Rectangular Dualization and Rectangular Dissections

KRZYSZTOF A. KOZMINSKI, MEMBER, IEEE, AND EDWIN KINNEN, SENIOR MEMBER, IEEE

Abstract **—Rectangular dualization refers to finding a dual of a planar graph, that can be drawn in the form of a rectangular dissection; it has applications in architectural design and in the floorplanning of VLSI IC's. This paper presents properties of rectangular dissections and discusses two related topics: (1) a problem of enumerating without repetitions all rectangular duals of a graph, and (2) transformations of rectangular dissections.**

I. Introduction

RECTANGULAR dualization and rectangular dissections have been of interest to the IC design community since their application to floorplanning was demonstrated [1], [2]. Subsequently, the problem of the existence of rectangular duals for a class of triangulated graphs has been studied by the authors and others [3]–[8], [10]. Algorithms for finding rectangular duals of a graph have also been developed [5], [8]. This paper presents new properties of rectangular dissections and discusses two related problems: (1) the enumeration without repetition of all rectangular duals of a graph, and (2) transformations of rectangular dissections.

The problem of *rectangular dualization* is to find a dissection of a rectangle into rectangles, an *R-dissection*, that satisfies specified adjacency relations among the component rectangles. This problem has been investigated for applications to architectural design [11]–[16], and more recently to floorplan design for VLSI integrated circuits [1]–[8]. These applications usually require finding an R-dissection that has the smallest area under a set of constraints on the component rectangles. Possible constraints are area related, such as bounds on the component rectangles areas, and perimeter related, for example, on the lengths of common boundaries of adjacent rectangles. An efficient search for an optimum or near optimum R-dissection subject to these constraints, however, is dependent on an understanding of the properties of R-dissections. Some definitions, theorems and algorithms are introduced here that can be used to speed up these searches, particularly when perimeter constraints are included. In VLSI applications of rectangular dualization, perimeter constraints can be used to provide necessary interfaces for wiring adjacent functional modules of IC's.

Section II summarizes previously published results and introduces terminology used in the remainder of the paper. Section III introduces oriented, directed graphs, called *O4-completions* that represent adjacency and relative spatial positions of component rectangles in rectangular dissections. Properties of O4-completions and a method to obtain a rectangular dissection from an O4-completion representation are discussed.

One method to search for the R-dual with a particular set of desired properties is to transform an existing R-dual. Therefore, Section IV presents some additional properties of O4-completions and theorems about R-dissections transformations in the context of O4-completions. A minimal set of operations is introduced that suffices to transform any R-dual of a graph into any other R-dual of the same graph.

Alternatively, the desired R-dual of a graph can be found by generating all R-duals. The time complexity of this exhaustive search is exponential, nevertheless admissible for small problems. This problem of generating all duals of a graph is considered in Section V. One enumeration algorithm, described in Section V.1, uses transformations from Section IV to exhaustively enumerate all R-duals of a graph. A second enumeration method, more suitable for a branch-and-bound optimization of perimeter- and area-constrained R-duals is described in Section V.2.

Most of the graph-theoretic terminology used in this paper is taken from a standard graph theory text [17]. The word *rectangular* is abbreviated as R-. Proofs of Lemma 1 and Theorems 1–4 are omitted; these are available elsewhere [6].

II. Construction of a Rectangular Dual

A *plane triangulation* is a graph that has all faces triangular; it is also called a maximal planar graph. In a *plane triangulated graph* only the exterior face may have more than three edges. This section presents a characterization of plane triangulated graphs that admit R-duals.

The problem of determining the existence of an R-dual can be stated as: given a plane graph G, find if there exists a dissection D of a rectangle into rectangles such that: (1) there exists a 1-1 correspondence between the vertices of G and the rectangles in the dissection and (2) for any edge of

Manuscript received September 17, 1987. This work was supported in part by Scientific Calculations, Inc., Fishers, NY, and by the Semiconductor Research Corporation, Research Triangle Park, NC. This paper was recommended by Associate Editor M. Ilic.

K. Kozminski was with the University of Rochester, Rochester, NY. He is now with the Institute of Microelectronics and Optoelectronics, Warsaw University of Technology, 00-662 Warsaw, Poland.

E. Kinnen is with the Department of Electrical Engineering, University of Rochester, Rochester, NY 14627.

IEEE Log Number 8823451.

0098-4094/88/1100-1401$01.00

G, the rectangles corresponding to its endpoints abut. Any such dissection D is called a *rectangular dual* of G.

A theoretical characterization of graphs that have R-duals was reported in 1970 [12], however, an efficient implementation of this theory as an algorithm does not appear possible [12], [14]. An efficient feasibility verification and construction of an R-dual is possible, however, if the nodes in the R-dual D have degree at most 3, or, equivalently, the internal faces of G have degree 3. As this constraint is not severe, it is assumed for the remainder of this paper; any dissection where four rectangles meet at a point can be considered by inserting a degenerate rectangle at this point. Several polynomial time algorithms for finding R-duals have been developed to date, ranging from $O(n^2)$ [5] to $O(n)$ [8].

In a graph G, each triangle corresponds to three rectangles adjacent to each other in the R-dual of G. If G contains a triangle T that is not a face, then G has no R-dual since it is impossible to enclose a rectilinear area dual to the interior of T with only 3 rectangles. The following lemma characterizes plane triangulations in which all triangles are faces:

Lemma 1: A plane triangulation G is 4-connected[1] if and only if each simple cycle of G that is not a face has length at least four.

Consider a plane triangulation G that is dual to a cube with one face dissected into rectangles. Such graphs are fully characterized by Theorem 1, based on the notion of a ***4-triangulation***, a 4-connected plane triangulation with at least 6 vertices and at least one vertex of degree 4. Theorem 2 is a corollary of Theorem 1 and characterizes plane triangulated graphs that have R-duals.

Theorem 1: For a graph D dual to a plane graph G to have an embedding in the form of a cube with one face dissected into rectangles, no 4 of which meet at a single point, it is necessary and sufficient that G is a 4-triangulation.

Theorem 2: For a plane triangulated graph G to have a rectangular dual, it is necessary and sufficient that there exists a 4-triangulation H such that G can be obtained by deleting a vertex of H of degree 4 together with its 4 neighbor vertices.

To find an R-dual of a plane triangulated graph G, 5 additional vertices and appropriate edges are first attached to G so as to create a 4-triangulation while insuring the 4-connectedness of the resulting graph. Clearly, the construction would fail if graph G contains triangles that were not faces; such graphs are therefore excluded from further consideration. Detecting triangles that are not faces can be done in $O(n)$ time [8].

It is convenient at this point to expand the terminology that can be used to describe the construction of an R-dual [6]. See Fig. 1. A *4-completion* of a graph G is any plane graph H such that the following conditions hold: (a) the outer face of H has exactly 4 vertices of degree at least 3, (b) all internal vertices of H have degree at least 4, (c) all internal faces of H have degree 3, (d) all cycles in H that are not internal faces have length at least 4, and (e) graph H' obtained from H by deleting the vertices of the outer face of H together with the incident edges is isomorphic to G. The 4 vertices in the difference $H-G$ are called *outer vertices*. If a 4-completion H exists, then adding one more vertex and 4 edges connecting it to the outer vertices of H would create a 4-triangulation. Therefore, the problems of the existence of a 4-completion and the existence of an R-dual are equivalent.

[1] G is 4-connected if deleting less than 4 vertices preserves the connectivity of G.

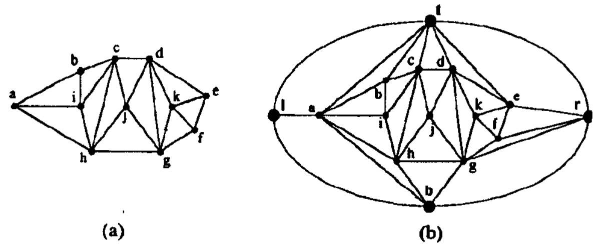


Fig. 1. Illustration of: outer vertices: r, t, l, b; corner vertices were chosen as a, e and g; outer paths: $abcde$, efg, gha; shortcuts: ck and dg (both are critical); CI-paths: $habc$ and $defg$. (a) Graph of G. (b) 4-completion of G.

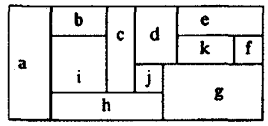


Fig. 2. Rectangular dual obtained from the 4-completion in Fig. 1(a).

In a 4-completion, the outer vertices create a cycle containing G in its interior. In the following, these vertices are always labeled t, l, b, and r, as in Fig. 1. These outer vertices have a simple interpretation: they correspond to the upper, left, bottom and right sides of the bounding rectangle of an R-dual of G; see also Fig. 2.

Assume a 4-completion of G exists. Let $v_{o_1o_2}$ denote a vertex of G adjacent to two consecutive outer vertices o_1 and o_2; the 4 possible such vertices, v_{br}, v_{bl}, v_{tl} and v_{tr} are called *corner vertices* as they correspond to the corner rectangles in any R-dual created from H. If a vertex is assigned two consecutive labels, then the face of an R-dual that corresponds to this doubly labeled vertex contains 2 corners. For example, in Fig. 2, rectangle a is dual to the vertex labeled with both v_{tl} and v_{bl} in Fig. 1 and contains the upper left and bottom left corners of the R-dual. The corner vertices divide the cycle bounding the outer face of G into $\leqslant 4$ edge disjoint paths called *outer paths*. Vertices on each outer path are connected to a common outer vertex. Graph G has an R-dual if and only if corner vertices can be chosen such that the construction of a 4-completion is possible.

Initially, consider the R-dualization of *blocks*, i.e., graphs without cut vertices. A *shortcut* in a block G is an edge that has both vertices on the outer face of G and that does not belong to the outer face. A uv path composed of the edges of the outer face of a plane block G is a *corner implying path* (CI-path) if the vertices u, v are endpoints of a uv shortcut and if this path contains no other endpoints of a shortcut; this uv shortcut is said to be a *critical shortcut*. In any 4-completion of G, at least one corner vertex must belong to the interior of each CI-path; otherwise, a non-face triangle containing a shortcut would exist. For the special case of a complete graph C_3 on 3 vertices, any two consecutive edges are a CI-path; although there are no shortcuts in C_3, each of its vertices must be a corner vertex.

Theorem 3: A plane block G with all internal faces of degree 3 and in which all cycles that are not faces have lengths ≥ 4 has a rectangular dual if and only if it has ≤ 4 corner implying paths.

A *block neighborhood graph* (BNG) for a given graph G is a graph where each vertex corresponds to each maximal block[2] of G, and two vertices are connected with an edge if and only if the corresponding blocks have a vertex in common.

A *critical CI-path* in a maximal block G_i of a plane graph G is a CI-path of G_i that does not contain cut-vertices of G in its interior.

Theorem 4: A plane graph G that has all faces of degree 3 and is not a block has a rectangular dual if and only if (1) each cycle that is not a face has length at least 4, (2) the BNG of G is a path, (3) maximal blocks corresponding to the endpoints of the BNG contain at most 2 critical CI-paths each, (4) no other maximal block contains a critical CI-path, and (5) no maximal block of G lies inside a face of some other maximal block.

