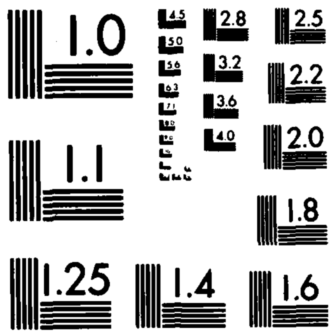
AD-A171 339 BLOCK PLAN CONSTRUCTION FROM A DELTAHEDRON BASED ADJACENCY GRAPH(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH D W KEENAN 1986 1/2

UNCLASSIFIED AFIT/CI/NR-86-128T F/G 12/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A171 339

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFIT/CI/NR 86-128T
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): Block Plan Construction From A Deltahedron Based Adjacency Graph
5. TYPE OF REPORT & PERIOD COVERED: THESIS/~~DISSERTATION~~
6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): David Wayne Keenan
8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: AFIT STUDENT AT: The University of Arizona
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: AFIT/NR WPAFB OH 45433-6583
12. REPORT DATE: 1986
13. NUMBER OF PAGES: 125
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
15. SECURITY CLASS. (of this report): UNCLAS
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report): APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

DTIC ELECTE AUG 2 9 1986 E

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
18. SUPPLEMENTARY NOTES: APPROVED FOR PUBLIC RELEASE: IAW AFR 190-1

Lynn Wolaver 13 Aug 86
LYNN E. WOLAVER
Dean for Research and Professional Development
AFIT/NR

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
20. ABSTRACT (Continue on reverse side if necessary and identify by block number): ATTACHED.

DTIC FILE COPY

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

86 8 28 027

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

BLOCK PLAN CONSTRUCTION FROM A DELTAHEDRON BASED ADJACENCY GRAPH

by

David Wayne Keenan

A Thesis Submitted to the Faculty of the

DEPARTMENT OF SYSTEMS AND INDUSTRIAL ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE
WITH A MAJOR IN INDUSTRIAL ENGINEERING

In the Graduate College

THE UNIVERSITY OF ARIZONA

1 9 8 6

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under the rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgement the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: ____________________

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

J. W. Giffin
Assistant Professor of
Industrial Engineering

04/11/86
Date

ACKNOWLEDGEMENTS

I would like to thank Dr. John W. Giffin for his encouragement, inspiration, and advice throughout this endeavor. I am indebted to thank The United States Air Force for sponsoring my education here at The University of Arizona. Most of all I would like to thank my wife Joyce and children Timothy and Breanna for their patience with me during my studies.

TABLE OF CONTENTS

TABLE OF CONTENTS--Continued

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS--Continued

Page

LIST OF ILLUSTRATIONS--Continued

LIST OF TABLES

ABSTRACT

A method for the construction of a rectangular geometric dual from a Deltahedron based maximally planar adjacency graph is given along with its computer implementation. In addition, a method and its computer implementation for the addition of areas to form a block plan is given. Comparisons with output from other computer methods is included. Possible extensions include the construction of a rectangular geometric dual with areas for all maximally planar adjacency graphs.

CHAPTER 1

INTRODUCTION

The problem of where to locate different facilities within a structure is a very old one indeed. Whenever a building serves more than one function with each function having specific equipment or space requirements, choices must be made to determine the best location for each function. Even the simple problem of locating a bed, fireplace, and table within a cabin requires choice among differing alternatives. This problem however, is not limited to location of rooms or functions within a building. Extensions can be made to include problems ranging from the location of different buildings on a site to electronic components on a circuit board. Many approaches to this problem have been taken over a great span of time. One approach sometimes referred to as iconic, includes building models of the different components and physically placing them in different locations within a model of the building. The analog approach is one that transforms the original problem into some analogous problem and then solves this analog problem. The solution for the original problem is then obtained by a reverse transformation. The approach that as of late has had by far the most attention is the symbolic or mathematical approach.

This thesis deals with the extension of several specific mathematical approaches. In particular, the development of the spacial relationships infered by the results of a special class of graph theoretic methods known as Deltahedron Heuristics.

The purpose of this thesis is to develop a systematic approach to construct a rectangular geometric dual from these Deltahedron based adjacency graphs and include areas to form a block plan. Chapter 2 describes the problem as well as some past work in the area. In addition to a systematic approach for developing a rectangular geometric dual and its block plan, a computer implementation of this method is included in chapter 3. Comparisons with two other computerized methods are given in chapter 4 while chapter 5 contains conclusions and suggestions for further work.

CHAPTER 2

PROBLEM STATEMENT AND PAST WORK IN THE AREA

The general purpose of all of the layout methods proposed is to specify locational relationships between facilities so as to optimize some performance criterion. These relationships are generally of two forms, the adjacency of facilities and the distance between facilities. The most common objective functions used to measure the performance criterion are maximization of total achievable adjacencies and minimization of total transportation cost. When maximizing the sum of adjacencies, each adjacency between two facilities has some specified score and the total of all adjacencies realized represents this total adjacency score. The minimization of total transportation cost usually assumes that transportation cost is a function of distance and therefore the overall pairwise distance between facilities that have some material being transferred must be minimized.

2-1 Classical Layout Approaches

The first formal mathematical model of the facility layout problem was in the form of the Quadratic

Assignment Problem proposed by Koopmans and Beckmann [1957]. This formulation takes the approach of dividing each facility into some number of equal size subfacilities, usually using the size of the smallest facility. The task is then to assign each subfacility to a location on an orthogonal grid representing the planar site, so that the total transportation cost is minimized and that each facility occupies a contiguous region. It has been shown that this problem has no algorithm for its solution that is polynomially bounded in time and belongs to the class of problems termed NP complete. This means that only relatively small problems can be solved to optimality using this method. Therefore, attempts have been made to find a good heuristic to provide solutions to this problem. Some of the well known methods are briefly described below.

2-1.1 Terminology, Notation, and Definitions

The following terms and notation are defined in the context of facility layout.

[1] Construction Heuristic. A construction type heuristic is one that constructs a layout by adding facilities one at a time until a completed layout is achieved.

[2] Improvement Heuristic. An improvement heuristic is one that requires an initial layout as

input. The heuristic then improves the layout by some local exchange technique until no further improvements can be made.

[3] Relationship Chart. The relationship chart, or REL chart, is an attempt to quantify the importance of relationships between facilities using closness ratings [Muther, 1961]. The closeness rating is a score, R_{ij}, that is achieved when the two appropriate facilities are adjacent. The ratings, their definitions, and frequently used scores for two common methods are listed in Table 2.1.

[4] Adjacency. Generally two facilities are considered adjacent if they share a common wall or divider of some minimal tolerance length that separates one from the other. One exception to this definition is the criterion of ALDEP which in addition to the above description, considers two facilities adjacent if they are diagonal to one another at the meeting of four walls.

[5] Initial Layout. The initial layout is the layout used for a starting point in improvement type heuristics.

[6] Flow Data. This is a matrix, sometimes refered to as a From-To chart, that represents the number of trips or volume of material flow per time period being made from one facility to another.

[7]_Cost_Data. This is also a matrix however it contains the cost to move one unit of distance between each facility.

[8]_Plant_Layout. Since the majority of layout planning has dealt with the design of manufacturing structures, the building or collection of buildings is commonly referred to as the plant; hence the term plant layout.

