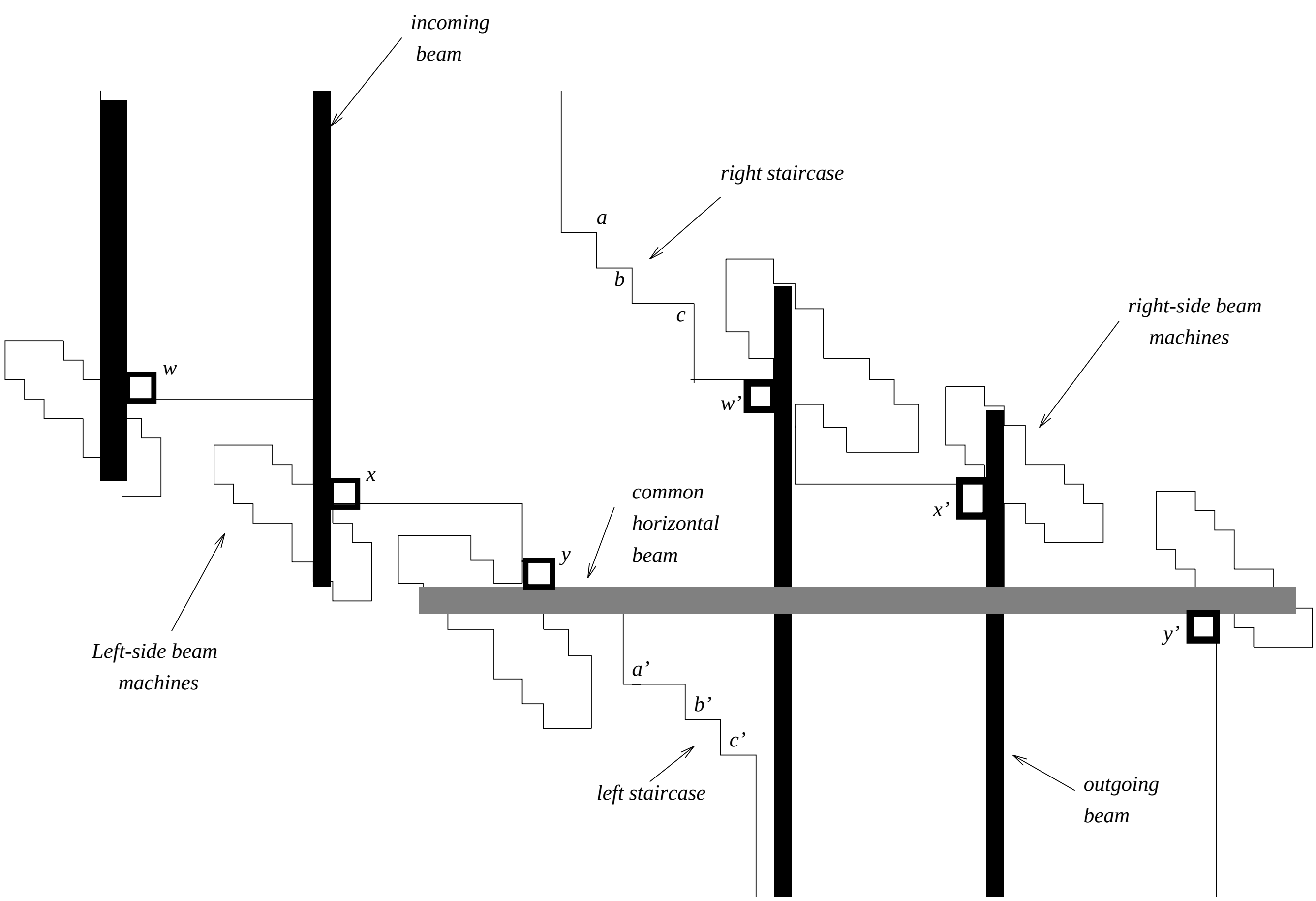

Figure 11: *A translation stage for 3 beams. The uncovered squares near the mouth of the beam machines are shown by thick lines. Two incoming beams are present and hence one pair of aligned beam machines use their common horizontal beams. For an optimal cover of the background of this stage, corners a, b and c must be covered with the uncovered squares w, x and y using three rectangles, and similarly corners a', b' and c' must be covered with the uncovered squares w', x' and y' using three rectangles.*