Fig. 3 shows examples of failed attempts to construct a 4-completion with one of the requirements of Theorem 4 violated. Normally, if a 4-completion was possible, the edges indicated by arrows would end in outer vertices.

An algorithm for finding an R-dual of a plane graph G has been developed and implemented from Theorems 1–4 and using the notion of a valid cutting path introduced below. The algorithm is based on splitting a 4-completion recursively until a trivial 4-completion on 5 vertices is found for which an R-dual is a single rectangle; the results are then merged into an R-dual of G. This process is shown symbolically in Fig. 4.

A *horizontal* (*vertical*) *valid cutting path* C in a 4-completion H is any simple path from vertex l to vertex r (from vertex b to vertex t) such that: (a) for any two vertices v, w in C which are not adjacent in C, there is no edge in H that would be simultaneously incident to v and w and (b) deleting the vertices of C from H separates H into two graphs each containing at least 2 vertices. A special case of a valid cutting path that was used in the description of the algorithm for construction of an R-dual

[2]A maximal block of graph G is a maximal 2-connected subgraph of G.

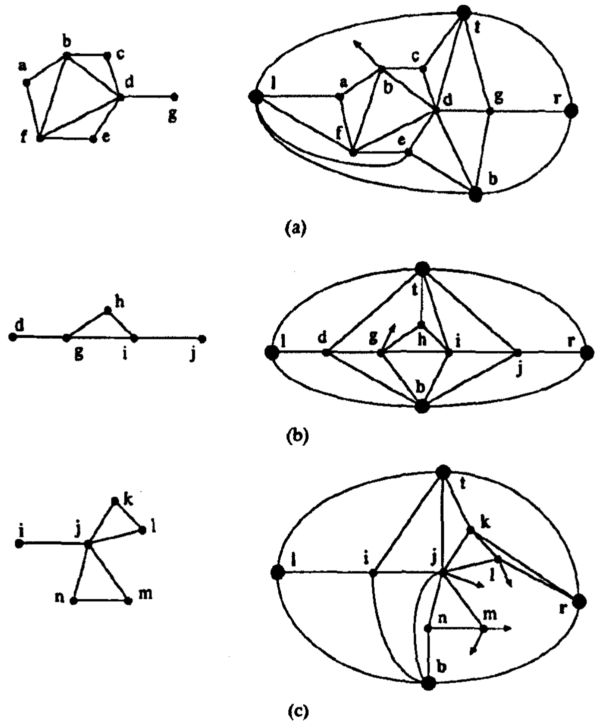


Fig. 3. Illustration of Theorem 4. Both G and the 4-completion of G are shown. (a) Three critical CI-paths in a single block. (b) Critical CI-path in a block (*ghi*) that is not an endpoint of the BNG graph. (c) BNG is not a path.

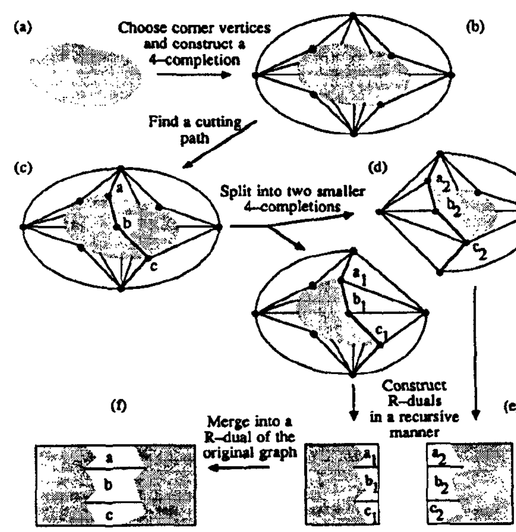


Fig. 4. Construction of a rectangular dual.

[5] is a *shortest cutting path*, i.e., a valid cutting path with the smallest possible number of edges; it can be obtained with a standard shortest path algorithm.

A cutting path P has a simple interpretation in terms of the spatial relations among the rectangles dual to vertices of P. If a vertical cutting path $v_0 = b, v_1, v_2, \cdots, v_n, t = v_{n+1}$ is used in the process of constructing an R-dual, then the rectangle dual to v_i for $i=1,2,\cdots,n$ is positioned immediately above the rectangle dual to v_{i-1}. Horizontal cutting paths possess analogous properties. Moreover, it is possible to assign dimensions to the edges in the R-dual such that the rectangles dual to v_i, $i=1,2,\cdots,n$, can be cut in a straight line [18].

The detailed description of the algorithm for finding an R-dual is given in Appendix II. The complexity of the algorithm, using shortest cutting paths, is $O(n^2)$, where n is the number of vertices in the dualized graph. In practice, almost linear running times have been observed. A recently proposed algorithm implicitly uses cutting paths with the property that each vertex on a cutting path is connected with an edge to some vertex on an outer face; the complexity of this algorithm is $O(n)$ [8].

III. Representation of Rectangular Dissections

Assume a 4-completion H of a graph G and a rectangular dual of G. An *oriented, directed 4-completion* (O4-completion) is obtained from H by assigning horizontal and vertical orientations to all its edges except for those on the outer cycle. Also, each edge is assigned a direction from one of its vertices to the other. A horizontal or vertical edge directed from v_i to v_j is denoted by $(v_i \to v_j)_h$ or $(v_i \to v_j)_v$, respectively. The orientations represent horizontal and vertical adjacencies, while the directions represent a left-to-right or bottom-to-top order of the rectangles dual to the vertices of the 4-completion.

A *partially ordered 4-completion* (PO4-completion), defined when some edges are unoriented, can be used to represent a set of R-dissections.

III.1. Basic Considerations

An O4-completion H corresponding to a given R-dissection D can be created readily. For every rectangle v in the dissection there is a corresponding vertex v in the O4-completion. Whenever two rectangles v_i, v_j are adjacent in D, the direction and orientation of the edge joint vertices v_i and v_j in H can be determined as follows: if rectangle v_i is to the left, right, below or above of rectangle v_j, make the edge $(v_i \to v_j)_h$, $(v_j \to v_i)_h$, $(v_i \to v_j)_v$ or $(v_j \to v_i)_v$, respectively. The four outer vertices r, t, l, b are added together with edges $(b \to v)_v$ for each rectangle v adjacent to the lower side of the bounding rectangle of the dissection, and, similarly, edges $(v \to t)_v$, $(l \to v)_h$, $(v \to r)_h$ for the other rectangles adjacent to the upper, left and right sides. An example of an O4-completion is shown in Fig. 5. Vertically oriented edges are drawn with heavy and horizontal edges with thin lines, a convention used throughout the remainder of the paper.

Any two R-duals D_1 and D_2 of some graph G that have identical sets of the corner faces also have identical unoriented 4-completions. If D_1 and D_2, however, differ in terms of spatial relations among the component rectangles, the O4-completions of G corresponding to D_1 and D_2 will have different orientations and/or directions of some

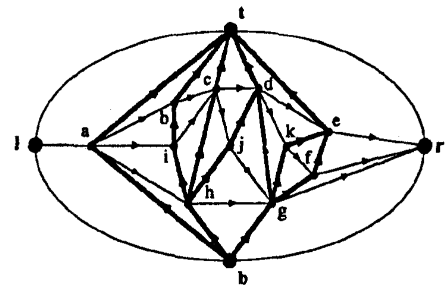


Fig. 5. O4-completion representation of a dissection from Fig. 2.

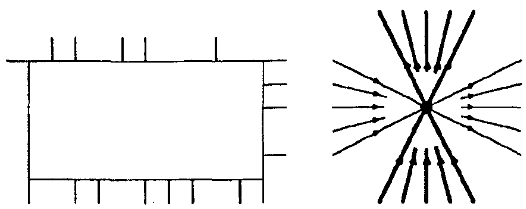


Fig. 6. Illustration of the condition of well-formed vertices.

edges. Not all arbitrary orientations of edges in a 4-completion have corresponding R-dissections. In general, an O4-completion where all edges are assigned some direction and orientation represents either a unique R-dissection or no feasible arrangement of rectangles. A PO4-completion H where only a subset E' of edges is oriented can be interpreted as representing a set of R-dissections corresponding to all R-feasible O4-completions that agree with H on the orientation of edges in E'. In particular, an unoriented 4-completion represents all possible R-duals with fixed corner faces of some graph. A PO4-completion H of some graph G is called *R-feasible* if it represents a non-empty set of R-duals of G, i.e., the unoriented edges in H can be assigned orientations and directions corresponding to spatial relations in some R-dissection.

Assigning a direction and orientation to some unoriented edge of a given PO4-completion may have a variety of effects, ranging from creating an infeasible orientation through decreasing the size of the set represented by the PO4-completion to no effect at all. To determine which of these possibilities occurs for some particular orientation assignment, necessary conditions must be analyzed for a partial orientation to be R-feasible. The results discussed here resemble those obtained elsewhere but were developed independently [12], [15].

Consider the orientations of all edges incident to some vertex v of graph G in an O4-completion representing an R-dual of G. It can be seen that the clockwise sequence of these edges is composed of 4 subsequences: vertical edges directed into v, followed by horizontal edges directed into v and then vertical and horizontal edges directed from v. This is illustrated in Fig. 6. This condition has been called a *condition of well-formed vertices* [12].

A *tentatively well-formed vertex* is defined as a vertex v in a PO4-completion that is adjacent to some unoriented edges, such that the sequence of the edges around v can create a well-formed vertex after an appropriate assignment of directions to the unoriented edges. A tentatively well-formed vertex is easily recognizable, in time proportional to the degree of this vertex, by a trial assignment of the missing orientations and directions to the edges around vertex v so that it becomes well formed.

A fully (partially) oriented 4-completion H of a graph G is called a *well-formed* (*tentatively well-formed*) 4-completion if the following conditions hold: (1) H is acyclic, (2) all oriented edges incident to b, l, t, and r are of the form $(b \rightarrow v)_v$, $(l \rightarrow v)_h$, $(v \rightarrow t)_v$, and $(v \rightarrow r)_h$, respectively, and (3) all vertices of G are well formed (tentatively well formed).

It is evident that for a PO4-completion H of a plane graph G to represent a non-empty set of R-duals of G, it is necessary that H is tentatively well formed. However, tentative well-formedness of H is not sufficient for R-feasibility. Also, characterizing R-feasible PO4-completions does not appear to be straightforward. On the other hand, well-formedness of vertices in an O4-completion H is both necessary and sufficient for R-feasibility, as shown below in Theorem 5.

Since the conditions of tentative well-formedness can be checked in time proportional to the number of edges in a graph, they can be used with Algorithm 3 in Section V.6 to speed up the enumeration of R-duals without increasing the computational complexity of the algorithm.

III.2. Properties of Well-Formed Oriented 4-Completions

The following lemmas are used in the proof of Theorem 5:

Lemma 2: In any well-formed oriented 4-completion H of a plane triangulated graph G, there is no directed cycle containing only vertical or only horizontal edges.