Table 2.1 Common REL Chart Ratings, Definitions, and Scores

Rating	Definition	Score ALDEP	Score CORELAP
A	Absolutely necessary	64	6
E	Especially important	16	5
I	Important	4	4
O	Ordinary closeness OK	1	3
U	Unimportant	0	2
X	Undesirable	-1024	1

2-1.2 Muther's Systematic Layout Planning

Muther, [1961] developed the organized approach to plant layout known as Systematic Layout Planning [SLP]. The three main areas of concern for this method are Analysis, Search, and Selection as illustrated in the method schematic shown in figure 2.1.

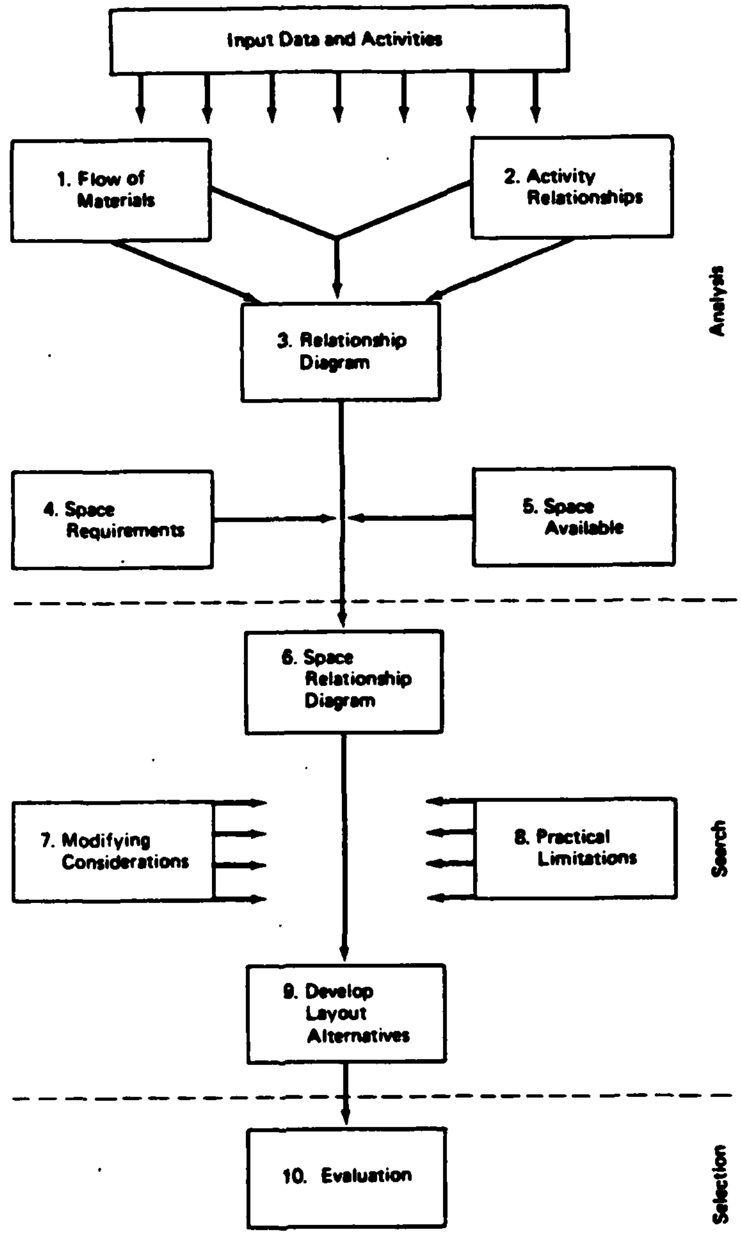
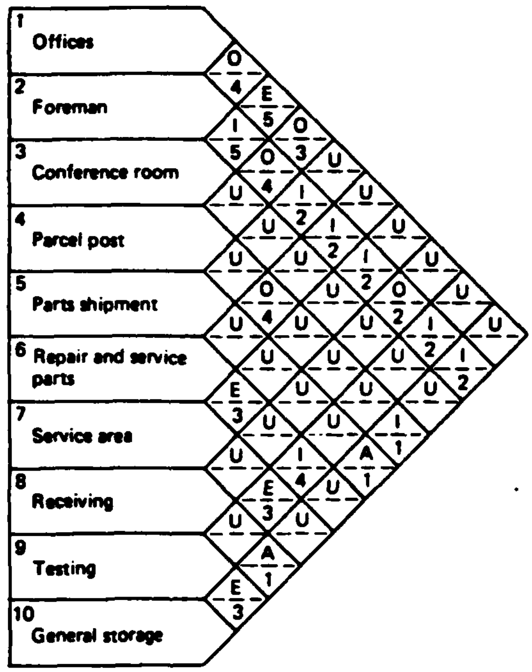


Figure 2.1. Systematic Layout Planning Procedure

(1) Analysis. The analysis begins by gathering data about the specific plant layout to be designed. Information concerning the flow of materials and workers

within the plant is collected in the form of a flow and a cost from-to chart. Additionally, quantifiable information about the desirability of having each pair of facilities within the plant adjacent to one another is collected in the form of a REL chart [see figure 2.2[a]]. The information from these three is then used to come up with a relationship diagram. The relationship diagram is constructed by arranging equal area squares that represent each facility into different configurations until one is found that has the desired level of preferred properties measured by the from-to and REL charts [see figure 2.2[b]]. This is often an iterative trial and error scheme that is performed manually with evaluation often done very subjectively and therefore many and possibly preferred arrangements may be overlooked. Space requirements for each facility are then determined as well as the total available space.

[2]_Search. The search operation is started by developing several space relationship diagrams [see figure 2.3[a]. This involves combining the relationship diagram with the space requirements and space availability to construct diagrams that have the relationships and areas suggested during the analysis stage. These space relationship diagrams are then condensed into a block plan as illustrated in figure



Code	Reason
1	Flow of materials
2	Ease of supervision
3	Common personnel
4	Contact necessary
5	Convenience
6	
7	
8	
9	
10	

Rating	Definition
A	Absolutely necessary
E	Especially Important
I	Important
O	Ordinary closeness OK
U	Unimportant
X	Undesirable

