Artificial Intelligence and Pattern Recognition in Computer Aided Design, Latombe, ed.

ARTIFICIAL INTELLIGENCE TECHNIQUES IN AUTOMATIC LAYOUT DESIGN

Luís Moniz Pereira
Divisão de Informática
Laboratório Nacional de Engenharia Civil
Av. do Brasil, Lisboa 5 - Portugal

ABSTRACT. This work is abridged from a thesis (2). In abstract terms, the class of problems addressed results from the following "desideratum": Given a list of imposed and of non-permissible adjacencies among a set of planar rectangular spaces, and given dimensional intervals constraining each space and each adjacency, to search and generate all possible layouts on the plane of such a set of spaces which satisfy the requisites, and to provide for a computer implementation of the algorithms developed, as well as for user/machine interaction in view to the alteration of the dimensional intervals imposed, especially in case incompatibilities arise. These layouts must also respect restraints upon their contour. The artificial intelligence concepts utilized to solve this class of problems pertain to: problem representation and definition; problem decomposition into modular subproblems; testing problem data well-formedness; identification of a typology of the constraints in the search-space in order to achieve an efficient exhaustive non-redundant backtracking search for solutions; generate-and-test mechanisms; interaction between local and global constraints; heuristic search.

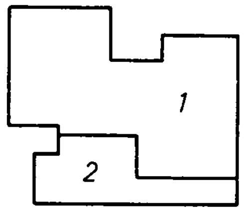
Computer programs were in fact written and tested which embody the theory and problem-solving strategies devised for this class of problems.

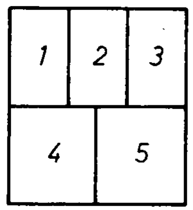
1 - PROBLEM FORMULATION

In this section we state the initial problem formulation, and justify the way we divide it into subproblems within a given representation. In addition, we delineate the methods used for resolution of these subproblems.

Hypothesis 1: By a space we mean the area inside a closed polygon on the plane, convex or not, whose consecutive sides form angles of 90 or 270 degrees between themselves.

By layout scheme we mean any partition of a certain space, whose polygon defines the contour of the layout scheme, into a finite number of other spaces. Examples of layout schemes:





The spaces of the partition are identified by the first positive integers, and any two different spaces whose polygons or boundaries have a common side are said to be adjacent. Each adjacency is expressed by an unordered pair of the appropriate numbers.

Hypothesis 2: A list is given of pairs of positive integers, and each pair (a, b) is interpreted as meaning that a and b are spaces and that space a is adjacent to space b. The following is a complete list L of the adjacencies occuring in one of the above layout schemes: L = ((1, 2), (1, 4), (2, 3), (2, 4), (2, 5), (3, 5), (4, 5)).

The given list cleary defines a graph, whose nodes and edges are spaces and adjacencies, respectively.

Hypothesis 3: It is understood that all spaces figuring in any given list are rectangular, i.e. have a four-sided boundary.

Hypothesis 4: For each rectangular space, two tolerance intervals are specified on the positive integers, i.e. for each of its two dimensions an ordered pair consisting of a minimum and a maximum value is stipulated by means of two positive integers.

Furthermore, an excess integer is uniformly assigned to each tolerance interval, indication by how much its maximum or minimum values may be exceeded, in case no solution can be found within the interval.

Hypothesis 5: For each adjacency pair (a, b), a tolerance interval is specified on the non-negative integers, which regulates the extent of contact between spaces a and b, along the adjacency expressed by (a, b).
When no particular tolerance interval for an adjacency pair is required, the unlimited interval on the non-negative integers is taken.

Hypothesis 6: A class of forms is specified for the contour, by giving a description of the class of sequences of convex and concave angles allowed for the contour polygon.

If no such description is given, a choice has to be made whether the class of forms allowed for the contour is comprised of all possible polygons, or if it is to be restricted to rectangular polygons alone.

Hypothesis 7: Some of the spaces, eventually all or none, may be marked as necessarily exterior spaces, i.e. as spaces having at least a point of their boundary in common with the contour.

Initial Problem: Given these hypothesis, obtain all possible layout schemes in agreement with them, and with the particular data they convey.

2 - PROBLEM REPRESENTATION

As we will see next, a graph representation will provide us with a framework for formulating and solving the envisaged class of problems. Also, it lends itself to automatic problem solving.

To start with, notice how the hypotheses are easily expressed in terms of graphs. Indeed, the graphs determined by means of H2, are those with no isolated nodes (i.e. nodes without any edges), and no loops. From H3 it follows, however, that such a graph must also be simple (i.e. no two edges sharing the same nodes) since any two disjont rectangles can only be adjacent along one common side at the most and, according to H1, they are disjont. Hypotheses 4 and 5 can be dealt with by considering flows of dimensions within the graph. The way hypotheses one, three, six and seven are taken into account in a graph representation space will now be mentioned. Consider the following figures.