Proof: Contrarily, suppose that there exists a well-formed O4-completion H of a plane graph G that contains cycles with identically oriented edges. Choose such a cycle C that contains the least number of faces in its interior. Without a loss of generality suppose that C consists of vertical edges directed clockwise with respect to its interior. Each of the vertices of C has at least one incoming and at least one outgoing vertical edge; therefore, none of the outer vertices t, r, b, l lies on the cycle. Pick any vertex v on C. From the well-formedness of v, there is at least one edge e_0 that is oriented horizontally, begins in v and lies in the interior of C. Construct a path P consisting of only horizontal edges that starts with e_0 by selecting any horizontal edge e_1 that begins in the endpoint of e_0, then any horizontal edge e_2 that begins in the endpoint of e_1, etc. Note that P cannot reach any of the outer vertices before crossing C, and none of the vertices of P can occur twice during the selection as then a cycle would be created with less faces in its interior than C. Therefore, some edge e_i will have its endpoint in some vertex w belonging to C. Since vertex w then would not be well formed, the assumption about the existence of an oriented cycle C in a well-formed H must be false (see Fig. 7). ■

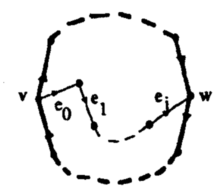


Fig. 7. The configuration after a sufficient number of steps of the selection procedure.

Lemma 3: In a well-formed 4-completion H of a plane graph G, every horizontal (vertical) edge lies on a directed path starting in $l(b)$ and ending in $r(t)$.

Proof: Apply the selection procedure described in Lemma 2 to any horizontal edge e in H. Due to the lack of oriented cycles in H, the procedure yields a path P_1 ending in r, as this is the only vertex in a well-formed 4-completion that has only incoming horizontal edges. The same procedure carried out in the reverse direction of edges yields a path P_2 ending in l. Concatenation of P_1 and P_2 is a directed lr path containing e. ■

Lemma 4: In a well-formed 4-completion H of a plane graph G, for any two vertices v, w on any directed path P starting in $l(b)$ and ending in $r(t)$, either there is no edge vw or edge vw belongs to P.

Proof: Contrarily, suppose there is a path P in H and two vertices v, w such that edge vw exists and is not in P, and that v precedes w in P. Denote the part of P starting in v and ending in w by P_{vw}. From all such paths P and associated vertices v and w, choose a path such that the unoriented cycle C created by edge vw together with P_{vw} has the least number of faces in its interior. Without a loss of generality, let P contain horizontal edges.

Consider the triangular face vwx belonging to the interior of C. Vertex x cannot belong to P as either (a) the cycle consisting of xw and the part of P between x and w would contain fewer faces than C, (b) the cycle consisting of vx and the part of P between v and x would contain fewer faces than C, or (c) P_{vw} would consist of two edges vx and xw with vertex x being not well formed. Therefore, x lies in the interior of C. Since x is well formed, there is an edge e incident to x that is oriented horizontally. From Lemma 3, there is a $l-r$ path P_e passing through e. P_e intersects P_{vw} at two vertices y and z, as in Fig. 8. Depending on the orientation of P_e, either the $y-z$ part of P_e creates a directed cycle with the $y-z$ part of P_{vw}, contrary to Lemma 2, or the $v-y$ part of P_{vw}, the $y-z$ part of P_e and the $z-w$ part of P_{vw} together with edge vw create a contradictory cycle C' with fewer faces than C. In either case, the contradictory assumption about the existence of edge vw cannot be supported ■

Theorem 5: For a fully oriented 4-completion H of a plane graph G to represent a rectangular dual of G, it is necessary and sufficient that H is well formed.

Proof: Necessity follows immediately from the discussion in the initial part of Section III. Sufficiency can be

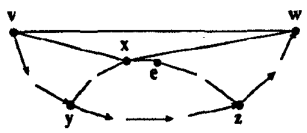
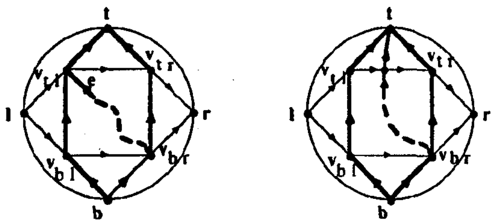


Fig. 8. The configuration of the contradictory cycle C.



(a) (b)

Fig. 9. A cutting path passing through a corner vertex: a) all outer vertices have degree 4, b) one of the outer vertices has degree ≥ 5.

proved by contradiction resembling a proof given for Theorem 1 [6].

Let H be the smallest well-formed O4-completion with no R-dissection corresponding to it. There are 3 possibilities.

Case 1: If an outer vertex of H, say r, has degree 3, then deleting r together with the adjacent edges from H produces a well-formed 4-completion H'.[3] Being smaller than H, H' has a corresponding R-dissection. Abutting the right side of this dissection with a rectangle creates an R-dissection corresponding to H.

Case 2: If all outer vertices of H have degree 4, then the well-formedness of H implies the existence of edges $(v_{bl} \to v_{tl})_v$, $(v_{br} \to v_{tr})_v$, $(v_{tl} \to v_{tr})_h$, $(v_{bl} \to v_{br})_h$. One of the corner vertices, say v_{tl}, must have degree at least 5, for otherwise the corner vertices would create a face of degree 4. One of the edges incident to v_{tl} belongs to two faces, neither of which contains an outer vertex. Without a loss of generality, let this edge, say e, be vertical. Choose any vertical path P from b to t passing through e. P cannot pass through v_{bl} or v_{tr}, as either edge $(v_{bl} \to v_{tl})_v$ or $(v_{tl} \to v_{tr})_h$ together with P would contradict Lemma 4. It must then pass through v_{br}. Thus, the unoriented cycle consisting of P and edges bl and lt contains vertex v_{bl} inside, and the cycle created by P together with edges br and rt contains vertex v_{tr} inside; see Fig. 9. In the manner shown in Fig. 4, P can be used as a cutting path to split H into two smaller well-formed 4-completions H_1 and H_2, and the R-dissections corresponding to H_1 and H_2 can be combined to create an R-dissection corresponding to H. Using Lemma 4, it can be shown that no triangles that are not faces are created during the construction of the smaller 4-completions.

Case 3: If one of the outer vertices, say t, has degree at least 5 then none of the edges incident to t is vertical and not incident to any corner vertex; the path from b to t that passes through this edge and separates v_{tl} from v_{tr} can be used in a similar manner as path P above. ∎

[3]Note that H contains at least 6 vertices, as otherwise it would be a 4-completion on 5 vertices and a plain rectangle would be the corresponding R-dissection. Consequently, H' has at least the 5 vertices necessary for it to be 4-completion.

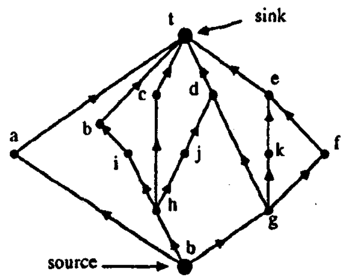


Fig. 10. Subgraph H_v of the O4-completion in Fig. 5.

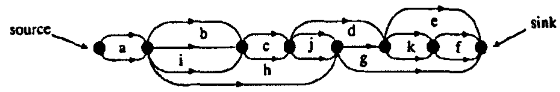


Fig. 11. Digraph D_v dual to H_v in Fig. 10.

III.3. Drawing of an R-Dissection from an Oriented 4-Completion

Coordinates of the endpoints of all vertical and horizontal segments must be known to draw an R-dissection. O4-completion by itself provides only the information about the directions of the segments. However, minimal required lengths of all segments can be easily represented by assigning their values as weights to the edges of the O4-completion dual to the segments. If rectangle dimensions are not important, e.g., when one wants only to obtain any R-dual of a graph, unit weights can be assumed.

Consider *vertical* and *horizontal subgraphs* H_v and H_h of an O4-completion H, containing, respectively, only those edges that are oriented vertically and horizontally. Subgraph H_v of the O4-completion from Fig. 5 is shown in Fig. 10.

Subgraphs H_v and H_h are directed, acyclic subgraphs with one source and sink, known also as *polar digraphs*. Their edges have weights representing minimal lengths ascribed to vertical and horizontal edges. Therefore, $H_v(H_h)$ can be used to determine distances between horizontal (vertical) edges by considering the graph dual to $H_v(H_h)$. Dualization of polar digraphs can be done by temporarily adding an edge e_{ss} from the sink to the source, performing the usual dualization and then deleting the edge dual to e_{ss}. The resulting polar digraph $D_v(D_h)$ is isomorphic to the graph obtained from the R-dissection by contracting to vertices all vertical (horizontal) segments and treating the remaining segments as graph edges. An example digraph D_v dual to H_v is shown in Fig. 11; refer to Fig. 2 for the corresponding R-dissection. Vertices of the digraph $D_v(D_h)$ represent maximal segments of the R-dissection while their edges, whose weights are identical to the weights of their duals from $H_v(H_h)$ represent requirements for minimal separations between the segments.

Finding minimal values of the coordinates that conform to the separation requirements can be done by obtaining maximal length paths from the source to all other vertices of a digraph; an $O(n)$ algorithm to accomplish this and to draw the resulting R-dissection is straightforward.

IV. Transformations of Rectangular Duals

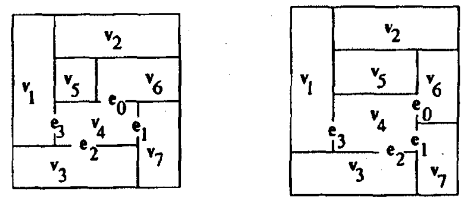
IV.1. Turnable Structures in Rectangular Dissections

Consider the R-dissections in Fig. 12(a) and (b) which depict two distinct R-duals of the same graph; the spatial relations among the component rectangles are identical except for the relative positions of rectangles v_4 and v_6. Edge e_0 separating these two rectangles is an example of a turnable structure (T-structure); this name indicates that a homeomorphic deformation of a plane which transforms one of the pictured morphologies into the other can be visualized as turning edge e_0 in Fig. 12(b) in the counterclockwise direction. In Fig. 12(b), edge e_0 is said to be a left-turnable structure, while in Fig. 12(a) this edge e_0 is right-turnable.

Formally, a *left-turnable structure* (LT-structure) in an R-dissection D_1 is a set of edges E such that there exists another R-dissection D_2 where (1) both D_1 and D_2 are R-duals of the same graph, (2) D_1 and D_2 have identical sets of corner faces, (3) relative positions of any two rectangles in D_1 and D_2 separated with an edge not from E are identical, and (4) if an edge e from E separates rectangle v_i from rectangle v_j that is placed above (to the left of) v_i in D_1, then in D_2 rectangle v_j is placed to the left of (below) v_i. *Right-turnable structures* (RT-structures) are defined analogously except for the condition (4) which would read: if e separates rectangle v_i from rectangle v_j that is placed above (to the right of) v_i in D_1 then in D_2 rectangle v_j is placed to the right of (below) v_i. Thus D_2 can be obtained from D_1 by applying a turn to E in D_1, denoted as $T_E(D_1) = D_2$, where the transformation T is said to be implied by E. Given an R-dissection D and a T-structure E in it, the left or right direction of the turn is determined uniquely by the context of E within D.