Figure 2.2. (a) Relationship Chart

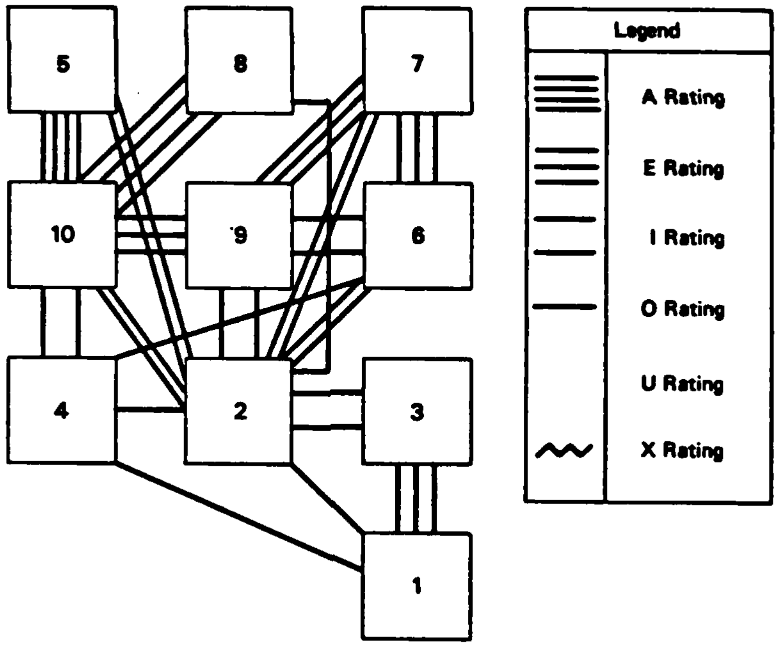
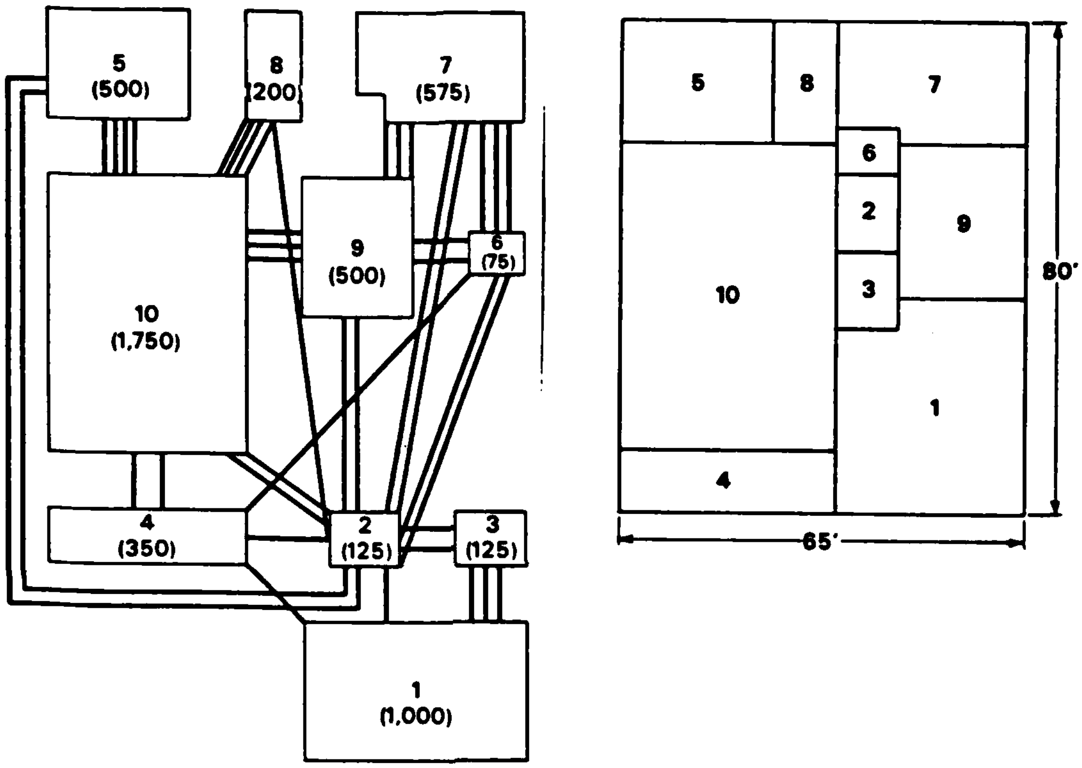


Figure 2.2--Continued. [b] Relationship Diagram

2.3[b]. This block plan is finally combined with any modifying considerations and practical limitations that are developed, to come up with alternatives for the plant layout.

[3] Selection. The final operation is to decide among the alternatives or to make any data changes that prove necessary and repeat the process.

All other methods presented here fit within the general context of this procedure. Any layout will involve collecting data and some selection among alternatives. The difference arises with the choice of the method one uses to construct the block plan from the data. The next three approaches discussed are well known classical computer based methods for developing a block plan.



(a) (b)

Figure 2.3. (a) Space Relationship Diagram (b) Block Plan

2-1.3 ALDEP

A method that was developed within IBM and originally presented by Seehof and Evans [1967] is called the Automated Layout Design Program, commonly referred to as ALDEP. ALDEP is a construction type heuristic as it requires no initial solution to begin, however it uses its past solutions as a basis for comparing new ones to see if any improvement has been made and therefore some improvement does take place. ALDEP divides each facility into subfacilities of some common square dimension based upon the scale specified. A facility is then chosen at random and layout is begun from the upper left corner of the layout. The subfacilities of the initial facility are added to the layout in vertical strips of a specified 'sweep width' until its area is exhausted. The REL chart is then scanned for a facility that has an A or E rating with the existing facility and it is then placed in the layout. As before the new facility is laid in a strip fashion until its area is exhausted. The vertical scanning nature of these strips is illustrated in figure 2.4. This addition process is then repeated until no facilities remain or until there are no facilities with an A or E rating with the last facility added. If the latter is the case, a facility is chosen at random and the process is continued. The score for this method is found using the values from REL chart. The eight squares that surround each facility are scanned and the score recorded. After a score is recorded it is deleted from the matrix to eliminate the possibility of including the same adjacency twice. The total of these values is the score for the layout. Usually the entire process is run many times with each improvement in score becoming the new goal for the program to attain. Runs that do not achieve the goal are rejected and the entire process stops when no improvement is made. Alternatively, a collection of good solutions can be developed to provide different options for the selection process. An example of the output produced is included in chapter 4.

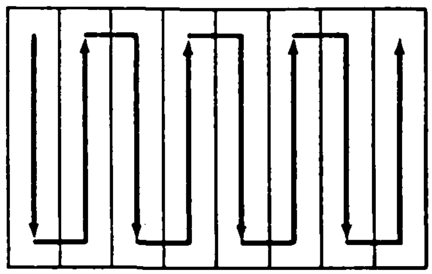


Figure 2.4. Vertical scanning pattern used by ALDEP

2-1.4 CORELAP

CORELAP is the acronym for Computerized Relationship Layout Planning and was developed by Lee and Moore [1967]. A number of improvements to the original method have been added since its introduction and the version known as CORELAP 8 will be discussed here. As with ALDEP, this is a construction type heuristic. This method begins by choosing the first facility according to its Total Closeness Rating [TCR], calculated for facility i by summing the REL chart scores from facility i to all others. The facility with the highest TCR is chosen to be added first, and placed in the center of the layout. Next a facility that has an A adjacency score with the first facility is selected. If no facility with an A rating is found, an E rating is searched for. If no E rating is found, the method continues down the hierarchy of scores until a U is reached. If no facility with a score of U or better is found, the facility with the highest TCR is chosen. If there is more than one facility with the same score, the facility with the highest TCR is chosen. The same type of search is employed at all sucessive steps with the search started by looking for a facility with an A adjacency to the first facility. If none is found, an A adjacency with the second facility is desired, followed by an E with the first, an E with the second, an I with the first, etc. All facilities are added to the exterior of the existing arrangement and are rectangular in shape. They are placed in a position that will yield the highest placement rating and boundary length,where the boundary length is the length of the boundaries common to the new facility and the existing layout. Some different configurations possible are illustrated in figure 2.5. The placement rating is the sum of the weighted ratings between the department being added to the layout and its neighbors if it is placed there. The weights are assigned to the adjacency ratings by the user. Therefore the score used for the TCR is not necessarily the same as that used to score the placement of each facility within the layout. An example of the output from this method is also included in chapter 4.