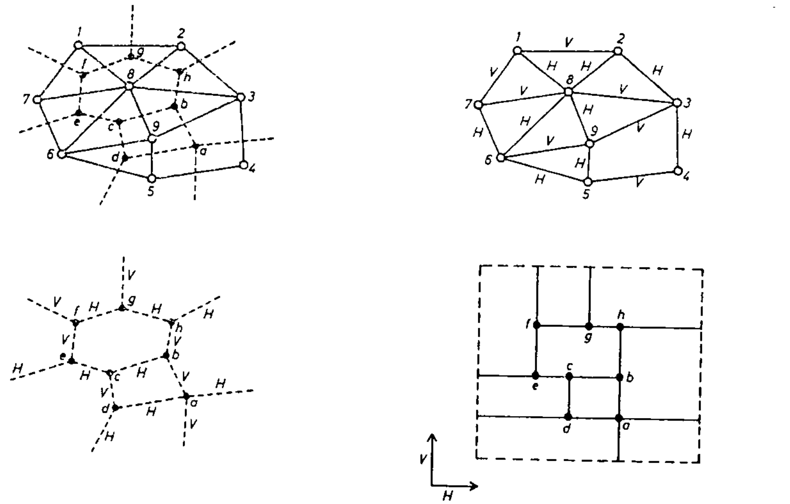
In the first one, a planar realization of a graph is shown (with numbered nodes and full edges), superimposed by its dual (with lettered nodes and dashed edges). This is an example of the graph refered in H2.

In the second one, an assignment of H's and V's has been made to the edges of the numbered graph; call it an HV - assignment.

Since there is a one-to-one correspondence between the edges of the graph and its dual, there is a resulting assignment of H's and V's to the edges of the dual, which is shown in the third figure.

We now define, relative to the horizontal and vertical axes of the sheet of paper, properties "horizontal" and "vertical" and intepret "H" and "V" as meaning that an edge must be drawn horizontally or vertically, respectively.

The dual can now be redrawn, attending to the assignement of H's and V's to its edges, in the unique way shown in the fourth figure, where a dashed rectangular contour has been added: this contour results from considering all spaces rectangular. Note that the particular dimensions utilized for drawing the figure are, at the moment, irrelevant; what is essential is the orientation of the edges, and the fact that they were redrawn attending only to their assignment value and the general condition that they could not have any common points except the extremities; this condition follows from the fact that the dual of a planar representation of a graph is itself planar.



We have this way obtained a layout scheme of rectangular spaces with a rectangular contour. Of course, not all HV - assignments in a given graph determine a layout scheme of rectangular spaces. There are also graphs for which there is no HV - assignment giving a layout scheme of rectangular spaces or R - layout-scheme.

2 - PROBLEM DECOMPOSITION

Problem 1: Given a graph G, under what necessary and sufficient conditions does there exist at least one HV - assignment leading to a layout scheme of rectangular spaces; i.e. a permissible HV - assignment.

Problem 2: When does a permissible HV - assignment, or PHV - assignment, lead to a R - layout-scheme with a specified type of contour, eventually rectangular.

Now, these two problems immediately suggest several others.

One obvious necessary condition of Problem 1 is for graph G to have at least a planar realization, and hence to be planar. It may have however various planar realization; for each such realization R, the question then arises if there exists at least one PHV - assignment giving rise to a R-layout-scheme by means of the dual of R.

These comments lead us to decompose Problem 1 into the following five subproblems.

Subproblem 1.1: Given graph G, find out whether or not it is planar.

Subproblem 1.2: If G is planar, find all its planar realizations.

Subproblem 1.3: If G is not planar, how to alter G to make it planar. The different ways of transforming G, result into corresponding subproblems: such transformations however, to be efficient, should be closely matched to the conditions of Subproblem 1.4. Similary any problem or subproblem arising from the necessity to transform initial data so as to comply with given conditions, should be solved by considering only those transformations which also comply to subsequent problem or subproblem conditions.

Subproblem 1.4: For any planar realization of G, under what conditions does it allow at least one PHV - assignment.

Subproblem 1.5: For each realization of G allowing at least one PHV - assignment, find all its PHV - assignments.

Problem 2 is actually made up of two subproblems.

Subproblem 2.1: How to specify, using a graph theoretical representation, the types of contour forms to be allowed for the R-layout-schemes derived from the dual of a planar realization of a graph.

Subproblem 2.2: Once a given type of contour form is specified, in graph theoretical terms, for a given planar realization of a graph, derive only those PHV - assignments leading to R-layout-schemes with that type of contour form.