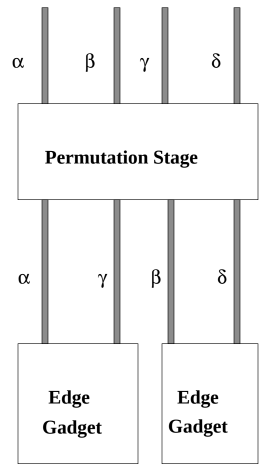

Figure 12: *An example showing why permutation stages are needed.*

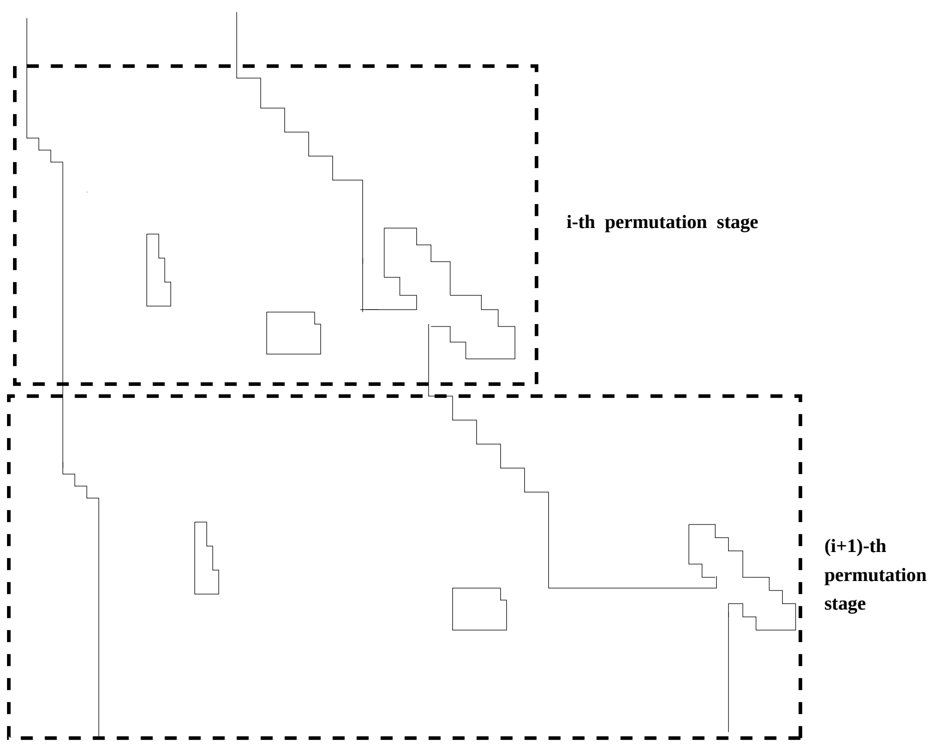

Figure 13: *Two consecutive permutation stages.*

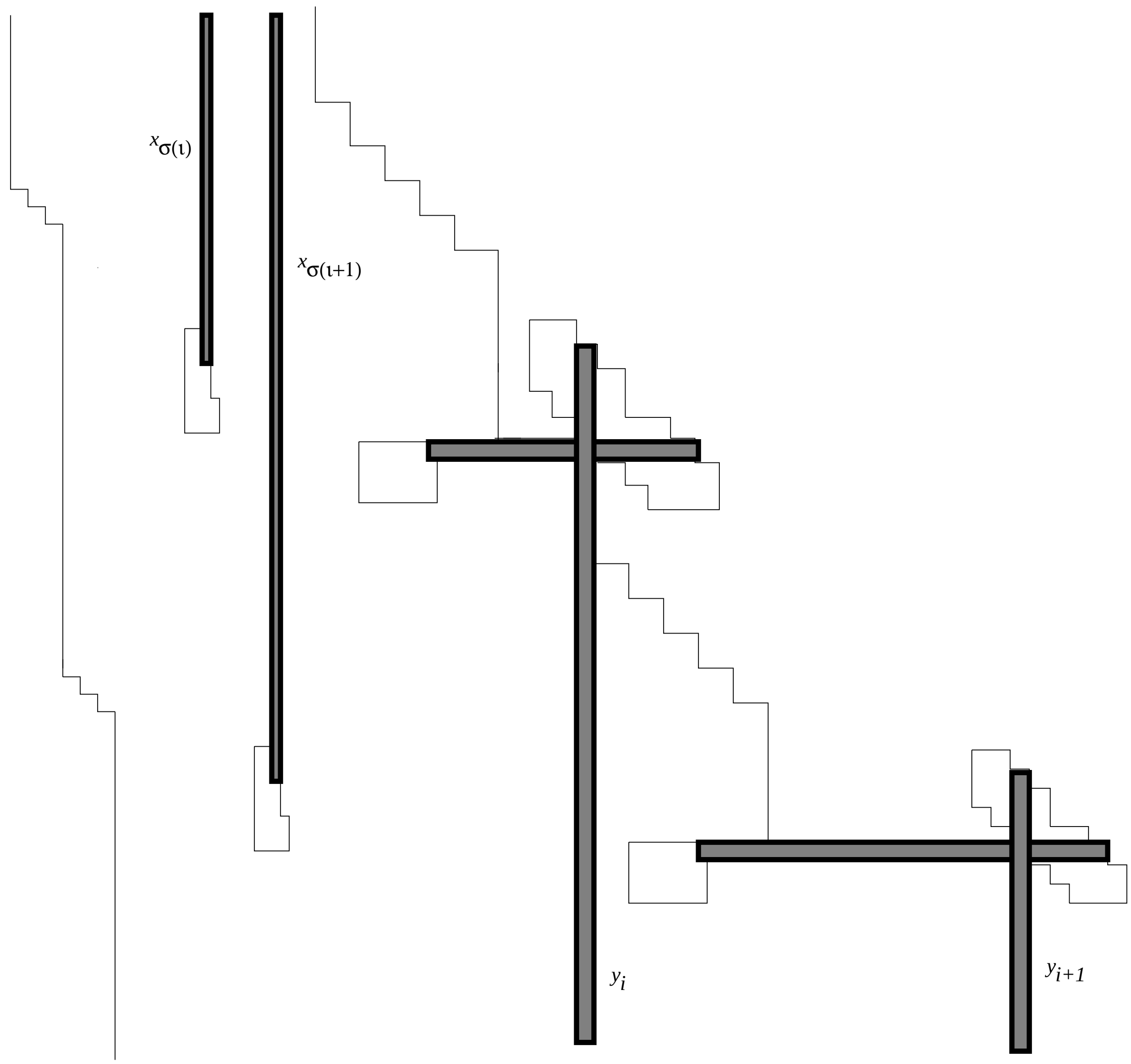

Figure 14: *The beams $x_{\sigma(i)}$ and $x_{\sigma(i+1)}$ which are permuted are shown. Note that there may be many vertical beams passing downwards between $x_{\sigma(i)}$ and $x_{\sigma(i+1)}$ as well as to the right of $x_{\sigma(i+1)}$; they are not shown for clarity and they will not be affected by these two stages.*