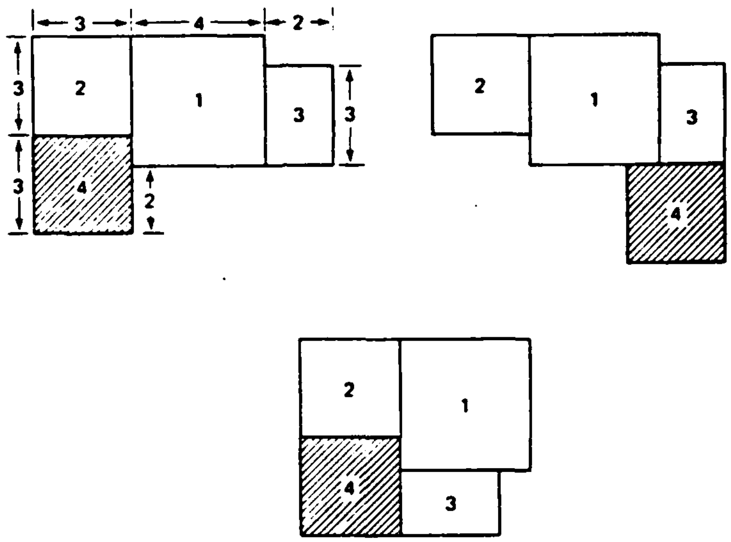


Figure 2.5. CORELAP's placement method

2-1.5 CRAFT

CRAFT is an improvement type heuristic and was introduced by Armour and Buffa [1963]. In addition to differing from ALDEP and CORELAP in the type of heuristic used [construction versus improvement], CRAFT employs a entirely different method for evaluating a layout.

Unlike ALDEP and CORELAP, CRAFT attempts to minimize transportation cost where this cost is expressed in terms of distance traveled. This is therefore an attempt to provide a solution to the QAP mentioned earlier. As an improvement heuristic, CRAFT requires an initial layout in order to apply its improvements. The score for a layout is the cost per unit distance (cost data) to move an item, multiplied by the rectilinear distance between facility centroids, multiplied by the number of trips required (flow data), for all pairs of facilities in the layout. The next step is to consider the exchange of two or three facilities within the layout. The possible combinations include 1) two-way interchanges, 2) three-way interchanges, 3) two-way followed by three-way interchanges, 4) three-way followed by two-way interchanges, and 5) the best of two-way and three-way interchanges. Exchanges of facilities are only possible if the facilities are adjacent to one another or if their areas are equal. The search for the best of these is done by interchanging the centroids which are used in the distance calculations as an estimate of the actual cost. The best exchange, lowest score, is then made and centroids recalculated according to the new shape of the facilities. If a savings still exists the process continues and if not the old layout is maintained and a different interchange is attempted. When no improvements can be made the process stops. A drawback with the method is that there appears to be no consistent method for the physical exchange of adjacent facilities of varying areas.

2-2 Graph Theoretical Approaches

2-2.1 Terminology, Notation, and Definitions

The following terminology and notation is defined:

[1] Graph. A graph is a pair of sets (V,E) where V is finite and not empty. The elements of V are called vertices and the elements of E are distinct pairs of vertices called edges. If there is no direction associated with the edges, they are known as undirected edges. If all edges are undirected, the graph is said to be an undirected graph.

[2] Weighted Graph. A graph that has a weight, W_e, assigned to each edge, e, is known as a weighted graph with W_e usually being an element of the positive real numbers.

[3] Complete Graph. A complete graph, denoted K_4, is one in which all pairs of vertices are joined by an edge. A complete undirected or symmetric graph has $(n(n-1))/2$ edges.

[4]_Planar_Graph. A graph is said to be planar if it can be drawn in the plane such that no two edges intersect except at a vertex to which both are incident.

[5]_Maximally_Planar_Graph. A graph is said to be maximally planar if it not possible to add an edge and still maintain planarity. Due to the fact that all faces of a maximally planar graph are triangles, a maximally planar graph is often known as a triangulation. A Maximally planar graph contains 3n-6 edges [Euler, 1752].

[6]_Tetrahedron. A tetrahedron [K_4] is a complete graph on four vertices which is also maximally planar [see figure 2.6].

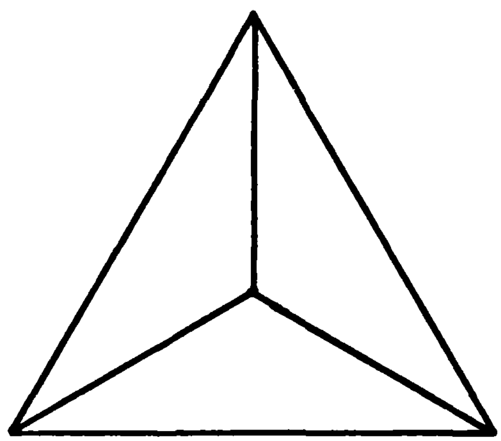


Figure 2.6. Tetrahedron

[7]_Deltahedron. A deltahedron is a graph that is constructed by beginning with a tetrahedron and adding vertices by the insertion of an additional vertex into a

triangle and adding edges from the new vertex to each of the three vetrices that define the triangle. Due to this fact a deltahedron must contain at least one vertex of degree three (three edges incident with it).

[8] Maximally Planar Adjacency Graph. A maximally planar adjacency graph is a maximally planar graph whose edges represent adjacency between pairs of facilities.

[9] Geometric Dual. The geometric dual of the maximally planar adjacency graph is a spacial representation of the facilities that are represented by the vertices of the graph. The edges of the graph represent the adjacency of two facilities in the dual. If a graph is maximally planar then its dual is also maximally planar or in other words no further adjacencies in the dual can be established without violating the planarity of the dual (Whitney, 1931).

[10] Rectangular Geometric Dual. For this discussion, a rectangular geometric dual is a geometric dual that contains only rectangular, L and T shaped areas.

All graph theoretical approaches presented here are of the construction type. One starts with a complete graph on N vertices corresponding to a REL chart with zero weight edges added if necessary, and attempts to find a maximally planar subgraph on the complete graph that has maximum weight since without loss of generality, with nonnegative weights, an optimal solution will be maximally planar. The problem of starting with the complete graph and deleting edges until it is maximally planar is a relatively difficult and very time consuming problem due to the methods required to check for maximal planarity. The methods shown here use construction techniques that start with either a graph that is not maximally planar and iteratively build it up until it is maximally planar or a graph that is maximally planar and then add vertices and edges to it in a specific manner so that it will always remain maximally planar. Several of the methods start with a complete graph on four vertices, K_4. There are basically two methods for determining which four vertices should make up this initial tetrahedron. The first is the greedy approach which finds the highest weight tetrahedron among all possibilities. The other is formed by first summing the scores of all columns from the square adjacency matrix. The vertices are then sorted in non-increasing order according to these column sums. Then the vertex with the highest adjacency rating to all other vertices is chosen first. It has been shown [Giffin, 1984] that there is empirically no clear difference in final triangulation solution quality for either starting procedure. The objective of all methods that follow (with the exception of Super Deltahedron] is to maximize the adjacency score where the values of having two facilities adjacent are the same as those used in ALDEP.

2-2.2 The Wheel Expansion Heuristic

The Wheel Expansion Heuristic [Eades, Foulds, and Giffin, 1982] begins with an initial tetrahedron and uses an operation known as a wheel expansion to add sucessive vertices to the graph. It has been shown that the wheel expansion operation is sufficient to form all maximally planar graphs possible if the starting point is K_4. An example of wheel expansion is illustrated in figure 2.7. The choice of vertex and location for expansion involves finding the vertex and expansion point that has the highest increase in adjacency score.