We will relegate the statement of problems and subproblems pertaining to questions of dimension to another section, after having solved some of the merely topological problems stated up to this point. An important property indeed of the method utilized for solving the Initial Problem is that the merely topological questions can be and have been separated from the metrical ones. This is accomplished by having on overall generative mechanism producing topological solutions, which function as input candidates to the dimension giving mechanisms for producing metrical layout schemes.

Separateness or modularity is in fact a characteristic which manifests itself throughout the overall solution method. It occurs, for example, when Problems one and two are decomposed into the sequence of steps:

a) finding if a graph is planar;
b) discovering one of its planar realization;
c) extracting from it all its other planar realizations;
d) finding for each such realization those allowing at least one PHV-assignment;
e) obtaining all PHV-assignments of any realization having at least one;
f) restricting the PHV-assignments to only those which conform to a specified contour type.

Other situations where separateness also occurs are those of:

g) dividing any given PHV-assignment of a given representation into the unique partial assignment present in all PHV-assignments for that representation (its nucleus) and the particular completions of that partial assignment which generate all its possible PHV-assignments.
h) considering that the absolute orientation of each is composed of its orientation relative to the layout scheme plus the absolute orientation of the whole layout. It will be shown that in a PHV-assignment the relative orientation of every edge is simply determined by the "H" or "V" assigned to it. Thus, the edges of a graph need not be oriented; only the layout as a whole will have to be oriented, relative to an external referential, for the purpose of being drawn.
i) since the spaces considered are rectangular, the dimensional flow of each dimension through a given R-layout-scheme is independent of the other, and can be treated separately by considering an appropriate subgraph of the original one, which expresses only those adjacencies pertinent to the dimensional flow in question. Furthermore, when no tolerance intervals are supplied (i.e. no dimensions are stipulated whatsoever), one can always draw a R-layout-scheme given its PHV-assignments, by finding, for example, the minimum dimensions each space must have in terms of a unit module (i.e.any dimensions will became a multiple of the module). Thus, a complete separeteness is possible between any PHV-assignment defining the relative positions of the spaces of a R-layout-scheme and the desired dimensions for those spaces conveyed fy given tolerance intervals.

The import of separateness resides in the fact that diverse heuristics can be used for each separate subproblem, and more easily implemented. Moreover, interaction facilities become better located and defined.

Separateness also permits program modularity and easiness of reformulation, because the design of each component can be carried out with some degree of independence of the design of others, since each will affect the others largely through its function, and independently of the details of the mechanisms that accomplish its function. One way of considering the decomposition, but aknowledging that the inter relations among the components cannot be ignored, is to think of the design process as involving first the generation of alternatives and then the testing of these alteratives against a whole array of requirements and constraints. There need not be merely a single generate-test cycle, but there can exist a whole interconnected series of such cycles. The generators implicitly define the decomposition of the design problem, and the tests guarantee that undesirable consequences will be noticed and pruned. Alternative decompositions correspond to different ways of dividing the responsabilities for the final design between generators and tests.

3 - LAYOUT SCHEMES FROM GRAPHS

To solve Problem 1 completely we need to solve its five Subproblems. Subproproblems 1.1 and 1.2 are treated in (2). Subproblem 1.3 will not concern us because we presume that alteration of a graph to make it planar will always be carried out by the user (although the planarity algorithm in (2) gives relevant information to that end). The solution given to Subproblem 1.5 only becomes comprehensible after we state the conditions refered to in Subproblem 1.4, and show that they constitute indeed its solution. However, the actual details of the solution contrived for Subproblem 1.5 will have to be found in (2), since they are rather involved and unessential to this exposition.

We will now work out the conditions mentioned in Subproblem 1.4, dividing them into three groups - conditions A, B and C. Group A expresses those conditions which are "rigid", i.e. that either are or are not obeyed by a graph, independently of any HV - assignment. Group B refers to conditions stipulating which HV - assignments are permissible, if any, once it is known that conditions A are met. For each realization of a graph, since there may be several HV - assignments obeying conditions B, finding all of them gives rise to the family of the R-layout-schemes obtained. from each given planar realization. Thus, the set of all planar realizations produces the set of all such families.

Group C refers to conditions on the overall contour. Conditions A and B will first be presented, and then justified. Conditions C follow.

Conditions A

A0 - Graph G must be finite, simple (i.e. at most one edge between any two nodes), connected, (i.e. at least one path between any two nodes), non-separable (i.e at least one cycle containing any two nodes), and planar, admitting at least one planar realization R which obeys all subsequent A and B conditions.

A1 - Every interior node of G in R must have four or more edges.

A2 - Every face of G in R must have either three or four edges; i.e. each of them is either "triangular" or "quadrangular".

Conditions B

B0 - Each edge of G in R must be assigned either an 'H' or a 'V', such that all other subsequent B conditions are met. We will often speak of 'H' and 'V' as two different colours, 'E' and 'V' respectively, and speak of an HV - assignment as a colouring of the graph.