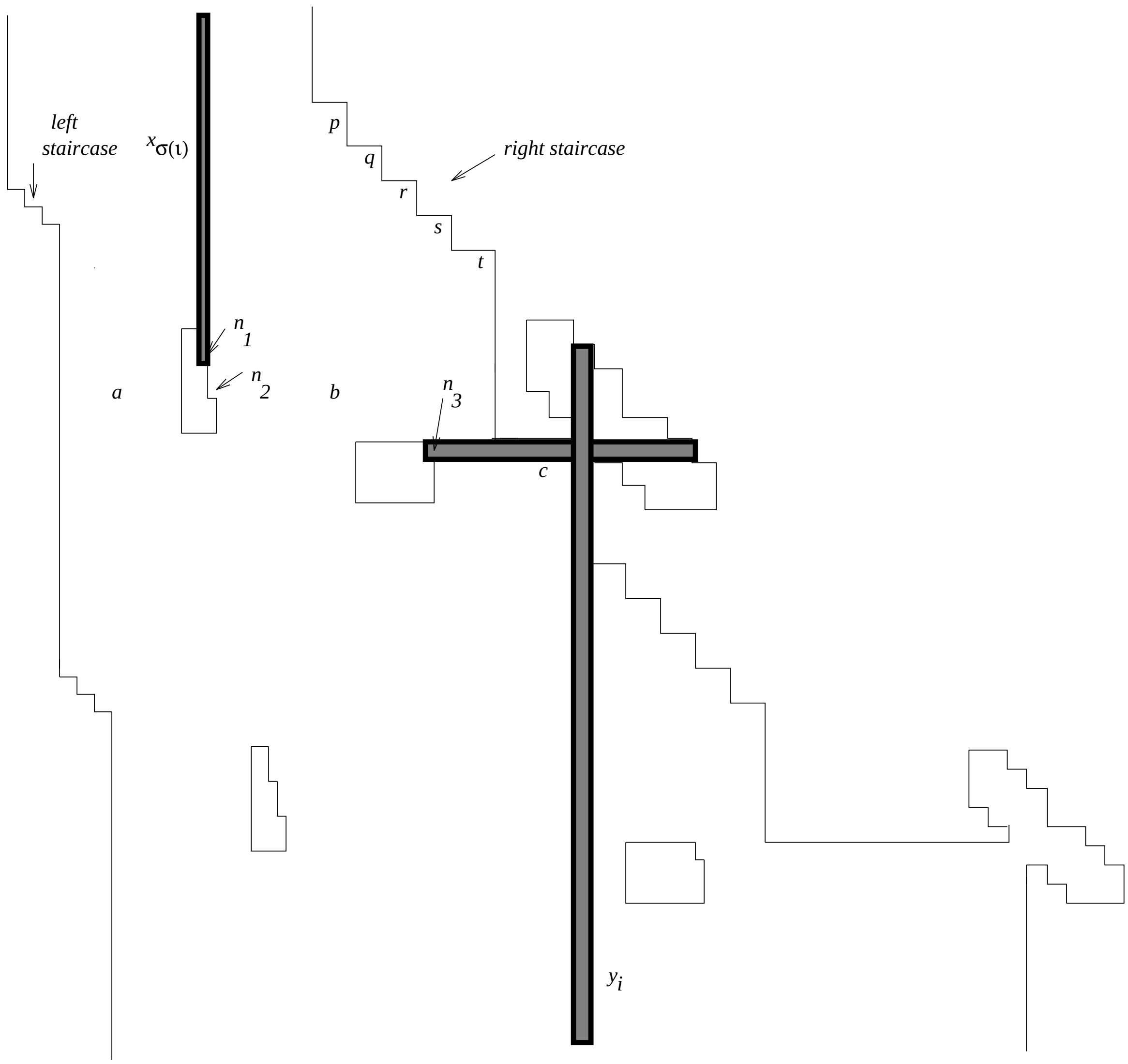

Figure 15: *The i^{th} permutation stage. The incoming beam is shown to be present. Points a, b, c, p, q, r, s, t are the anti-rectangle points placed. The point b is placed below the right-side hole of the previous stage.*