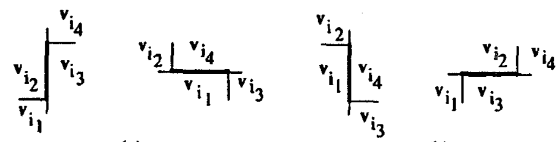
If the O4-completions describing dissections D_1 and D_2 are H_1 and H_2 and $T_E(D_1) = D_2$, transformation T_E can be interpreted in terms of changing the orientations and directions of edges in H_1 and H_2; this can be also written as $T_E(H_1) = H_2$. Let E^* be a set of edges in H_2 that correspond to the set of edges E in H_1. If the edges in D_1 that are dual to E are an RT-structure, then the transformation T_E has the following effect on the edges in E^*: (1) if some vertical edge $e = (v \to w)_v \in E$, then in E^* the corresponding edge is oriented horizontally and directed from v to w, $e = (v \to w)_h \in E^*$; and (2) if $e = (v \to w)_h \in E$, then in E^* this edge is vertical and has the reverse direction, $e = (w \to v)_v \in E^*$. An analogous rule can be stated for LT-structures.

A single edge is a T-structure iff it occurs in any one of the 4 configurations shown in Fig. 13, called *alternating-T configurations* [2]. An example of a T-structure consisting of more than one edge is the set $E_1 = \{e_0, e_1, e_2, e_3\}$ in Fig. 12(a). E_1 is an RT-structure that after the appropriate transformation yields the R-dissection in Fig. 14(a). Set E_2 consisting of the four edges bounding rectangle v_5 in Fig. 14(a) can be used to obtain the dissection in Fig. 14(b). Finally, $E_3 = E_1 \cup E_2$ is an RT-structure in Fig. 12(a) that can be used to obtain the dissection in Fig. 14(b).



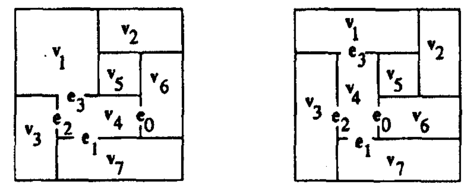
(a) (b)

Fig. 12. Turnable structures in R-dissections.



(a) Right-turnable (b) Left-turnable

Fig. 13. Alternating-T configurations. Turnable edges are marked with heavy lines.



(a) (b)

Fig. 14. Additional examples of T-structures in R-dissections.

An important difference exists between the T-structures E_2 and E_3 or E_1 in Figs. 12 and 14. There are exactly 4 edges that are adjacent to E_2 while T-structures E_1 and E_3 each have more than 4 adjacent edges. The property of having exactly 4 adjacent edges occurs also for T-structures consisting of a single edge and has some interesting implications that are discussed below in Theorems 6–8. Because of this property, single-edge T-structures and T-structures similar to E_2 are called simple T-structures. Formally, a *simple left- or simple right-turnable structure E* in an R-dissection D is defined as a T-structure for which there are exactly 4 edges in D that do not belong to E but share endpoints with E.

Simple T-structures are of importance because (a) the number of all T-structures can be significantly larger than the number of simple T-structures,[4] (b) simple T-structures can be determined in a straightforward manner, and

[4] It is easy to exhibit an R-dissection where the set of all RT-structures is the power set of all simple RT-structures.

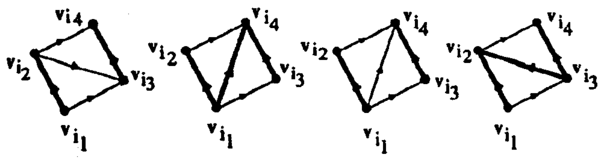


Fig. 15. Single-edge simple T-structures in 4-completions.

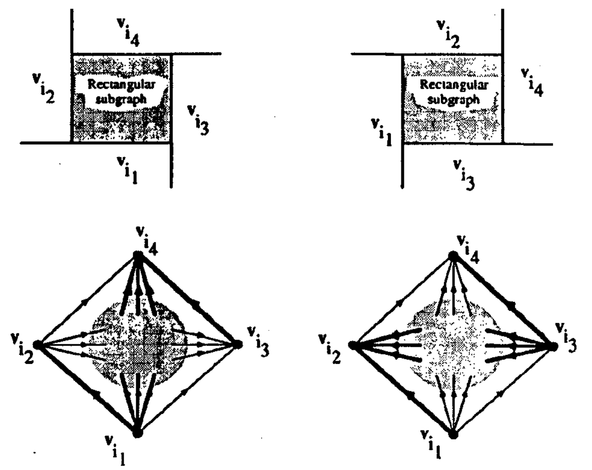


Fig. 16. Multiple-edge simple T-structures.

(c) for any two R-duals of a graph that have identical sets of corner faces, either of these duals can be obtained from the other by applying transformations implied by simple T-structures. The following theorem refers to property (b).

Theorem 6: For a set E of edges in an R-dissection D to be a simple T-structure, it is necessary and sufficient that the subgraph E^* consisting of edges that are dual to E in the oriented 4-completion H representing D is the subgraph contained in the interior of some unoriented cycle C on four vertices $v_{i_1}, v_{i_2}, v_{i_3}, v_{i_4}$, where the orientations and directions of the edges are: $(v_{i_1} \rightarrow v_{i_2})_v$, $(v_{i_1} \rightarrow v_{i_3})_h$, $(v_{i_3} \rightarrow v_{i_4})_v$, $(v_{i_2} \rightarrow v_{i_4})_h$.

Proof: As H is a 4-triangulation, cycle C either contains a single edge in its interior or it contains at least one vertex inside.

If E is a single edge then from nodes $v_{i_1} - v_{i_4}$ being well-formed, the only 4 possible configurations for this edge are those shown in Fig. 15. The corresponding parts of the R-dissections are exactly the set of alternating-T configurations from Fig. 13. Both the necessity and sufficiency conditions stated in the theorem are evident.

For the necessity part of the theorem when C contains at least one vertex inside, note that the part of D dual to the interior of C must be an R-dissection itself, as it is contained in an area bounded by the 4 rectangles dual to the vertices $v_{i_1}, v_{i_2}, v_{i_3}, v_{i_4}$. The only two possibilities for an R-subgraph to be a T-structure are those shown in Fig. 16.

For the sufficiency, suppose there is a cycle C on 4 vertices $v_{i_1}, v_{i_2}, v_{i_3}, v_{i_4}$ in H with edges in this cycle oriented as stated in the theorem. Suppose that some edge e incident to v_{i_1} that lies inside C is oriented vertically; well formedness of v_{i_1} implies that e is directed from v_{i_1}. The vertical $b-t$ path P_1 passing through e has to pass through v_{i_4} so as not to violate Lemma 4. If there was a vertical edge f incident to v_{i_2} inside $C, b-t$ path P_2 passing through f would intersect P_1 and $P_1 \cup P_2$ would contain another path that would violate Lemma 4. Therefore, if there is a vertical edge incident to v_{i_1} inside C, all edges incident to v_{i_2} inside C must be horizontal. Extending this reasoning for the other orientations and vertices and considering the well formedness of the vertices of C results in the conclusion that the appearance of the edges inside C must be one of the two variants shown in Fig. 16. Then, the corresponding parts of R-duals must also appear as shown and create a simple T-structure. In both cases shown in Fig. 16, cycle C containing a simple T-structure in its interior can be seen to fulfill Theorem 6. ■

From Theorem 6 it follows that finding simple T-structures in an R-dissection is a task that can be accomplished in polynomial time by examining an O4-completion representation of this dissection and finding cycles of length 4 with appropriately oriented edges. This problem of finding simple T-structures is discussed in Section IV.2.

A *leftmost* (*rightmost*) *R-dual* of graph G is defined as an R-dual of G that contains no left (right) T-structures.

The proofs of the following two theorems are given in Appendix I.

Theorem 7: For any R-dual D of a graph G there exists a unique leftmost (rightmost) R-dual of G having the same set of corner faces as D.

Theorem 8: For the leftmost (rightmost) R-dual D_0 of a graph G and any R-dual D of G having the same set of corner faces as D_0, there exists a sequence of right (left) simple T-structures E_i and R-dissections D_i such that $T_{E_0}(D_0) = D_1, \cdots, T_{E_i}(D_i) = D_{i+1}$, $T_{E_n}(D_n) = D$.

Corollary: For any two R-duals D', D'' of a graph G having the same set of corner faces, there exists a sequence of simple LT- or RT-structures E_i and rectangular dissections D_i such that $T_{E_0}(D_0) = D_1, \cdots T_{E_i}(D_i) = D_{i+1}$, $T_{E_n}(D_n) = D''$.

IV.2. Finding all Simple Turnable Structures in an Oriented 4-Completion

One method to find all simple LT- and RT-structures in an O4-completion H of a plane graph G is to find all cycles having length 4 in H (*4-cycles*) and then, using Theorem 6 and the directions and orientations of the edges of H, uncover those 4-cycles that bound LT- or RT-structures. If T-structures have to be found in several different O4-completions of the same graph, this 2-stage procedure is particularly suitable, as it is then useful to have a list of all 4-cycles that potentially may bound a T-structure. However, some 4-cycles can never bound a T-structure and can be eliminated from the list. In any 4-cycle C that contains an outer vertex of H, the two edges incident to the outer vertex are oriented in the same manner in all well-formed 4-completions. For example, all edges adjacent to t are vertical and directed into t and can never

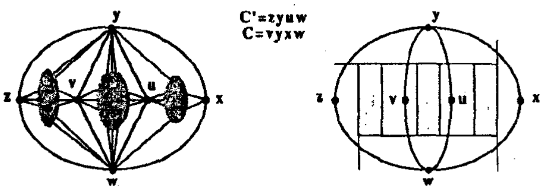


Fig. 17. Two intersecting 4-cycles.

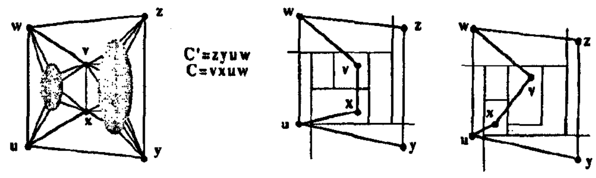


Fig. 18. Two 4-cycles sharing an edge.

conform to the requirements of Theorem 6. Another example is any 4-cycle C that lies completely inside another 4-cycle C' and shares a vertex with C'. The vertices of C' must be dualized into rectangles bounding a rectangular area and therefore in any well-formed 4-completion the subgraph consisting of C' together with its interior is also a well-formed 4-completion. The 4-cycle C described above exemplifies one class of 4-cycles that never bound T-structures.