B1 - No triangular face of G in R can have its three edges assigned the same colour.

B2 - All quadrangular faces of G in R must have opposite edges assigned the same colour, and non-opposite edges with different colours.

B3 - The edges of any interior node of G in R must be coloured in such a way that they may be grouped into four sucessive colour groups around the node, opposite groups having the same colour, and non-opposite groups having different colours.

Justification of conditions A:

A0 - Since the layout-schemes are to have rectangular spaces the graph must be simple, because no two disjoint rectangular spaces can be adjacent on more than one side. Note, however, that the process of obtaining the dual would work as well with non-simple graphs, giving then rise in a natural way to non-rectangular spaces. We assume the graph is connected since if it were not each of its connected components could be separately considered without loss of generality. (For a similar reason we also assume that the graph is non-separable, since each of its separate blocks could be considered separately, given a planar realization of the graph.) The graph must be planar if one of its realizations is to have a dual. Furthermore, only its planar realizations can eventually determine R-layout-schemes. Finiteness is assumed by hypothesis one.

A1 - Each interior node of G in R is going to determine a space in a layout-scheme. If such a space is to be rectangular, and surrounded by rectangular spaces adjacent to it, then each interior node must have four or more edges since a rectangular space cannot be completely surrounded by less than four other rectangular spaces adjacent to it.

A2 - First of all, each face will necessarily have three or more edges because the graph is simple. Now, in the dual R' of R, each node stands for a corner where several spaces meet, and corresponds to a face of G in R. Since by hypothesis three all spaces are rectangular, each of the spaces meeting in a node of R' will occupy, around the node, an angle of either 90 or 180 degrees. Because around the node only 360 degrees are available, this gives a maximum of 360/90 = 4 spaces meeting in any corner; i.e. each face of G in R will have to have a maximum of four edges, otherwise it would give rise to a node in R' with more than four edges, thus dividing the region around that node into more than four regions, making it impossible for all those regions to be rectangular.

Justification of conditions B

B0 - In a R-layout-scheme the edges form partitions between spaces. Since the spaces are rectangular, once an external reference direction is provided, the whole layout scheme can be oriented in such a way that the partitions between the spaces are either parallel or perpendicular to that direction. Thus, we have adopted the designations horizontal and verticalto express the relative orientation of the edges of any layout scheme.

Up to this point we have been using 'H' and 'V', or 'E' and 'V', when referring to the edges of G in R. Because a one-to-one correspondence exists between the edges of G in R and the edges of the dual R' of R, we way also refer to each edge of G in R as being necessarily either horizontal or vertical. Hence the reason for having to assign to each and every edge of G in R one of two different colours, denoted by 'E' and 'V'.

Whereas conditions A establish necessary properties G must have for at least one PHV - assignment to be possible, conditions B indicate the restrictions to which a HV - assignment must comply with for being a PHV - assignment, since it is clear from their justifications that not all HV - assignments are permissible.

B1 - To a triangular face of G in R there corresponds a node in the dual R' of R with three edges. These edges are the partitions dividing the region around the node into rectangular spaces. These partitions can only be horizontal or vertical, as has been argued. It is apparent that no more than two partitions of the same type (i.e. horizontal or vertical) can meet in any given node. It follows that the edges of a triangular face cannot all have the same colour.

B2 - The same way as before, each quadrangular face gives a node where four partitions meet; two of them will have to be horizontal and the other two vertical. Furthermore, the horizontal and vertical partitions will have to alternate around the node. Condition B2 follows.

B3 - An interior node I of G in R of any R-layout-scheme obtainable by a PHV - assignment gives rise to a rectangular space which is completely surrounded by other rectangular spaces. The partitions between that space S and the surrounding ones are the four sides of its boundary. To each side of S there corresponds a number of edges of G in R, one for each surrounding rectangular space adjacent to S on that side. Thus, the four sides of S determine four groups of edges around node I, each group corresponding to the spaces adjacent to S on one of its sides. Since the four sides of S are alternately horizontal and vertical, the groups of edges around node I are also alternately horizontal and vertical, with all the edges of any one group being of the same type. By a horizontal (vertical) group of edges of a node I, we mean a group of sucessive edges around I which are all horizontal (vertical). Of course, all edges of a group of edges have the same colour. Condition B3 follows.

Comments on conditions A and B

It should be remarked that colouring conditions B expressed above take into account all necessary rules for colouring a realization R of a graph except the exterior nodes; i.e. there are no other rules besides those which derive from them. Thus, the set of necessary rules given is also a sufficient set of rules for colouring R, exception being made for the exterior nodes, whose colouring rules are dependent upon the contour description prescribed by Hypothesis 6.

In fact, conditions B apply to all edges through B0, to all face through B1 and B2, and to all interior nodes through B3.

Conditions C

Since without a further condition on the exterior nodes we cannot properly speak of PHV - assignments, we shall give three contour conditions, C0, C1 and C2.