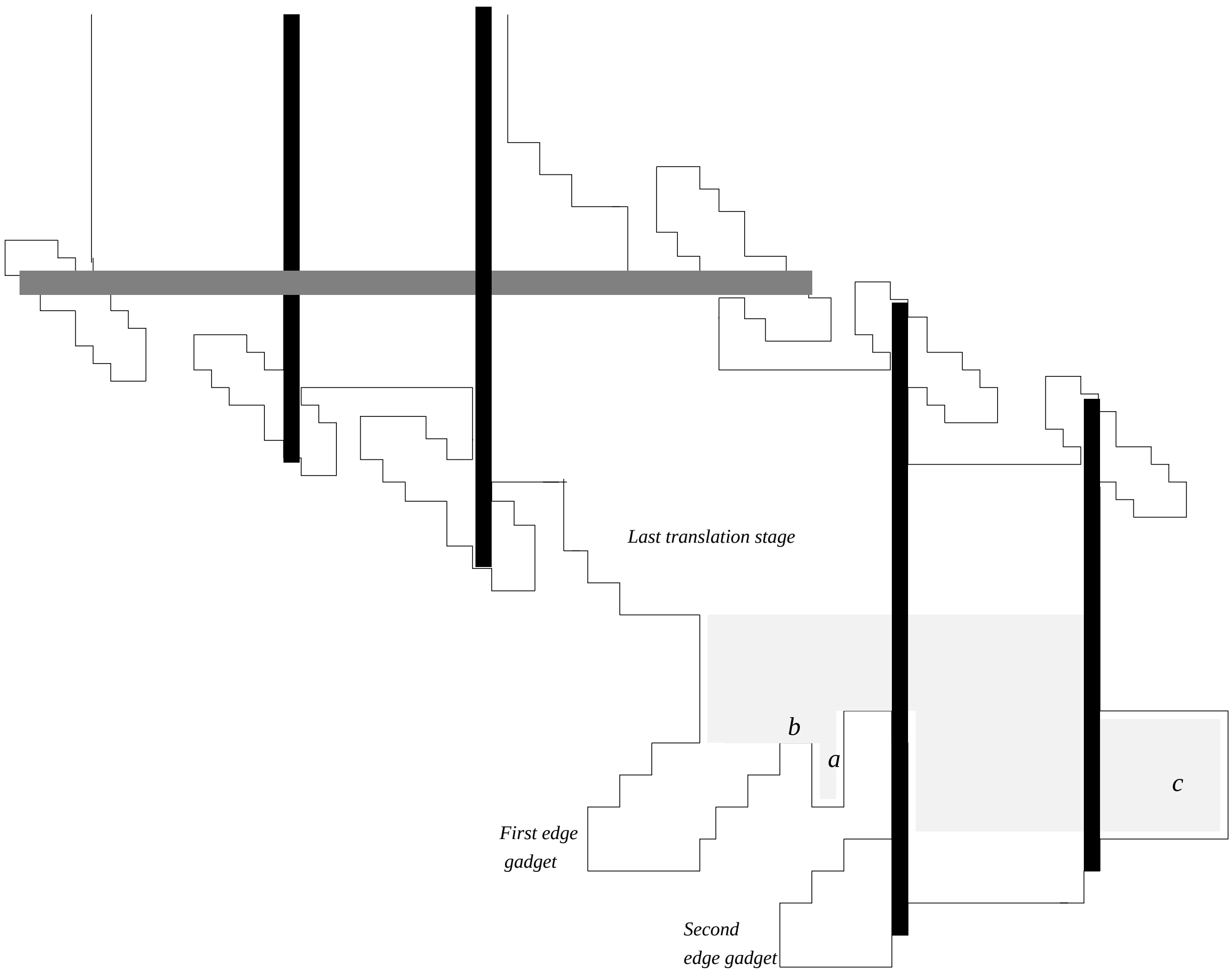

Figure 16: *Two edge gadgets for a graph having at most two edges. a, b, c are the anti-rectangle points in the optimal background cover of the edge-gadgets. The background of the edge gadget is shown shaded.*