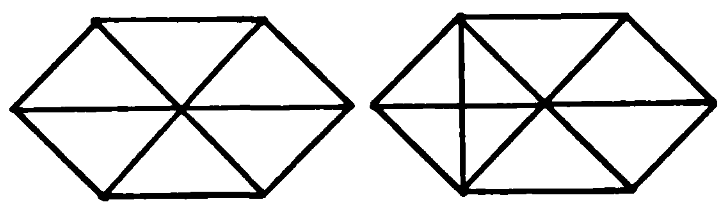


Figure 2.7. Wheel Expansion

2-2.3 The Greedy Heuristic

The idea behind the Greedy Heuristic [Foulds, Gibbons, & Giffin, 1985] is very straight forward. First, all edges are listed so that all edges with A values are first followed by those that have a value of E etc. Next an edge is taken from the top of the list and it becomes the first edge of the subgraph. The edges are then sequentially taken from the top and added to the graph as long as planarity is not violated. When 3n-6 edges have been added the subgraph construction is completed. It is noted that this method requires an explicit planarity test [Hopcroft & Tarjan, 1974].

2-2.4 The N-Boundary Greedy Heuristic

The N-boundary Greedy Heuristic [Giffin & Foulds, 1986] is an extension of the Greedy Heuristic that includes benefits to the final score for not only facilities that are immediately adjacent to one another but for facilities that are k facilities apart from each other. In addition to the normal adjacency matrix required, additional matrices that give values for having two facilities 2, 3, 4, etc. facilities apart are required. Under the assumption of approximately equal areas, normally a score is higher if a facility is fewer facilities distant. Due to this fact when adding a facility the shortest path to reach all other facilities must be calculated in order to find an appropriate addition.

2-2.5 An Oriented Graph Theoretic Heuristic

A paper by Roth, Hashimshony, and Wachman [1982] suggests a method for turning a graph into a rectangular floor plan, again requiring the development of a planar adjacency graph. The adjacencies have no degree of desirability in this method, only a requirement for their presence or absence. These incidence requirements are converted into a planar graph by the subtraction of edges or the addition of dummy vertices. This planar graph is then split into two subgraphs representing north south and east west orientations by a coloring technique and dimensions are calculated using the PERT algorithm. From this technique, several alternative plans are generated for further evaluation. A requirement for the dimension calculations is the orientation of certain facilities to given directions. These calculations use the PERT algorithm to find the critical path from the north side of the building to the south as well as a critical path from the west to the east and thereby determine the necessary dimensions.

2-3 Deltahedron Based Methods

The graph theoretic heuristics above have a major disadvantage compared to the Deltahedron based heuristics that follow. This disadvantage is that as yet there is no systematic method for finding the rectangular geometric dual to the adjacency graphs generated. The main purpose of this thesis is to describe such a systematic approach for the deltahedron based heuristics.

A feature that all of the deltahedron approaches have in common is that they begin with an initial tetrahedron. Short descriptions of the deltahedron approaches follow.

2-3.1 The Deltahedron Heuristic

The Deltahedron Heuristic [Foulds and Robinson, 1978] sequentially adds a vertex into the triangle of the existing graph that will give the greatest increase in adjacency score. This increase in score is calculated by summing the weights of the three edges used to connect the new vertex to the existing graph. The order that the vertices are added is the continuation of the column sum ordering used in the initial tetrahedron selection or the selection at each iteration, of the vertex and triangle that will yield the greatest increase in score among all choices [sometimes referred to as the greedy order]. This method is described in greater detail in chapter 3 since it is used to generate the adjacency graphs used to demonstrate the development of a block plan from a Deltahedron based method.

2-3.2 The Improved Deltahedron Heuristic

The Improved Deltahedron Heuristic [Foulds and Robinson, 1978] uses the solution obtained with the Deltahedron Heuristic as input. This graph is examined to see if any improvements can be made, in the form of edge swapping or vertex relocation. In most cases, if an edge is deleted from the graph, a quadrilateral is formed. The edge that was removed formed a diagonal in this quadrilateral. If the edge that is identified with the other diagonal is added a new graph is formed that is also maximally planar. If the score is increased by this swap, it is performed, and if not, it is ignored. All possibilities are examined and when no improvements can be made, the process stops. In some specific instances after an edge is removed, the one that would be added is already a part of the graph. These situations are either ignored, or a well defined sequence of equivalent swaps made.

2-3.3 The N-Boundary Deltahedron Heuristic

As the N-Boundary Greedy Heuristic is an extension of the Greedy Heuristic, so too is the N-Boundary Deltahedron Heuristic [Giffin & Foulds, 1986] the same type of extension to the Deltahedron Heuristic. An increase to the score of the N-Boundary Deltahedron is determined by the adjacencies of facilities 2, 3, 4, etc.

facilities distant in addition to the immediate adjacencies scored in the Deltahedron Heuristic. This heuristic begins with the same initial tetrahedron selection method as the Deltahedron method and adds to it by choosing the vertex that will yield the highest increase in score for adjacency or near adjacency to all other facilities. As with the N-Boundary Greedy Heuristic, an update version of a shortest path algorithm must be run at every iteration.

2-3.4 The Super Deltahedron Heuristic

The Super Deltahedron Heuristic [Giffin & Foulds, 1985] is fundamentally different from the other graph theoretic methods in that its objective function is not the maximization of total adjacency scores; instead it attempts to minimize transportation costs much like the QAP formulation or the CRAFT method. The method again starts with the initial tetrahedron selection process used in the Deltahedron method since maximizing the proximity of four facilities with high mutual flows should provide reasonably low transportation cost. The order of insertion is either the column sum or the greedy approach used in the Deltahedron method. The triangle selected for insertion is the one that minimizies the sum of the product of the cost per unit distance traveled, the number of trips per time period, and the distance between two facilities, over all pairs of facilities contained in the adjacency graph. The shortest path routine is also required in this method for the computation of pairwise facility distances. The distance traveled between two facilities x and y is approximated by the sum of half the square root of the area of x, the sum of the square root of the area of all facilities on the shortest path from x to y, and half the square root of the area of y. This metric assumes that all facilities are squares with side length equal to the square root of the area, the travel between two facilities is between centroids of the two facilities, and that all travel is done in a rectilinear fashion. These assumptions are not very likely in the final block plan; however, they are only designed to give a ranking among triangles for the insertion process.

CHAPTER 3

RECTANGULAR GEOMETRIC DUAL AND BLOCK PLAN CONSTRUCTION

3-1 Terminology, Notation, and Definitions

The following terminology and notation is defined:

[1] Vertex. A point on the adjacency graph at which edges converge is known as a vertex.

[2] Edge. An edge is a line connecting two vertices on the adjacency graph.

[3] Insertion Order. The insertion order is the order in which the vertices are added to the initial tetrahedron to form the completed adjacency graph.

[4] Rectangular Geometric Dual. A rectangular spacial realization of vertices and their adjacencies represented in the adjacency graph.

[5] Node. Each node is a point in the dual which has a one-to-one correspondance with a triangle formed by three vertices and three edges of the adjacency graph.

[6] Wall. A wall is a line that connects two nodes in the dual. Each wall has a one-to-one correspondance with an edge in the adjacency graph.

[7] Placed in. When a facility i is added to the dual, a portion of the dual is renamed to represent i.

The designation being replaced is called the facility that i was placed in. If another facility j was added so that a portion of facility i is renamed, facility j is placed in i, not the original facility.

[8] Corner. The right angle sometimes required to connect two nodes of the dual in a rectangular fashion is called a corner.

[9] Addition Sequence. The addition sequence is identical to the insertion order, however it refers to additions to the dual not the adjacency graph.

[10] Node Expansion. Node expansion is the redesignation of the structure surrounding a node in the dual when a facility is added at that node.

[11] Inhibitor. An inhibitor is a dummy node added to the dual matrix to prevent the loss of adjacencies when areas are later added to the dual to form the block plan.

[12] N. N is the number of facilities or vertices.