C1 is the most general condition on contour forms: i.e. it admits all contour forms complying with Hypothesis 1. Condition C2 expresses the requirement to be met if the contour is rectangular. C0 is a rectangularity condition on any exterior space of any contour.

Le N1 be the number of exterior nodes with exactly one group of edges. Let N2 be the number of exterior nodes with exactly two differently coloured groups of edges and with its two boundary edges of a different colour. Let N4 be the number of exterior nodes with exactly four alternately coloured groups of edges around them. Intuitively, a node contributes to N1 if it gives rise to a rectangular space occupying exactly two convex corners of the contour of the layout scheme; to N2 if just one convex corner is occupied; to N4 if just one concave corner is occupied. If no corner is occupied then the node contributes to N0 (i.e. all other nodes).

C0 - No exterior node may have more than four alternately coloured groups of edges around it. Thus it can only have one, two, or four groups, because three groups is impossible to have.

C1 - All contour forms allowed by Hypothesis 1 comply to the following equation:

$$(2 \times N1 + N2) - N4 = 4$$

This result means intuitively that the difference between the number of convex and concave rectangular corners of the contour of the layout scheme must be four; i.e. the layout must be closed.

C2 - All rectangular contour forms and only them comply to the following equation, besides the general equation stipulated by condition C1:

$$N4 = 0$$

The introduction of this condition into the general equation of C1 gives the intuitive result that the total number of convex corners of a rectangular contour form is four.

Methods for imposing other contour forms besides the recantgular will be found in (2). Essentially they consist in specifying abstract finite state machines which accept only certain sequences of corner types.

Conditions A, B and C place a restriction on the classe of planar realizations that can produce R-layout-schemes through PHV-assignments. Some such restrictions can eventually be expressed in purely "geometrical" terms. Thus, one is confronted with the question of deciding to what extent should ever more sophisticated "geometrical" conditions be derived. If on the one hand they make it possible to "filter" unproductive realizations, on the other, being difficult to evaluate, they may well lead to a decrease in efficiency. These comments are meant to call into attention two different perspectives from which to envisage conditions B and C:

(1) as originating the rules to be followed in the process of colouring a planar realization.

(2) as providing tests to be carried out at each step of that colouring process.

4 - THE COLOURING METHODS

Once a given realization is known to obey conditions A then, to obtain all its PHV - assignments, one must obtain all possible HV - assignments obeying conditions B, CO, and Cl. Of course, it may happen that no PHV - assignment exists for the given realization. However, from a problem - solving perspective, once it is known that a given realization conforms to conditions A, the methods for determining if there exists at least one PHV - assignment and of finding them all, can be fused in to one and the same process. Such methods are said construtive because while determining whether or not there exists at least one PHV - assignment for a given realization, one already obtains, in the affirmative case, the possible PHV - assignment (s).

A first question is whether there exists a systematic process of engendering all PHV - assignments. There is one easily conceived process. For N edges, the number of different HV - assignments is 2^N. However, we do not distinguish two HV - assignments resulting one from the other by exchanging the H's and the V's throughout. Such a change is immaterial because conditions B and C are symmetrical with respect to 'H' and 'V', and because such an exchange amounts to a simple rotation of the layout-scheme by ninety degrees. Thus, the number of "distinguishable" HV - assignments reduces to $2^N/2 = 2^{N-1}$. The process itself consists in generating all HV -assignments and pruning them with the colouring and contour conditions. Its principal drawback is its lack of selectiveness, given that the number of PHV - assignments is always much smaller than the number of HV - assignments.

A sensible way of colouring will be one that explores the structure of the problem so as to avoid the repetition of assignments and avoid as many as possible non-permissible HV - assignments. Between the two extremes of systematically trying out all assignments and of randomly generating assignments, the methods for colouring will have to be able to take advantage from information about the problem structure although it may not be guaranteed that they will make the most out if it. In this they will be heuristic. Which does not mean that they cannot be thorough.

To gather information about the structure of the problem of colouring any particular realization, we will have to carry out a detailed analysis encompassing the generality of all possible colouring "situations"; and by colouring situation we mean any partially coloured stage of a realization. The set of all colouring situations of a realization defines the state space for the problem of colouring that realization. This analysis will consist in discerning relevant properties of a realization from the point of view of colouring it, and in establishing a typology of colouring situations fundamented on those properties. The importance of such a classification is that, by defining appropriate sets of colouring situations, one can construct a hierarchy of problems and subproblems able to cope with the whole variety of candidate graphs.

Colour Properties and Colour Consequences

By colour property we mean a set of colouring situations specified in "geometric" terms (i.e. nodes, edges, faces, and boundary).