Let two edges of a 4-cycle C lie in the interior of another 4-cycle C' that intersects C at two non-adjacent vertices and let these vertices also be non-adjacent in C'. C typifies a second class of 4-cycles that can never bound a T-structure. Referring to Fig. 17, it can be seen that vertices of C' dualize into four rectangles bounding a rectangular area that contains at least the rectangle dual to one of the vertices of C; this implies that two consecutive edges wv, vy of C are always oriented so that they create a directed path and are both vertical or horizontal, thus never complying with Theorem 5. Hence this C never bounds a T-structure.

Let a 4-cycle C share an edge with another 4-cycle C' and let C lie in the interior of C'. This C belongs to a third class of 4-cycles that can never bound a T-structure. The reasoning is similar to the previous case. Vertices of C' dualize into rectangles bounding a rectangular area containing at least two rectangles dual to the vertices of C disjoint from C'. By examining possible configurations of rectangles that are dual to the vertices of C and C', it can be seen that two consecutive edges of C must be both vertical or horizontal and create a directed path. In Fig. 18, either xv and vw have to be vertical or ux and xy have to be horizontal.

An algorithm for finding all 4-cycles and removing those falling into the three classes described above is straightforward [18]. The remaining cycles are said to be *potentially turnable structures* (PT-structures).

IV.3. Limit on the Number of Simple T-Structures in a Graph

In addition to being easily recognizable, T-structures have another attractive feature. The number of all PT-structures is $O(n)$, where n is the number of edges in the 4-completion. After discarding all 4-cycles belonging to the three classes discussed above, it can be seen that a single edge can belong to at most four PT-structures. In Fig. 19, an example configuration maximizing the number of 4-cycles containing edge ab was constructed by starting with 4-cycle $abcd$. This cycle cannot be contained in the interior

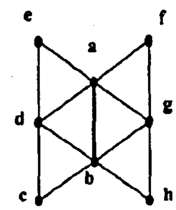


Fig. 19. Edge ab belongs to 4 PT-structures: $abcd$, $abde$, $abhg$, and $abgf$.

of any other 4-cycle passing through a and b, as $abcd$ would then belong to either the second or the third class of non-PT-structures. If $abcd$ contained any vertices in its interior, no other PT-structure could include an interior vertex of $abcd$ without falling into one of these two classes. If $abcd$ does not contain any vertices in its interior, it contains a single edge, say bd, and another PT-structure $abde$ is possible.[5] In a similar manner, PT-structures $abhg$ and $abgf$ can be constructed. This exhausts the possibilities of adding PT-structures containing ab.

V. Enumeration of Rectangular Duals

V.1. Enumeration of Rectangular Duals — Method 1

This section describes the first of two algorithms for the enumeration of R-duals. It is faster than the second method but is less suited for applications to the area minimization of R-duals. For examples having 40–60 component rectangles, an implementation of this first algorithm written in C language generated approximately 600 different R-duals per second of CPU time on a VAX 11/750.

The entire algorithm has two separate and conceptually distinct levels: (1) generating all possible 4-completions of a graph, and (2) generating all possible orientations of each 4-completion. Note that each R-dual can have eight possible renditions, each of which can be obtained from the others by the appropriate mirroring and rotating transformations.

All possible 4-completions of a graph can be found based on Theorems 1–4 in Section II by generating all distinct sets of corner vertices such that at least one labeled vertex belongs to each corner implying path [18]. The detailed algorithm is given in Appendix III.

Algorithm 1 performs the enumeration, based on Theorems 6–8 that provide for its correctness. Data for Algorithm 1 includes the leftmost dual D_L of a graph G. D_L can

[5] Note that if edge ce existed, neither $abcd$ nor $abde$ would be PT-structures; they would fall into the second class of non-PT-structures due to the interaction with 4-cycle $abce$.

be found by starting with any R-dual of a graph G and applying transformations implied by simple LT-structures until no LT-structure remains. Since D_L is unique, the transformations can be applied in any order.

Algorithm 1 uses simple T-structures to introduce local changes in the relative positions of some adjacent rectangles in the dissection, while leaving the majority of the spatial relations among the rectangles undisturbed. The corner rectangles in the dissection remain in the corners of the bounding rectangle. Repeating Algorithm 1 for each 4-completion of a graph results in the enumeration of all R-duals.

Algorithm 1 acts similar to a branch-and-bound (B&B) paradigm. Any R-dual can be interpreted as representing not only itself, but also all other duals that can be obtained from it by applying a sequence of transformations implied only by RT-structures; in this context, the leftmost dual D_L represents all possible R-duals. The branching is done by separating the set of R-duals represented by a given D into two disjoint subsets. Suppose that D contains some RT-structures. Pick any of these structures, say C. The set of all R-duals that can be obtained from D can be subdivided into those that require a transformation T_C implied by C and those for which C is not to be turned. This interpretation of branching shows that each dual will be generated exactly once.

In addition to the O4-completion H corresponding to the leftmost R-dual of G, the input data to Algorithm 1 include a list of PT-structures in H. Each edge of H is associated with the list contained within their bounding cycles. All PT-structures are initially flagged as free.

V.2. First Method of R-Dual Enumeration (Algorithm 1)

Step 1:

Find all RT-structures on the list of PT-structures and put them on a list L.

Step 2:

Find an RT-structure $C \in L$ that is flagged free. If no such structure exists, output the orientations of edges in the 4-completion and terminate the current application of the algorithm. Otherwise, perform transformation T_C by changing the orientation of all horizontal edges in $L(C)$ to vertical, and by reversing the direction of all vertical edges in $L(C)$ and changing their orientation to horizontal. While scanning $L(C)$, for each edge $e \in L(C)$ mark the PT-structures containing e in their 4-cycle.

Step 3:

Based on Theorem 6, determine the turnability status of each PT-structure marked in Step 2, and append those that became right-turnable to the list L. Apply Algorithm 1 recursively.

Step 4:

Undo the transformation that was done in Step 2. Flag C as not free and apply Algorithm 1 recursively.

V.3. Complexity of the First Method of R-Dual Enumeration

During the enumerating of the R-duals, the complexity of generating a single dual is of more interest than the complexity of the entire enumeration process, as the number of rectangular duals of a graph can be exponential in terms of the graph size. All operations in Algorithm 1 can be done in time proportional to the number of edges in a graph; these operations involve scanning the list of PT-structures in steps 1 and 3, changing orientation of edges in the T-structures and outputting the O4-completion in steps 2 and 4. Based on the knowledge of orientations and directions of edges in an O4-completion, a drawing of a rectangular dissection can be produced in time proportional to the number of edges in the 4-completion. Since in planar graphs the numbers of edges and vertices are linearly related, the complexity of generating a single rectangular dual of a graph is $O(n)$, where n is the graph size defined as the total number of edges and vertices.

V.4. Enumeration of Rectangular Duals — Method 2

Method 2 comprises two algorithms. The second of the two, presented in Section V-6, is equivalent to Algorithm 1, in that it generates all R-duals of a graph that have some particular set of corner faces. Unlike Algorithm 1, Algorithm 3 can be easily extended for a more efficient area minimization of R-duals than simple enumeration [18].

The algorithm presented here employs the idea of an assignment of orientations to the edges of a PO4-completion in a B&B manner, starting initially with a 4-completion with all edges unoriented. The basic concept is to repreatedly partition the set of all possible R-duals represented by a PO4-completion. This partitioning is achieved by assigning orientations and directions to the unoriented edges. At each partitioning stage, a PO4-completion H obtained from the preceding partition phase is considered. Several different sets of unoriented edges are chosen in H together with some particular orientations for the edges in these sets. Each set and orientation yields a new PO4-completion H_i that has more oriented edges than H. The sets and assigned orientations are chosen in such a way that all new PO4-completions $H_1, H_2, \cdots, H_n$ together represent the same set of R-duals as H. Also, the sets represented by $H_i, H_j, i \neq j$, are disjoint, due to the choice of the orientations such that for any H_i, H_j there exists some edge e that is oriented differently in H_i than in H_j. Thus relative positions of some pair of rectangles in all the R-duals represented by H_i are different from the relative positions of the same pair of rectangles in any R-dissection represented by H_j.

The process of selecting the unoriented sets, assigning orientations, and extending the B&B tree is repeated until either (1) some PO4-completion is proved not to be R-feasible or (2) a 4-completion becomes fully oriented. In the second case, the R-feasibility of the O4-completion can be determined by verifying its well-formedness, and then a new R-dissection is added to the set of already known R-duals. In either case, the B&B node containing the particular 4-completion is fathomed.

A similarity exists between the branching process in Method 2 and the Algorithm A-2 used for finding an

R-dual of a graph in Appendix II. Algorithm A-2 essentially follows the same idea of assigning new orientations to unoriented edges. However, in A-2 the set of R-duals represented by the current PO4-completion is not partitioned into many subsets but rather one subset is selected and all others discarded. The process of finding an R-dual of a graph using Algorithm A-2 consists of assigning directions and orientations to the edges chosen so as to create a valid cutting path, thus insuring at all times that PO4-completions remain R-feasible. The initial unoriented 4-completion can be considered to represent all possible R-duals. Each choice of a cutting path selects a subset of the R-duals represented by the current PO4-completion until an O4-completion representing a single R-dual is obtained.

In the algorithm presented below, edges to be assigned orientations at each stage of the B&B process need not form valid cutting paths; in fact, they can be chosen in an arbitrary sequence. Consequently, some PO4-completions may represent empty sets of R-duals. It is in the interest of algorithm speed, however, that these representations be eliminated quickly after their introduction in the branching tree. In Step 6 of Algorithm 2, selecting the newly oriented edges so as to form cutting paths with edges that are already oriented can be used intuitively to increase the number of R-feasible PO4-completions in the B&B.

It should be noted that for a 4-completion with n edges, the number of possible different edge orientations is 4^n. As many of these orientations will violate the condition of well-formed nodes and therefore will be not R-feasible, it is important to devise the algorithm so that the sets containing no R-feasible orientations can be discovered efficiently.

Algorithm 2 describes the processing of a single node in a B&B tree. This algorithm is then applied in a conventional B&B manner in Algorithm 3, resulting in the enumeration of R-duals. The data supplied to Algorithm 2 consist of the triplet $(\boldsymbol{H}, \boldsymbol{E}, \boldsymbol{O})$, where $\boldsymbol{H}$ is a tentatively R-feasible PO4-completion, $\boldsymbol{E} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_n\}$ is a set of unoriented edges of $\boldsymbol{H}$, and $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \cdots, \boldsymbol{o}_n\}$ is a set of orientations to be assigned to the edges in $\boldsymbol{E}$. Each $0 \leqslant \boldsymbol{o}_i \leqslant 3$ is an integer denoting one of four possible orientations of edge $\boldsymbol{e}_i$.

Data for Algorithm 3 in Section V.6 is a description of an unoriented 4-completion. The result of Algorithm 3 is all sets of orientations of edges in the possible O4-completions. The information about the horizontal/vertical orientation of each edge in a 4-completion, together with the unoriented 4-completion, is sufficient to create an R-dissection.