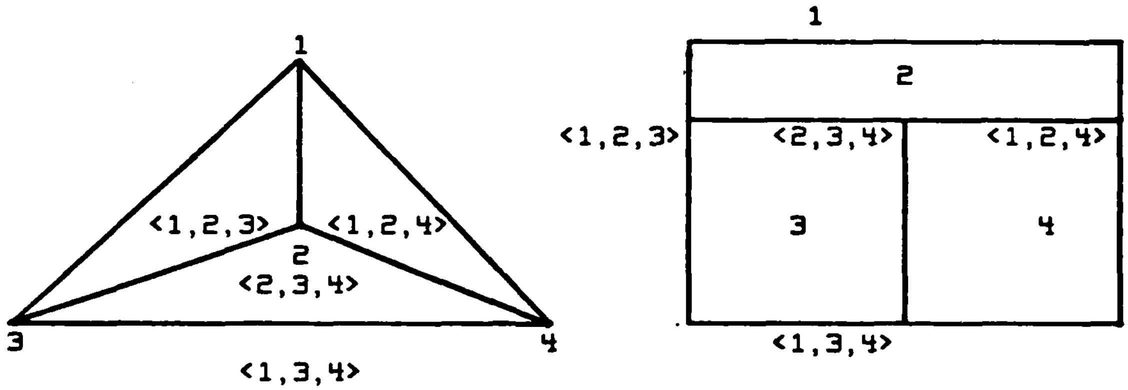
[13] <i,j,k>. The combination of symbols <i,j,k> represent the triangle formed by vertices i, j, and k with edges ij, jk, and ki.

3.2 Deltahedron Method Used

The Deltahedron Heuristic seeks to find a maximally weighted maximally planar adjacency subgraph of a complete adjacency graph. The method used here is the simplest of the variants of the Deltahedron Heuristic. The first step is to construct the NxN matrix of R_{ij} values. The scores for each R_{ij} are entered in the matrix W_{ij}. The columns are then summed and reordered in nonincreasing order by these column sums with the exception of facility 1 which is always the exterior. For ease of discussion, suppose that the vertices were initially in nonincreasing order of column sums and therefore their order is 1, 2, ... ,N. This is now refered to as the Insertion Order. The first four verticies are combined to form the complete graph on four verticies K_4 which comprises the Initial Tetrahedron (see figure 3.1(a)). The vertices are then added according to the insertion order. Consider the insertion of vertex r into triangle <i,j,k>. The benefit to the total score is the sum of $W_{ir} + W_{jr} + W_{kr}$. Therefore r is chosen to maximize this sum over all available triangles. After adding vertex r into triangle <i,j,k>, this triangle <i,j,k> is replaced by triangles <i,j,r>, <i,k,r>, and <j,k,r>. The next vertex is then selected and inserted into the triangle that will achieve the greatest benefit. If there is a tie among several triangles for this maximum benefit, several different strategies can be incorporated. One such strategy is arrange them lexicographically and chose the first one. Another is to chose one of the possible triangles randomly and this approach is taken here to avoid a large concentration of insertions in one section of the graph.

3-3 Description of the Rectangular Geometric Dual Construction

The method used for constructing the rectangular geometric dual, hereafter referred to as the dual, is limited to the class of adjacency graphs that can be constructed using any variant of the deltahedron heuristic. The only input required is the triangle insertion order. The process begins with a rectangular representation of the dual corresponding to the initial tetrahedron. This is shown in figure 3.1. The facilities are numbered as shown with facility 1 being defined as the exterior. It should be noted that each node of the rectangular geometric dual has three and only three edges incident with it. Each node has a one to one correspondence with a triangle that exists in the deltahedron at the current stage of the adjacency graph construction. If a facility is added to the rectangular geometric dual by expanding about one of these nodes, its adjacencies will correspond exactly to those in the adjacency graph.



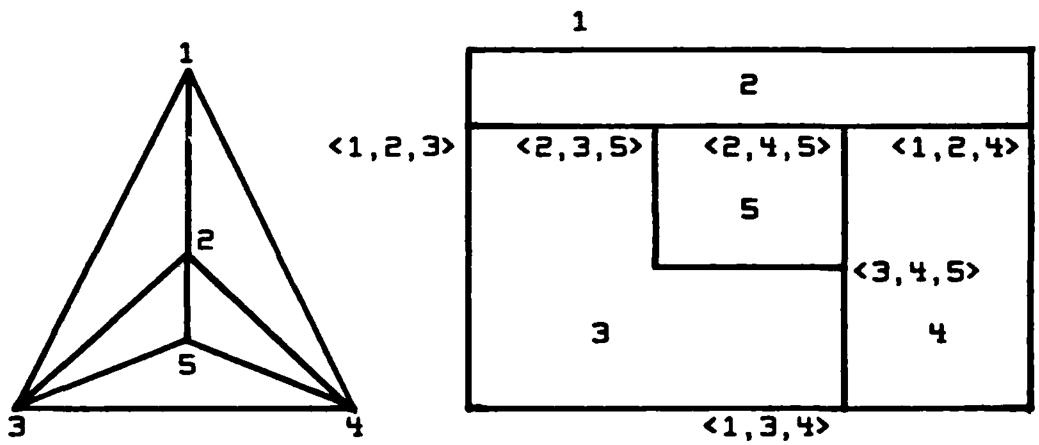
Adjacency Graph [a]　　Rectangular Geometric Dual [b]

Figure 3.1. Initial Tetrahedron

There are two ways that a facility may be added to the dual with the decision being made by inspection of the nodes in the dual that are only one edge distant. If there are no corners that are between the node of interest and any of the three adjacent nodes, then the facility is added by a BOX operation. If there is a corner immediately adjacent to the node of interest, a CARVE operation is used. An example of each follows.

3-3.1 Boxing

From inspection of the initial block plan, figure 3.1[b], it can be seen that the only node that has no corners adjacent to it is <2,3,4> therefore consider the insertion of facility 5 at this node. From the adjacency graph in figure 3.2[a], it can be seen that when facility 5 is inserted into <2,3,4>, the triangle <2,3,4> is replaced by three triangles, namely <2,3,5>, <2,4,5>, and <3,4,5>. Figure 3.2[b] illustrates this insertion and the necessary relabeling.



[a]
Adjacency Graph

[b]
Rectangular Geometric Dual

Figure 3.2. Insertion of facility 5 into triangle <2,3,4> [BOX]

Since facility 5 replaced a portion of facility 3, this is defined as placing facility 5 in facility 3. This operation is called a "box" for obvious reasons. The box could also be flipped to the opposite side of the wall separating facilities 3 and 4. The choice is arbitrary, however it does affect the orientation of the block plan from the decision point on. For any given location this flipping is not always possible for other reasons that will be shown later. Four orientations of the boxing operation are possible and for implementation purposes are defined as left-down, left-up, right-down, and right-up (see figure 3.3).

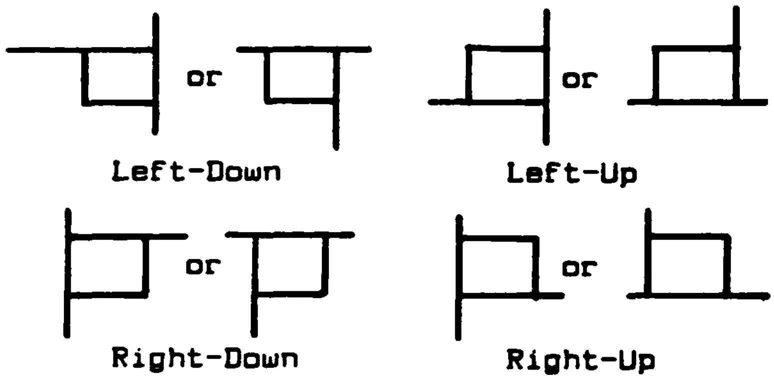


Figure 3.3. Possible Boxing Alternatives

3-3.2 Carving

Now consider instead, the insertion of facility 5 into triangle <1,2,3>. This could be done as a boxing operation (right and down) however this would unnecessarily create an "L" shape which is not as desirable as a rectangle. This is avoided by an operation called a "carve." Figure 3.4 is an illustration of this operation.