By colour consequence we mean a unique assignment of colour to a set of non-coloured edges of a colouring situation, which follows inevitably from the application of colouring conditions B. The reason for not taking conditions C into account is that, since the contour forms desired may vary, we do not use them for extracting colour consequences in the general program. Instead, they are used for pruning undesirable or unpermissible contour forms of completed colour assignments. This decomposing of a realtization into contour and interior divides the colouring problem in two, making it easier to conceive and implement man/machine interaction facilities aimed at the reformulation of contour requirements, separate from those providing for the modification of interior contraints. Le us examine some general colour conseqences: 1) If a triangular face has two of its edges assigned one same colour, and the third edge uncoloured, then, by condition B1, the latter must be assigned the other colour. 2) In a quadrangular face with one, two, or three of its edges already coloured permissibly, the assignment of colours, to the remaining non-coloured edges is determined by condition B2. 3) Any interior node with four edges, with one, two, or three of them already permissibly coloured, has the assignment of colour to the remaining uncoloured edges determined by condition B3. 4) in interior nodes with more than four edges there exist a great variety of colour consequences. All result, however, from applying condition B3 to the node. They can be categorized into 25 cases, where 9 present a non-empty colour consequence, 9 an empty colour consequence, and 7 are impermissible colouring situation. Cf. (2) for the traits which allow such a typology.

Components

Let C be a graph made entirely of nodes with exactly four edges, linked together. Let D be a graph made entirely of "quadrangular" faces linked together. Colouring an edge of C or D has as a consequence the colouring of all their edges.

We call component to a subgraph of a realzation provide structural information about its R - layout-schemes. Take a component of R and obtain all its colour consequences. The coloured subgraph of G in R obtained defines a nucleus of R, in case no colouring conditions are violated. A nucleus is invariant in every R-layout-scheme of R. If the nuclei are pairwise disjoint and some colour condition is infringed, there are no PHV-assignments whatsoever. If not, everytime the colour consequences of a nucleus extends into some other nucleus, and an incompatibility arises between the two, an exchange of colours is tried for their components. If it does not lift the incompatibility, the two nuclei are incompatible whatever way they are coloured. Otherwise, a partial subgraph has been permissibly coloured which may extend to some other nucleus, etc.. Each component determines a set of edges acting as a whole from the colouring point of view, thus defining stable solved subproblems.

Generating all solutions

After the nuclei have been identified, colouration of the graph proceeds by assigning an arbitrary colour to some non-coloured edge and extracting all colour consequences. If no colour condition is violated another edge is chosen for colouring etc. Whenever some colour condition is infringed, backtracking must take place. The most recently arbitrarily coloured edge is given a different colour, unless both colours have already been tried for it. In that case, backtracking goes back to the next most recently arbitrarily coloured edge, etc. When all edges are eventually permissibly coloured, a solution has been obtained. To obtain all, simply backtrack after each solution found until there are no more arbitrarily coloured edges to backtrack to.

Which edges are chosen for arbitrary colouration? Here there is scope for heuristic rules. These will combine to chose those edges with the greatest potential for a large number of colour consequences. They are heuristic because their combination cannot guarantee the best choice in all situations (2).

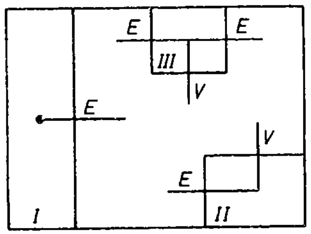
5 - ASSIGNING DIMENSIONS

In this section a method is given for dimensioning the spaces of the R-layout -scheme corresponding to a permissibly coloured realization with an indented (possibly rectangular) contour form. An indented contour is one with no more than two consecutive concave corners, and where any two such sequences are separated by at least two consecutive convex corners. Ie. any concave indentation must be "undone" before another is possible, so that there is no chance the layout will fold onto itself.

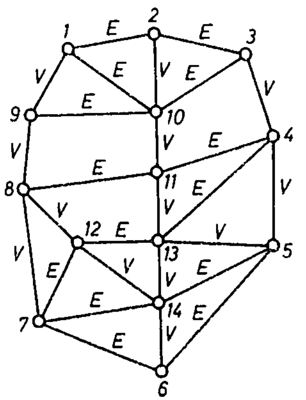
In this paper we deal solely with the case of finding modular dimensions. The general case of dimensions within specified limits is an extension of this method. It is dealt with by means of graph flux algorithms, combined with heuristic rules for dimension compatibilization which will find a solution if there is one. Interactive modification of dimensions is also allowed. No attention is given here to the problem of finding drawing coordinates once the dimensions of each space are known. These topics are dealt with in (2).