V.5. Processing of a Single Branch-and-Bound Node (Algorithm 2)

Step 1:

Set $\boldsymbol{V} = \varnothing$. $\boldsymbol{V}$ denotes a set of vertices that may lose well-formedness due to some edge orientation assignment.

Step 2:

For each edge $\boldsymbol{e}_i = (\boldsymbol{v}, \boldsymbol{w}) \in \boldsymbol{E}$, assign the orientation given in $\boldsymbol{O}$. Set $\boldsymbol{V} = \boldsymbol{V} \cup \{\boldsymbol{v}, \boldsymbol{w}\}$.

Step 3:

Set $\boldsymbol{U} = \boldsymbol{V}$ and $\boldsymbol{V} = \varnothing$. Here $\boldsymbol{V}$ denotes a set of vertices that may become not well-formed due to some edge orientation assignment obtained in step 4.

Step 4:

Find edges that have only one possible orientation by repeating the following two steps for each $\boldsymbol{v} \in \boldsymbol{U}$: (1) find edges incident to $\boldsymbol{v}$ for which only a single orientation is possible if well formedness is to be achieved, (2) for each edge $\boldsymbol{e} = (\boldsymbol{v}, \boldsymbol{w})$ found in (1), set $\boldsymbol{V} = \boldsymbol{V} \cup \{\boldsymbol{v}, \boldsymbol{w}\}$ and assign the appropriate orientation. If $\boldsymbol{H}$ it not tentatively well formed, fathom the current B&B node and exit the algorithm. If $\boldsymbol{V} \neq \varnothing$, go to step 3, otherwise proceed to Step 5.

Step 5:[6]

If some edges of $\boldsymbol{H}$ are not oriented, go to Step 6. Otherwise, $\boldsymbol{H}$ is a well-formed representation of some R-dissection. This dissection is a new R-dual to be added to the set of all R-duals of $\boldsymbol{G}$. Output a set of directions and orientations of the edges of $\boldsymbol{H}$, fathom the current B&B node and exit the algorithm.

Step 6:

Select a set $\boldsymbol{E} = \{\boldsymbol{e}_0, \boldsymbol{e}_1, \cdots, \boldsymbol{e}_n\}$ of unoriented edges of $\boldsymbol{H}$. Choose orientations $\boldsymbol{O} = \{\boldsymbol{o}_0, \boldsymbol{o}_1, \cdots, \boldsymbol{o}_n\}$ for the edges in $\boldsymbol{E}$. Create $3n+1$ B&B nodes, $3n$ of which are characterized by triplets $\boldsymbol{B}_{i,j} = (\boldsymbol{H}, \boldsymbol{E}_i, \boldsymbol{O}_j)$, $i = 1, \cdots, n$, $j = 1, 2, 3$, where $\boldsymbol{E}_i = \{\boldsymbol{e}_0, \boldsymbol{e}_1, \cdots, \boldsymbol{e}_i\}$, $\boldsymbol{O}_i = \{\boldsymbol{o}_0, \boldsymbol{o}_1, \cdots, (\boldsymbol{o}_i + j) \bmod 4\}$. The $(3n+1)$th node is characterized by the triplet $\boldsymbol{B}_0 = (\boldsymbol{H}, \boldsymbol{E}, \boldsymbol{O})$. Terminate the processing of the current B&B node.

V.6. Second Method of R-Dual Enumeration (Algorithm 3)

Step 1:

Assign appropriate orientations to the edges of the 4-completion that are incident to the outer vertices as in the definition of a well-formed O4-completion and produce a PO4-completion $\boldsymbol{H}_0$. Let $\boldsymbol{E}$ and $\boldsymbol{O}$ be empty. Enter $(\boldsymbol{H}_0, \boldsymbol{E}, \boldsymbol{O})$ on the list of B&B nodes.

Step 2:

Select any node $(\boldsymbol{H}^*, \boldsymbol{E}, \boldsymbol{O})$ from the list of live B&B nodes. If the list is empty, terminate the algorithm.

Step 3:

Check the tentative feasibility of $\boldsymbol{H}^*$. Assign orientations to those unoriented edges for which only a single orientation is possible if well formedness is to be achieved. Remove node $(\boldsymbol{H}^*, \boldsymbol{E}, \boldsymbol{O})$ from the list of live B&B nodes. If $\boldsymbol{H}^*$ is fully oriented and R-feasible, produce a set of edge orientations and go to step 2. If $\boldsymbol{H}^*$ is not R-feasible, go to step 2. Otherwise apply Algorithm 2 to generate descendants of the B&B node, enter them on the list of live vertices and go to Step 2.

V.7. Additional Considerations

It is asserted that Algorithm 3 applied to an unoriented 4-completion $\boldsymbol{H}$ of a graph $\boldsymbol{G}$ generates all possible R-

[6]Note that at this moment, the 4-completion $\boldsymbol{H}$ is still tentatively R-feasible. Ideally, a maximally oriented 4-completion is desired to represent the current set of R-dissections. However, finding a maximally oriented 4-completion remains a task of considerable difficulty by itself and has not been considered here.

duals of G that have corner faces corresponding to vertices $v_{bl}, v_{tl}, v_{br}, v_{tr}$ of H. To prove this assertion, it is necessary to show that the separation of each B&B node into descendants in Algorithm 3 has the following properties: (1) no set of orientations that does not correspond to any R-dual of G can occur in the algorithm output, and (2) that the set of orientations of the edges in an O4-completion describing D is eventually found by the algorithm for any R-dual D of G.

Condition (1) follows from steps 4 and 5 of Algorithm 2 which guarantee that only well-formed 4-completions are generated; Theorem 5 establishes the existence of an R-dissection corresponding to each such 4-completion.

For condition (2), denote by $D(B)$ a set of R-dissections represented by a PO4-completion H after assigning orientations O to edges E in the triplet $B=(H,E,O)$. $D(B_0)$ is then the set of duals represented by H after assigning orientations $o_0, o_1, \cdots, o_n$ to edges $e_0, e_1, \cdots, e_n$. Similarly, $D(B_{n,j})$ is the set of duals represented by H after assigning orientations $o_0, o_1, \cdots, o_{n-1}$ to edges $e_0, e_1, \cdots, e_{n-1}$ and orientation $(o_n + j) \bmod 4$ to edge e_n; together, $D(B_0) \cup D(B_{n,1}) \cup D(B_{n,2}) \cup D(B_{n,3})$ are all duals represented by H after assigning orientations $o_0, o_1, \cdots, o_{n-1}$ to edges $e_0, e_1, \cdots, e_{n-1}$. Generally, $D(B_{k,j})$ is a set of all duals where rectangles corresponding to the endpoints of edge e_i, $i=1,\cdots,k-1$, remain in relative positions described by the orientation corresponding to o_i; rectangles corresponding to the endpoints of edge e_k remain in relative positions described by the orientation corresponding to

$$(o_k + j) \bmod 4; \quad \bigcup_{\substack{i=k,\cdots,n \\ j=1,2,3}} D(B_{k+1,j}) \cup D(B_0)$$

is the set of all duals where rectangles corresponding to the endpoints of edge e_i, $i=1,\cdots,k$, remain in relative positions described by the orientation corresponding to o_i. Thus

$$\bigcup_{\substack{i=1,\cdots,n \\ j=1,2,3}} D(B_{i,j}) \cup D(B_0)$$

after Step 7 of Algorithm 2 are all R-dissections represented by H before step 7. Hence the partitioning preverves the representation of all R-duals and all of them are eventually found by the algorithm.

The complexity of the algorithm is not easy to determine in a general case, as the time spent on searching the B&B tree branches that describe non-R-feasible PO4-completions would have to be considered. If this time is disregarded, then the remaining activities in the algorithm consume time proportional to the number of edges in the 4-completion.

VI. Conclusions

This paper presents a graph-theoretical characterization of rectangular dissections. Theorems are given for necessary and sufficient conditions for a graph to have a rectangular dual. A representation of rectangular dissections in the form of a directed graph called an oriented 4-completion is defined. The perimeter constraints which frequently occur in the applications of rectangular dualization, can be expressed in the form of polar subgraphs of a 4-completion. Possibilities of changing the morphology of a rectangular dissection by applying local transformations have been analyzed; this led to the development of a fast algorithm for enumerating rectangular duals that is based on a limited set of possible local changes in the structure of a rectangular dissection. A second procedure for systematically generating all rectangular duals of a graph was developed from the notion of partitioning the sets of rectangular dissections into disjoint subsets by sequentially assigning orientations to the unoriented edges in a partially oriented 4-completion. A partially oriented 4-completion is found to be a complete representation of a set of rectangular dissections that have fixed geometrical relations that correspond to oriented edges. In contrast, unoriented edges express only the adjacency of component rectangles without specifying the orientation of common boundaries. In particular, an unoriented 4-completion represents all possible rectangular dissections having identical adjacency relations among the component rectangles. The scheme for enumerating rectangular duals based on a sequential orientation assignment of the edges of a 4-completion can be adapted for a perimeter-constrained branch-and-bound area optimization of floorplans [18].

Appendix I
Proof of Theorems 7 and 8

General Remarks about the Proof of Theorem 7

Because of rotational symmetry, it suffices to prove the theorem for the leftmost R-dual. Contrary to the theorem, suppose that there are two different R-dissections D_1 and D_2 with no LT-structures and identical unoriented 4-completions. The reasoning in the proof is to take the smallest possible contradictory dissections and show that the difference cannot occur in the vicinity of the upper left corner of D_1 or D_2; this argument can then be extended to show that the difference cannot occur in the vicinity of the left side of the bounding rectangles. Finally, it is shown that two contradictory dissections can be constructed that are smaller than D_1 and D_2, thereby invalidating the contrary assumption.

Proof of Theorem 7

Take contradictory dissections D_1 and D_2 with the least possible number of component rectangles. Both D_1 and D_2 are R-duals of some graph G.

Consider the upper left corner rectangle v_{tl} in D_1. This rectangle cannot contain two corners, as the corresponding rectangle in D_2 would also contain two corners, under which circumstances, deleting v_{tl} from both D_1 and D_2 would produce two smaller contradictory dissections D_1^* and D_2^*. These would be different leftmost R-duals of the graph obtained from G by deleting v_{tl}.

Since v_{tl} contains only one corner of the bounding rectangle, there are two possibilities for the bottom edge of

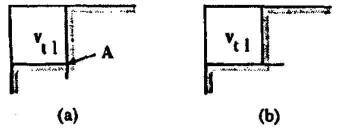


Fig. 20. Configurations of the left upper corner of a bounding rectangle.

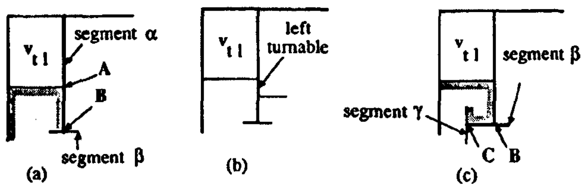


Fig. 21. Extended configuration of the left upper corner of the bounding rectangle.