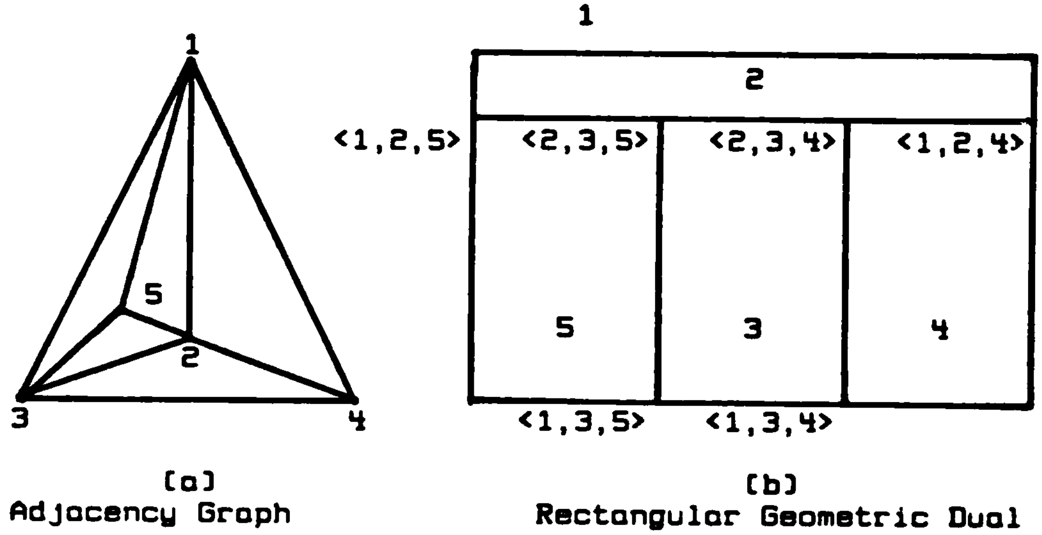


Figure 3.4. Insertion of facility 5 into triangle <1,2,3> [CARVE]

The same general triangle replacement is done as above. The eight orientations for the carve operation are shown in figure 3.5 along with their designations. These designations indicate first the direction in which the corner is encountered followed by the direction not cut off by the corner. A carve operation cannot be flipped to the opposite side of a wall like the box since there is no corner to "carve" towards. Boxing might be an alternative; however, as will be shown later there could be a problem with maintaining the required adjacencies in the dual when areas are introduced.

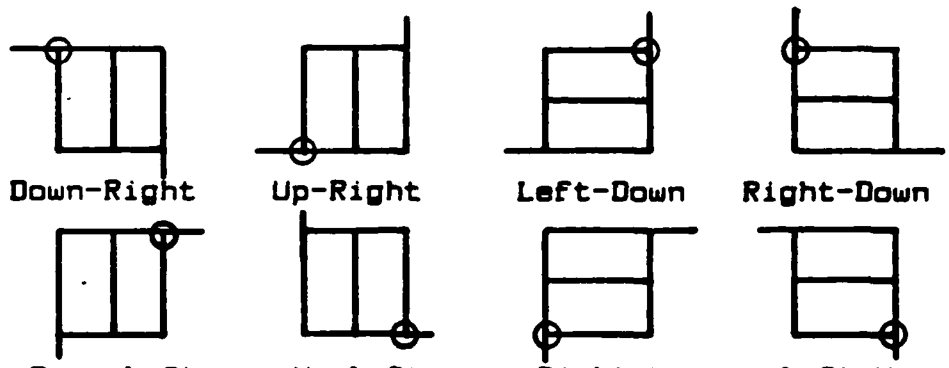


Figure 3.5. Possible Carving Alternatives

Using these two operations, the entire dual is constructed by adding each facility to the existing dual using the same sequence used when inserting the triangles in the adjacency graph. After the dual is completed, the block plan is made by incorporating the areas of the individual facilities into the orientation developed during the dual construction.

3.4 Data Structure and Computer Implementation -- DELTAPLAN

The computer program for this method is called DELTAPLAN and was written in BASICA on an IBM Personal Computer. Due to the amount of memory available in BASICA, the problem size is somewhat limited however; 11 facility problems can be handled routinely and in some cases it will run completely with as many as 22 facilities.

3-4.1 Initialization

To facilitate an easily envisioned and manipulated representation of the dual, a matrix of alphanumeric strings is generated that contains the elements common to all initial block plans. As can be seen in figure 3.6, all of the initial triangles are represented as six character strings. For example triangle <1,2,3> is represented by 010203. The walls are represented by a single dash "-" and the interior of a facility by a two digit numerical string for example "04" for facility 4 and "12" for facility 12. Since each corner is adjacent to only two facilities the first two elements of the string are letters that represent the orientation of the corner [see figure 3.7.] The two corners in facility 2 [upper left and upper right corners] are not used as no facilities are added within facility 2 and therefore are represented by "000102".

01	01	01	01	01	01	01	01	01	01	01
01	000102	-	-	-	-	-	-	-	000102	01
01	-	02	02	02	02	02	02	02	-	01
01	010203	-	-	-	020304	000000	-	-	010204	01
01	-	03	03	03	-	04	04	04	-	01
01	-	03	03	03	-	04	04	04	-	01
01	-	03	03	03	-	04	04	04	-	01
01	-	03	03	03	-	04	04	04	-	01
01	AA0103	-	-	-	010304	000000	-	-	BB0104	01
01	01	01	01	01	01	01	01	01	01	01

Figure 3.6. Matrix representation of the rectangular geometric dual

Figure 3.7. Corner Labels

With the exception of two, all of the elements listed above have a direct counterpart in the dual shown in figure 3.1(b). These exceptions are called "inhibitors" and their purpose will be defined later.

3-4.2 Addition of Facilities to the Rectangular Geometric Dual

Upon completion of the deltahedron heuristic, for simplicity all facilities are relabelled according to their position in the insertion order. Hence, we assume facility (i+4) is added to the dual at the (i)th stage and that facilities 1 through 4 make up the initial tetrahedron. As can be seen from figure 3.1(a), only four options exist for the placement of this first facility and the output of the deltahedron heuristic used to generate the insertion order has chosen the appropriate one. A search is then made to match the triangle in which facility 5 is to be inserted, with its identical element in the dual matrix. A sort routine is included in the program to insure consistent ordering of the three two digit pairs within each element. Since a search of the whole matrix is rather time-consuming, a table is constructed which contains each possible insertion triangle along with its coordinates [I,J] in the matrix.