Rectangular contour modular layout

First of all, we will treat the case where the contour is rectangular. The hypotheses of contour rectangularity and realization non-separability bring as a consequence that there can only be three types of exterior (rectangular) spaces. They are illustrated in the next figure, where each coloured line belongs to a group having the indicated colour and at least one edge. Spaces of type I correspond to exteriors nodes with just one group of edges; type II spaces to exterior nodes with just two groups of edges and having their two boundary edges of a different colour; type III spaces to exterior nodes with exactly two groups of edges, and having their two boundary edges belonging to the same group. Let N1, N2, and N0, be the number of spaces of type I, II and III, respectively. Intuitively, a node contributes to N1 if it gives rise to a space occupying exactly two convex corners of the contour of the layout; to N2 if just one convex corner is occupied; to N0 if no corner is occupied.



The following facts hold: (1) there cannot be more than two nodes of type I; (2) there cannot be more than four nodes of type II; (3) only two nodes of type II may coexist with one node of type I, there not being any other possible combination of these two types. In a rectangular layout the total horizontal (vertical) dimension "entering" the layout must be equal to the total horizontal (vertical) dimension with "exits" from it. Consider the following permissibly coloured realization:



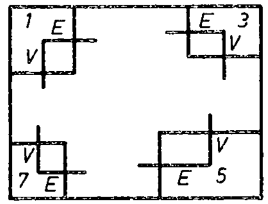
Let us now identify the exterior nodes according to the three types defined:

I - there are none

II - 1, 3, 5 and 7

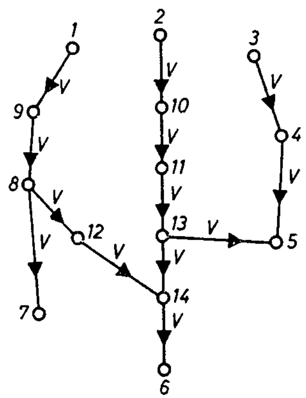
III - the remaining ones

Consequently, the layout scheme to be derived from the given realization will respect the form:



This way, the nodes in between nodes 1 and 3, inclusive, delimit the total horizontal dimension. The same of nodes 5 and 7, inclusive. Similarly, the total vertical dimension is delimited by nodes 3, 4, and 5 or, equivalently, by nodes 7, 8, 9, and 1.

Let us now draw the following subgraph, obtained from the previous graph by taking only those edges that are coloured with V. Recall that to a V coloured edge there corresponds in the layout a horizontal partition segment. We shall call this subgraph the horizontal subgraph of the realization. The vertical subgraph of the realization is similarly obtained. The sense of the arrows is justified as follows:



(1) For interior nodes. Since to each space, and thus to each node, there corresponds a horizontal (vertical) dimension "entering" equal to the horizontal (vertical) dimension "exiting" the edges of each of the two pairs of opposite groups of an interior node must have their arrows oriented differently with respect to the node.

(2) For exterior nodes. From the point of view of edge orientation, the two edges belonging to the boundary in a type III node belong to two different edge groups, although they are consecutive and have the same colour; furthermore, the two groups are opposite groups. Bearing this in mind, the orientation of the two op

posite groups of edges with the same colour of a type III node, is performed as for interior nodes. As for the orientation of the isolated colour groups in the three types of exterior nodes, the only condition is that in each group all the edges must be given the same arrow sense relative to the nodes; which arrow sense it is depends upon the orientation of adjacent nodes. Note that in non-rectangular contours there is another type of exterior node, which has four colour groups. The orientation of its edges follows the rule used for interior nodes. We call them type IV nodes.

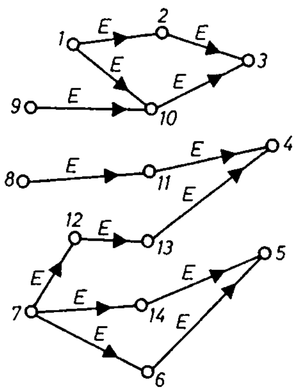
The foregoing rules are sufficient for orienting all the edges of a permissibly coloured realization, as long as any arbitrary orientation is first given to any two edges of different colour. The choice of those two edges and of their orientation defines the orientation of the whole layout scheme.

Obtaining the modular dimensions. Consider again the previous figure. In that subgraph the total horizontal dimension "enters" through nodes 1, 2 and 3, and distributes itself by all other nodes in the way indicated by the arrows: (1) space 9 will be immediately "below" space 1; (2) space 8 will be immediately "below" space 9, and so indirectly "below " space 1, and so on; space 14, for example, will be directly "below" both space 12 and space 13; (3) note that directly "below" space 8, for example, will be spaces 7 and 12, which thus become spaces indirectly "below" spaces 9 and 1.

The information of this type contained in the whole subgraph is expressed by the following table of columns:

1	1	2	2	3
9	9	10	10	4
8	8	11	11	5
7	12	13	13	
	14	14	5	
	6	6		

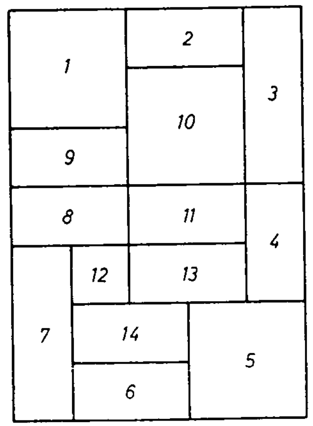
To obtain the table of rows we consider now the vertical subgraph.