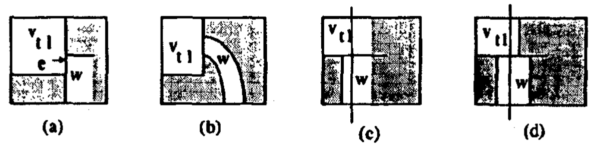


Fig. 22. Possible spatial relations between w and v_{tl}.

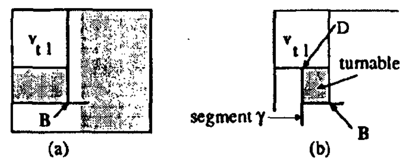


Fig. 23. Possible configurations of the neighborhood of the left side of D_1.

rectangle v_{tl}: either it is a maximal horizontal segment, Fig. 20(a), or it is not a maximal horizontal segment, Fig. 20(b).

Suppose first that the condition depicted in Fig. 20(a) occurs in D_1 and consider the maximal vertical segment α containing the right boundary of v_{tl}. Segment α must end somewhere below the lower right corner A of v_{tl}, in the inverted-T configuration marked as point B in Fig. 21(a). Simultaneously, below point A, this segment must have only one rectangle w to its right, for otherwise an LT-structure would exist in D_1, as shown in Fig. 21(b).

If the horizontal bar of the inverted-T, marked as segment β, is the lower edge of the bounding rectangle of D_1, then D_1 appears as in Fig. 22(a), that is, segment α containing edge e separating w from v_{tl} runs the entire height of D_1. Cutting D_1 along α produces two R-dissections with no LT-structures, D_{1L} to the left of α and D_{1R} to the right. Consider the appearance of rectangle w in the leftmost R-dual D_2. There are three possibilities: (1) w is adjacent to v_{tl} along a vertical line and touches the right lower corner of v_{tl}, (2) w is adjacent to v_{tl} along a vertical line and does not touch the right lower corner of v_{tl}, and (3) w is adjacent to v_{tl} along a horizontal line. In case (1), D_2 has an appearance similar to that in Fig. 22(a) and it could also be split into D_{2L} and D_{2R}; one of the two pairs (D_{1L}, D_{2L}) or (D_{1R}, D_{2R}) would constitute a smaller pair of contradictory dissections than the assumed smallest pair D_1, D_2. Case 2 is impossible, as w is also adjacent to the lower side of the bounding rectangle; as depicted in Fig. 22(b), w could not be a rectangle. In case 3, D_2 appears as shown in Fig. 22(c). Apply a right turn to the edge e in D_1; the result, $T_e(D_1) = D_3$, must resemble Fig. 22(d). Cutting D_2 and D_3 with a vertical line passing through w and v_{tl} would produce two pairs of R-dissections (D_{1L}, D_{2L}), (D_{1R}, D_{2R}) without LT-structures, as edge e which is left-turnable in D_3, becomes fixed to the cut-line in D_{1L} and D_{2L}; one of these pairs would be smaller than the assumed smallest contradictory pair.

Returning to Fig. 21, the above argument shows that the maximal segment β belongs to the interior of D_1; the left end of this segment, therefore, creates a configuration resembling a left turned-T as in Fig. 21(c) at point C. The horizontal edge CB can have only one adjacent rectangle below it, otherwise a left-turnable single edge would exist; this condition resembles the one depicted in Fig. 21(b) but rotated clockwise. Now consider segment γ shown in Fig. 21(c). If it is the left side of the bounding rectangle, then there would exist an R-dissection D_1^* that is a subgraph of D_1 with its lower right corner at the point B and its upper left corner overlapping the left upper corner of D_1. D_1^* is composed of rectangle v_{tl} and the shaded R-dissection below v_{tl}, see Fig. 23(a). In D_2, the corresponding subgraph D_2^* also has to be an R-dissection, as it is dual to the interior of a cycle of length 4 in the 4-completion of G. Therefore either D_1^* and D_2^* is a pair of smaller contradictory R-dissections or they can be substituted in both D_1 and D_2 by single rectangles, again creating a pair of smaller contradictory dissections. Thus segment γ is not the left side of the bounding rectangle and has to be in the interior of D_1. Consider the upper end of this segment; it ends in a straight-T configuration and has a single rectangle to its left for analogous reasons as those depicted in Fig. 21(b). If segment γ ends in a point D on the lower side of v_{tl}, then the R-dissection contained between points B and D would be left turnable as shown in Fig. 23(b).

By now a clear pattern has emerged; it is possible to trace a spiral path taking a left turn at each T-configuration until the path closes on itself creating an LT-structure resembling Fig. 23(b). Since all possibilities implied in Fig. 20(a) would invalidate the contradictory assumption, then in both D_1 and D_2 the right side of the upper left rectangle must be a maximal vertical segment as shown in Fig. 20(b). Also, there must be a single rectangle adjacent to v_{tl} from below as otherwise a left-turnable segment would exist. For analogous reasons, the part of D_2 adjacent to the left side of its bounding rectangle appears similarly.

It follows, then, that the sequence of all rectangles adjacent to the left side of D_1 with the exception of v_{tl}, followed by the sequence of all rectangles to the right of v_{tl} correspond to a vertical cutting path in the O4-completion describing D_1; the same statement is true for D_2. By

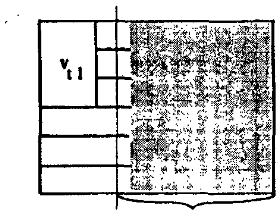


Fig. 24. Construction for obtaining smaller contradictory dissections.

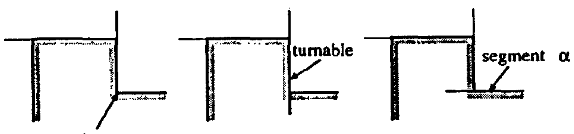


Fig. 25. Non-simple T-structure.

making rectangle v_{t1} sufficiently narrow, D_1 and D_2 can be given such dimensions that this sequence of rectangles can be cut with the single vertical line shown in Fig. 24. The part D_{1R} of D_1 to the right of this cutting line contains all rectangles except v_{t1} and no LT-structures are introduced by the cut. The same statement is true for D_{2R} that can be cut from D_2. Thus D_{1R} and D_{2R} would be a pair of smaller contradictory dissections than the assumed smallest pair.

In summary, as the assumption that the set of contradictory graphs has a smallest element cannot be supported, this set must be empty and the theorem is proved. ■

Proof of Theorem 8:

Given D_0, perform the transformations implied by simple LT-structures in D until there are no more simple LT-structures. The R-dissection D^* at this stage is either identical to D_0, or D^* contains an LT-structure that is not simple. If D^* is identical to D_0, the sequence of the T-structures taken in the reverse order is the sequence of T-structures to be turned right to obtain D from the leftmost D_0. It remains to be shown that the second case is impossible. Suppose that there is an R-dissection D^* containing no simple LT-structure but containing a non-simple LT-structure C. C must have at least 5 vertices in common with the rest of D^* and the boundary of C is a rectilinear polygon with at least 6 vertices. A clockwise walk on this boundary must encounter a sequence of two 90 degree right angles followed by a 90-deg left angle. This is depicted in Fig. 25, where the interior of D^* is shaded. Note that the orientations of the external adjacent edges in D^* must be as shown in Fig. 25 to enable left-turnability. Consider the vicinity of vertex A. The maximal vertical segment must end there, otherwise a simple, single-edge LT-structure would exist as shown in Fig. 25(b). Therefore, at A an inverted T-configuration must occur, as in Fig. 25(c). Following the left part of segment α and proceeding as in the proof of Theorem 7, the existence of a simple T-structure in D^* follows. Therefore, the assumption that all possible simple T-structures in D^* were right-turnable is false. ■

Appendix II
Algorithm for Finding a Rectangular Dual [5]

Introduction

The first step of the algorithm consists of numerical checks on the graph G which are done in linear time. Failure of G to comply with these checks shows that no 4-completion exists for G and causes the algorithm to terminate. Step 2 constructs a graph H which is conjectured to be a 4-completion and determines the corner faces of an R-dual, should this R-dual exist. The only possible violation of the definition of a 4-completion in H occurs if G itself contains a triangle that is not a face. Step 3 labels the vertices of G in order to facilitate determining offending triangles in G in the recursive calls of the Algorithm A-2. After Step 3, the orientations of the edges in the dual graph are the only missing information that is necessary to create an R-embedding; generating an incidence matrix of a dual graph from H is straightforward. The orientation assignment is described as a separate Algorithm A-2 as it can be best explained in a recursive manner. Algorithm A-2 terminates unsuccessfully upon encountering a triangle that is not a face. Algorithm A-2 recursively divides the initial 4-completion into smaller 4-completions until all edges of the R-dual of G are assigned vertical or horizontal directions or an offending triangle is identified, as shown in Fig. 4. This recursive application of a divide and conquer paradigm can be interpreted as cutting the (as yet unknown) R-embedding into parts having a smaller number of faces; these parts are then merged along the cutting lines.

The data for the algorithm are assumed to represent a plane embedding of a planar connected graph G. Thus in addition to the information about the structure of G, given, say, in the form of a list of the edges with their endpoints, some information is needed to allow the drawing of the plane embedding of G. Sufficient information to draw a planar graph without intersecting edges, for example, is a list of all faces of G, where each face is given as an ordered set of edges $f_i = \{e_1, e_2, \cdots, e_n\}$ creating the boundary of the face, and the external face is indicated.

Algorithm A-1

Step 1:

For all internal vertices v_i of G, check if the degree $d(v_i) \geq 4$. If for some $i, d(v_i) < 4$, go to Step 5. For all internal faces f_i of G, check if the degree $d(f_i) = 3$. If for some $i, d(f_i) \neq 3$, go to Step 5. Find all maximal blocks b_i of G using a depth first search [9]. If some b_i does not contain any edge of the external face, go to Step 5. Obtain the block neighborhood graph of G and if it is not a path, or if it contains a cycle or a vertex of degree 3, go to Step 5. Identify shortcuts and critical CI-paths of each block b_i in G. Proceed only if (1) there are at most 4 critical CI-paths and (2) if G is not a block, the critical CI-paths occur in the end-blocks of G, at most 2 per end-block; otherwise go to Step 5.

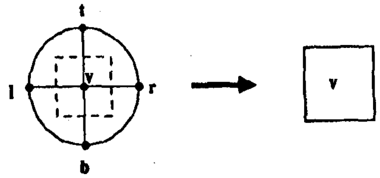


Fig. 26. When graph G consists of a single vertex, its trivial 4-completion yields an R-dual of G that is a single rectangle.

Step 2:

Select a vertex from the interior of each of the critical corner implying paths. If the number of the critical CI-paths is $n < 4$, select additional $4\text{-}n$ vertices on the outer face of G. Label the chosen corner vertices v_{bl}, v_{tl}, v_{tr} and v_{br}. Construct a 4-completion H as in Fig. 4(a)–(c).

Step 3:

Assign a label $L(v)$ to each vertex v of the 4-completion H. Initially, each label is the name of the respective vertex, $L(v) = v$.