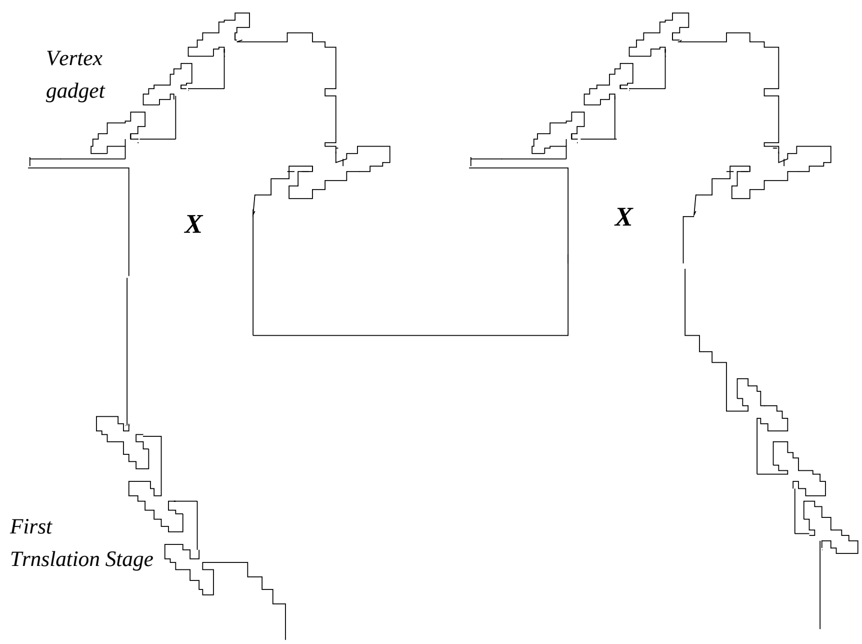

Figure 17: *Joining vertex gadgets (for a graph having 2 vertices) to the first translation stage (anti-rectangle points are marked by* **X***).*

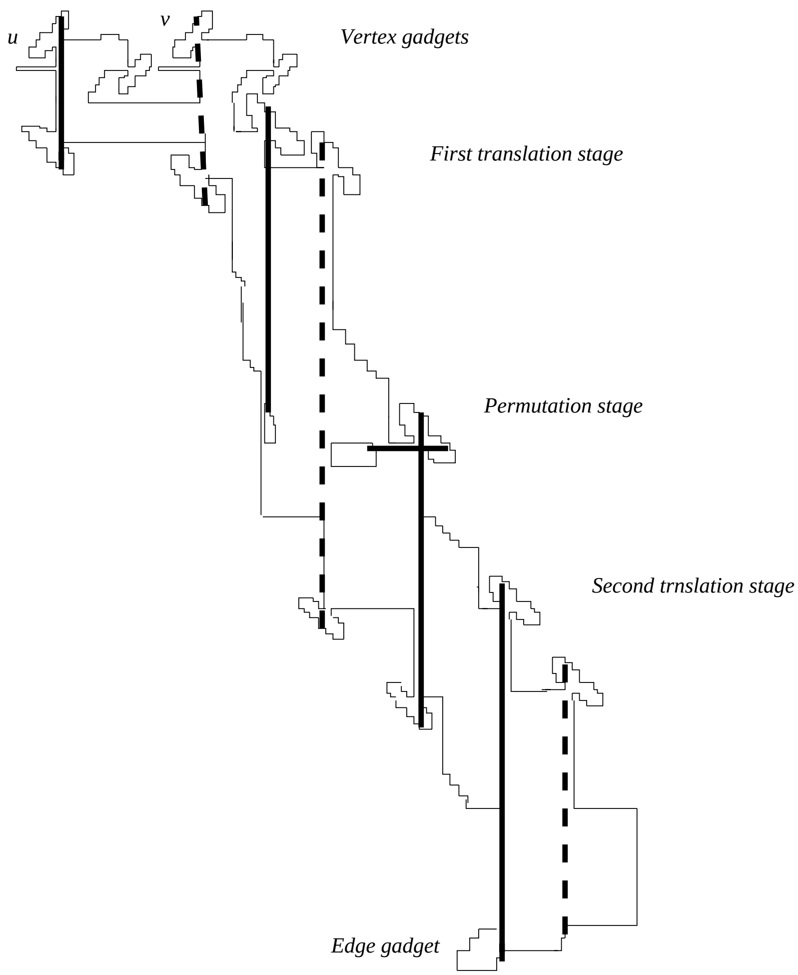

Figure 18: *The complete polygon for a graph* $G = (V, E)$ *with* $V = \{u, v\}$ *and* $E = \{\{u, v\}\}$. *Although there is no need for a permutation stage, we have used one permutation stage to illustrate how it can permute the beams. The beams for the vertices* u *and* v *are shown by thick and dotted lines, respectively. Both the vertices participate in the vertex cover since both the beams are used. Due to finite resolution of the plotting device, some aligned edges may seem a little offset.*