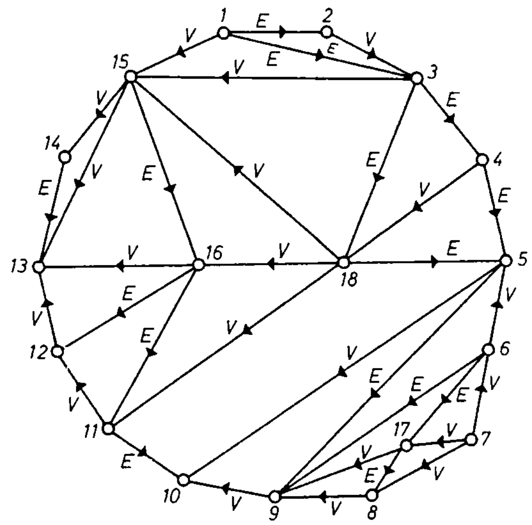
1	2	3	
1	10	3	
9	10	3	
8	11	4	
7	12	13	4
7	14	5	
7	6	5	

Once we obtain both tables, the dimensions of each space are found by counting the number of occurences of that space in each the tables. The next layout ensues.

Modular Indented Layouts. We do not need another method. The only new type of node is the exterior node with four groups of edges and the way to orient them is the same as for interior nodes.



Another example:



Types of exterior nodes:

I - 7
II - 1, 2, 10, 13, 14
III - 4, 6, 8, 9, 12, 15
IV - 3, 5, 11

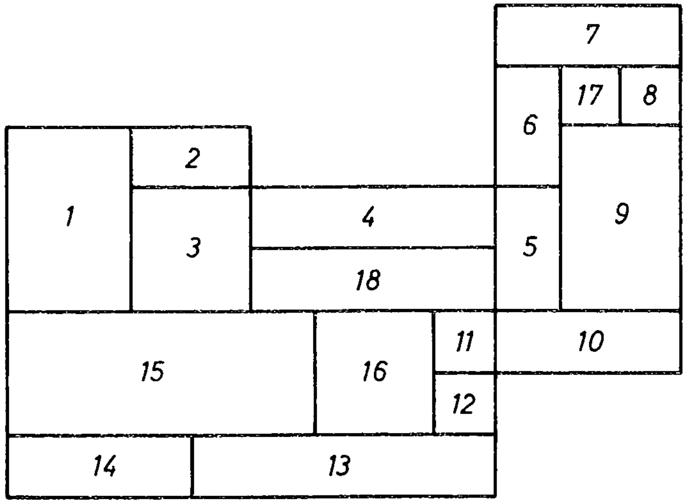
Interior nodes: 16, 17, 18

Horizontal Dimension Table (columns):

1	1	2	2	4	4	4	4	7	7	7
15	15	3	3	18	18	18	18	6	17	8
13	14	15	15	15	15	16	11	5	9	9
		14	13	13	14	13	12	10	10	10
							13			

Vertical Dimension Table (rows):

1	2			
1	3	4	5	9
1	3	18	5	9
15	16	12		
15	16	11	10	
14	13			
6	9			
6	17	8		
7				



6 - CONCLUSION

Artificial intelligence concepts and paradigms were paramount in providing us with a framework for approaching and tackling the envisaged class of problems. In particular, the notions and techniques of problem representation, problem decomposition, space-state, search strategy, heuristic search, and formal languages recognizing automata. The graph labeling paradigm for representing constraints has independently been applied to the recognition of visual scenes in computer vision (4). Good introductions to basic artificial intelligence are (1), (3) and (5).

We foresee the artificial intelligence approach to became more and more important in computer aided design in the near future. Use of programming languages specially developed by the artificial intelligence comunity will be crucial.

7 - REFERENCES

(1) - NILSSON, N. (1971) - Problem solving methods in artificial intelligence, McGraw-Hill.

(2) - PEREIRA, L.M. (1974) - Layout schemes from adjacency graphs. PhD thesis, Brunel University, published by Laboratório Nacional de Engenharia Civil, Lisbon.

(3) - RAPHAEL, B. (1976) - The thinking computer, WH Freeman.

(4) - WALTZ, D. (1975) - Understanding line drawings of scenes with shadows, in The psychology of computer vision, P. Winston (editor), McGraw-Hill.

(5) - WINSTON, P. (1977) - Artificial Intelligence, Addison-Wesley.

Editor's Note : The discussion of L.M. Pereira's paper follows M. Henrion's paper.