Step 4:

Use Algorithm A-2 to determine horizontal and vertical edges of the R-dual of G. If Algorithm A-2 terminates unsuccessfully, go to Step 5. Otherwise, output the orientations of the dual edges so that based on this information the R-dual of G can be drawn. Terminate the algorithm successfully.

Step 5:

Terminate the algorithm unsuccessfully as no 4-completion and no R-dual exists for graph G.

Algorithm A-2

Step 1:

If there is an edge tb in H, the original graph G contains two adjacent triangles $L(t)L(b)L(l)$ and $L(t)L(b)L(r)$, at least one of which is not a face; go to Step 6. If there is an edge rl in H, the original graph G contains two adjacent triangles $L(l)L(r)L(t)$ and $L(l)L(r)L(b)$, at least one of which is not a face; go to Step 6.

Step 2:

If H has more than 5 vertices, go to Step 3. Otherwise H must be a trivial 4-completion consisting of 5 vertices and 8 edges as shown in Fig. 26. Edges dual to $L(t)L(v)$ and $L(b)L(v)$ are made horizontal, edges dual to $L(l)L(v)$ and $L(r)L(v)$ are made vertical. Terminate the algorithm successfully.

Step 3:

If all outer vertices of H have degree at least 3, go to Step 4. Otherwise, some outer vertex has degree 2 and the remaining outer vertices create a triangle with a non-empty interior; go to Step 6. Fig. 27 illustrates this case for r having degree 2; in a situation such as shown there, $L(t)L(b)L(l)$ would be identified as the offending triangle in the original graph G.

Step 4:

If all outer vertices of H have degree at least 4, go to Step 5. Otherwise, some outer vertex of H has degree 3. Select one of these vertices; assuming that vertex t has been selected, H would appear as in Fig. 28; edges dual to

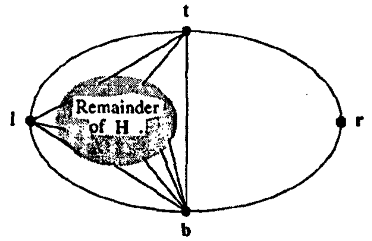


Fig. 27. A 4-completion with an outer vertex having degree 2.

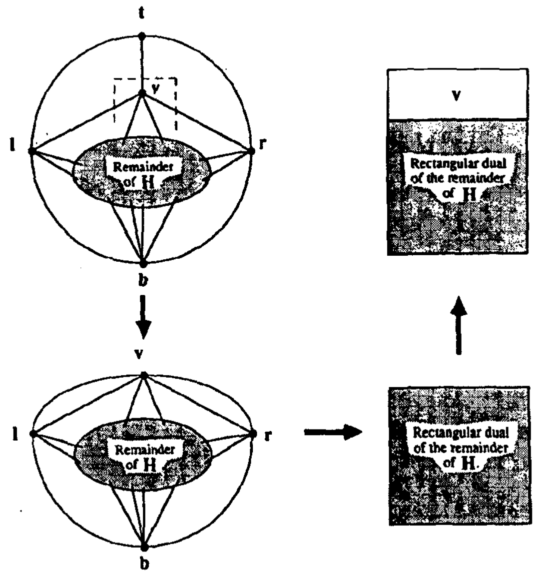


Fig. 28. A 4-completion with some outer vertex having degree 3 and an interpretation of the resulting direction assignment.

$L(l)L(v)$ and $L(v)L(r)$ would be made vertical and the edge dual to $L(v)L(t)$ would be made horizontal. Delete vertex t together with the corresponding edges and obtain graph H'. Apply Algorithm A-2 recursively to H'. If this recursive application terminates unsuccessfully, go to Step 6, otherwise terminate the current application successfully. The former case may occur, for example, if some outer vertex, say r, acquires degree 2 in H'. Otherwise, if the outer face of H has exactly 4 vertices of degree at least 3, and if H is a non-trivial 4-completion, then H' is also a 4-completion, for all properties (b)–(d) from the definition of a 4-completion are automatically preserved in H', if H is not trivial.

Step 5:

All outer vertices of H have degree 4 or more. H must appear as in Fig. 29, with a vertex v_{bl} adjacent to l and b and a vertex v_{tr} adjacent to t and r. If edge $v_{bl}v_{tr}$ exists, take path $P = tv_{bl}v_{tr}b$. Otherwise, let P be any valid cutting path between t and b, not passing through r, l, v_{bl} nor v_{tr}. It can be, for example, a shortest path between t and b in a graph H' obtained by deleting r, l, v_{bl} and v_{tr} from H.[7]

[7] H' is connected if edge $v_{bl}q_{tr}$ does not exist [6].

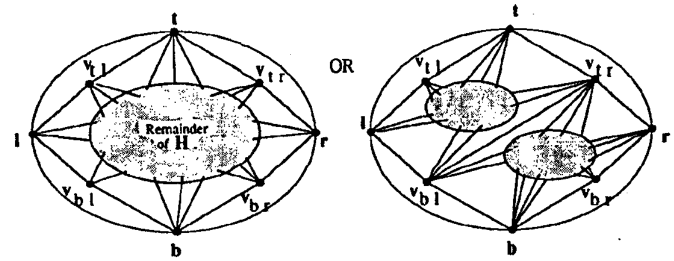


Fig. 29. Two possibilities for a 4-completion with all outer vertices having degree 4 or more.

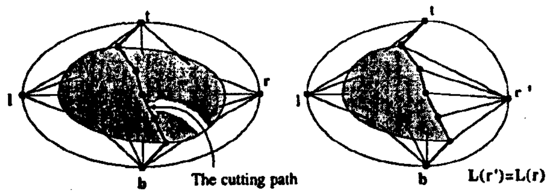


Fig. 30. Construction of H'.

P is a valid cutting path separating either v_{bl} and v_{tr} or v_{br} and v_{tl}, thereby insuring that the two graphs H' and H'' that are subsequently constructed from H have fewer vertices than H. Assign a horizontal orientation to all edges dual to the edges of the cutting path. Construct H' and H'' by splitting H along the cutting path and adding vertices l', r' with edges to all the vertices of the cutting path, as shown in Fig. 30 for H'.

Assign labels $L(r)$ and $L(l)$ to r' and l', respectively. Every other vertex of H' and H'' is assigned the label of the corresponding vertex in H. H' and H'' are 4-completions, as the construction preserves all properties of a 4-completion. Apply Algorithm A-2 to H' and H''. If either recursive application terminates unsuccessfully, go to Step 6. Otherwise terminate the current application of the algorithm successfully.

Step 6:

H is not a 4-completion and graph G has no R-dual. Terminate the algorithm unsuccessfully.

References

[1] W. R. Heller, G. Sorkin, and K. Maling, "The planar package planner for system designers," in *Proc. 19th Design Autom. Conf.*, Las Vegas, NV, pp. 253–260, June 1982.

[2] K. Maling, S. H. Mueller, and W. R. Heller, "On finding most optimal rectangular package plans," in *Proc. 19th Design Autom. Conf.*, Las Vegas, NV, pp. 663–670, June 1982.

[3] D. S. Buchanan and G. J. Brebner, "On compiling structural descriptions to floorplans," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, Santa Clara, CA, pp. 6–7, Sept. 1983.

[4] R. Schmid and U. G. Baitinger, "The role of floorplan tools in the VLSI design process," Univ. Karlsruhe, Inst. Tech. der Informationsverarbeitung, West Germany, Tech. Rep. 1984.

[5] K. Kozminski and E. Kinnen, "An algorithm for finding a rectangular dual of a planar graph for use in area planning for VLSI integrated circuits," in *Proc. 21st Design Autom. Conf.*, Albuquerque, NM, pp. 655–656, June 1984.

[6] ——, "Rectangular duals of planar graphs," *Networks*, vol. 15, pp. 145–157, Jan. 1985.

[7] W. R. Heller, "Package planning in VLSI," in *Proc. 1985 Int. Symp. Circuits Syst.*, Kyoto, Japan, pp. 1–9, June 1985.

[8] J. Bhasker and S. Sahni, "A linear algorithm to find a rectangular dual of a planar triangulated graph," in *Proc. 23rd Design Autom. Conf.*, Miami, FL, pp. 108–114, June 29–July 2, 1986.

[9] S. V. Aho, J. E. Hopcroft, and J. D. Ullman, *Data Structures and Algorithms*. Reading, MA: Addison-Wesley, 1983.

[10] S. Tsukiyama, K. Tani, and I. Shirakawa, "Enumeration of rectangular duals of a maximal planar graph," to be published.

[11] J. Grason, "A dual linear graph representation for space-filling location problems of the floor plan type," in *Emerging Methods in Environmental Design and Planning, Proc. of The Design Methods Group, 1st Int. Conf.*, G. T. Moore, Ed., Cambridge, MA, 1968, pp. 170–178.

[12] ——, "Methods for the computer-implemented solution of a class of 'floor plan' design problems," Ph.D. dissertation, Carnegie-Mellon Univ., Pittsburgh, PA, 1970.

[13] ——, "An approach to computerized space planning using graph theory," in *Proc. 8th Design Autom. Workshop*, Atlantic City, NJ, pp. 170–179, June 1971.

[14] P. Steadman, "Graph-theoretic representation of architectural arrangement," in *The Architecture of Form*, L. March (Ed.), London, England: Cambridge Univ. Press, 1976, pp. 94–115.

[15] U. Flemming, "Wall representations of rectangular dissections and their use in automated space allocation," *Environment and Planning B*, vol. 5, pp. 215–232, Dec. 1978.

[16] P. Galle, "An algorithm for exhaustive generation of building floor plans," *Commun. ACM*, vol. 24, pp. 813–825, Dec. 1981.

[17] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*. New York: North Holland, 1981.

[18] K. Kozminski, "Rectangular dualization for computer-aided design of VLSI integrated circuit floorplans," Ph.D. dissertation, Univ. of Rochester, Rochester, NY, 1985.

[19] R. L. Brooks, C. A. B. Smith, A. H. Stone, and W. T. Tutte, "The dissection of rectangles into squares," *Duke Math. J.*, vol. 7, pp. 312–340, 1940.

Krzysztof A. Kozminski (S'82–M'85) received the M.S. degree (honors) from Warsaw University of Technology in 1978 and the Ph.D. degree from the University of Rochester in 1985.

Since 1978 he was on the faculty of Warsaw University of Technology, where now he is an Assistant Professor in the Institute of Microelectronics and Optoelectronics. During 1985–1986 he was a Visiting Assistant Professor at the Department of Computer Science, Duke University. His research interests are in the computer-aided design of integrated circuits, with particular emphasis on layout and synthesis of digital circuits.

Edwin Kinnen (S'50–M'51–SM'59) received the B.S. degree from the University of Buffalo, the M.S. degree from Yale University, New Haven, CT, and in 1958 the Ph.D. degree from Purdue University, West Lafayette, IN.

He has been a member of the staff of the Westinghouse Research Laboratory, and also of Purdue University and the University of Minnesota. He has been with the University of Rochester since 1963, and is Professor of Electrical Engineering.