(1) Searching. Before the search is done, all flags (described below) and all direction indicators are set to zero. Starting at the coordinates [I,J], a search is performed to the left to identify the structure of the dual to the left of the triangle in question. A variable "L" is used to keep track of the search and is initially equal to J. L is decremented by one and the element with coordinates [I,L] is examined. If L is less than 1, the border of the matrix has been reached and the left direction is "unusable." An unusable direction means that no box or carve operation is possible in this direction. In the program this is accomplished by setting LFLAGO=1. If the element is a dash, "-", the search continues with the next element. If a six digit

element is found, the search stops. If the first digit of the element is "A" or "D" [these are the only possible corners when searching to the left], LFLAG1=1 or 4 respectively. This flag indicates whether a box or a carve operation is appropriate where a type A corner is indicated with a 1, type B with a 2, type C with a 3, and type D with a 4. The presence of a "000000" element indicates a inhibitor and the inhibitor flag LFLAG2 is set to 1 [inhibitors are described later in this chapter.] If L=J-1 or J-2, the left direction is again unusable since there are not enough elements between J and L to define a new facility. After the search to the left, a similar search is done in the right, down, and up directions.

[2] Carve/Box Decision. The flags LFLAG0, LFLAG1, RFLAG0, RFLAG1, DFALG0, DFLAG1, UFLAG0, and UFLAG1 are compared to the set of values required for each carve operation to see if it is possible to carve. For each carve operation three flags must be set to specific values. For example, to carve left-up the corner encountered in the left search must be a type A [LFLAG1=1], the left direction must be usable [LFLAG0=0] and the up direction must be usable [UFLAG0=0]. If none of the above conditions are satisfied, the flags required for the boxing operations are checked. In this case there are only two flags required for a box operation.

For the box left-up operation the flags needed are the same as for the carve left-up [LFLAG0=0 and UFLAG0=0] except there must be no corner present so the left and up corner indicators must be 0 [LFLAG1=0 and UFLAG1=0].

[3] Carving. The left-up carving operation will be used here for description purposes. However, the same general format applies to all eight carving operations. Consider the insertion of facility 5 into triangle <1,3,4>. An inspection of figure 3.6 gives the structure surrounding 010304 and indicates that a left-up carve is appropriate. The coordinates [I,J] of 010304 are determined and will become the location of one of the new nodes of facility 5. In this case L equals the j coordinate of AA0102, U equals the i coordinate of 020304 and both the right and down directions are unusable. Next, the coordinates [I1,J1] of the point diagonally across the new facility from [I,J] are determined. If the element which determines U is not an inhibitor, I1 is half way between I and U. If it is, I1=U+1, since if an inhibitor is present, the element above has an unusable down direction. A carve that goes only half way up wastes the entire portion above the carve and is then lost to further insertions. However, if the carve goes as close as possible to the node above, only a few elements in the matrix are lost. The J coordinate J1 is

equal to L. In order to determine the orientation of the facilities which border the new one, three more variables are set. In this case they are L$="01", U$="03", and R$="04", and they are taken from the matrix by determining which facilities are to the left, right and above the new facility. These three pairs along with the number of the new facility [FAC$] are combined to form the four new nodes in the matrix. The upper right node is U$ + R$ + FAC$ [030405] with coordinates [I1,J], while the upper left node is L$ + U$ + FAC$ [010305] at [I1,J1]. The lower left element is "AA" + L$ + FAC$ [AA0105] at [I,J1] and finally the lower left node is L$ + R$ + FAC$ [010405] at [I,J]. The walls are then inserted by renaming the elements between each node on the perimeter of the new facility with "-". The interior of the facility is then filled in with FAC$ or in our case "05". Two inhibitors are then added in place of the elements immediately above the upper left and upper right nodes. The purpose of these is described later. Figure 3.8 shows the matrix with facility 5 added at <1,3,4>.

01	01	01	01	01	01	01	01	01	01	01
01	000102	-	-	-	-	-	-	-	000102	01
01	-	02	02	02	02	02	02	02	-	01
01	010203	-	-	-	020304	000000	-	-	010204	01
01	-	03	03	03	-	04	04	04	-	01
01	-	03	03	03	-	04	04	04	-	01
01	-	03	03	03	-	04	04	04	-	01
01	000000	03	03	03	000000	04	04	04	-	01
01	010305	-	-	-	030405	04	04	04	-	01
01	-	05	05	05	-	04	04	04	-	01
01	-	05	05	05	-	04	04	04	-	01
01	AA0105	-	-	-	010405	000000	-	-	BB0104	01
01	01	01	01	01	01	01	01	01	01	01

Figure 3.8. Matrix representation with facility 5 added at <1,3,4>

Two additional items are required for the area calculations that begin following the completion of the dual. The first of these is the operation with which the facility was added. In the above example, the operation is carve left up therefore the variable OPER$[5] [operation for facility 5] is designated "CLU". The other requirement for the area calculations is the number of the facility in which the new facility was placed. The variable for this is PLIN$, and its value in the above example is 3 since the 05 elements replaced 03 elements.

<u>[4] Boxing</u>. The box operation is accomplished in much the same manner as the carve. For this description, the addition of facility 5 at <2,3,4> will be used. The surrounding structure here indicates that a box left down operation is appropriate. Notice that without the inhibitor to the right of 020304 a box right down would

also be possibility. As noted earlier, this topic will be discussed later. As in the carve operation, the coordinates [I,J] of 020304 are determined, as well as L and D, in this case, L is the j coordinate of 010203 and D is the i coordinate of 010304. Since neither of these is an inhibitor, I1 is half way between I and D and J1 is half way between J and L. If the node to the left had been an inhibitor, J1 would have been L+1 and if the node below was and inhibitor, I1 would have been D-1. The same matrix conservation reasoning applies here as in the carve operation. The variables L$, U$, and R$ are set as described above in order to define the new nodes. Here L$="03", U$="02", R$="04", and FAC$ is again "05". The new nodes are 020305 for the upper left, 020405 for the upper right, 030405 for the lower right, and AA0305 for the lower left element. As before, the walls are inserted, interior of the new facility is relabelled, OPER$[5] is set to its value of BLU, and PLIN$[5] is set to its appropriate value which is 03. A representation of this is given in figure 3.9. It is noted here that as above there are two inhibitors, one to the left of the upper left node and one below the lower right node. The purpose of the inhibitor is defined next.

01	01	01	01	01	01	01	01	01	01	01	01	01
01	000102	-	-	-	-	-	-	-	-	-	000102	01
01	-	02	02	02	02	02	02	02	02	02	-	01
01	010203	-	000000	020305	-	-	020405	000000	-	-	010204	01
01	-	03	03	-	05	05	-	04	04	04	-	01
01	-	03	03	-	05	05	-	04	04	04	-	01
01	-	03	03	AA0305	-	-	030405	04	04	04	-	01
01	-	03	03	03	03	03	000000	04	04	04	-	01
01	-	03	03	03	03	03	-	04	04	04	-	01
01	-	03	03	03	03	03	-	04	04	04	-	01
01	AA0103	-	-	-	-	-	010304	-	-	-	BB0104	01
01	01	01	01	01	01	01	01	01	01	01	01	01

Figure 3.9. Matrix representation with facility 5 added at <2,3,4>

[5] Inhibitors. The purpose of inhibitors is to block the insertion of facilities at certain locations that could possibly destroy an existing adjacency once areas are added. Consider the addition of facility 5 to <2,3,4> and the subsequent addition of facility 6 to <2,4,5>. If facility 5 were added as described above, it is noted that the coordinates of 020405 are the same as were the coordinates of 020304. With the inhibitor present, as is shown in figure 3.9, the only possible operation is a box left down. However, if the inhibitor were not present, a right down box would also be possible. If the box left down for facility 5 were followed by a box right down for facility 6, the result would be as is shown in figure 3.10(a). The problem arises when areas are introduced. If the area of facility 6 is larger than that of facility 5, the adjacency between facilities 4 and 5 is lost and an adjacency between 3 and 6 is gained as is shown in figure

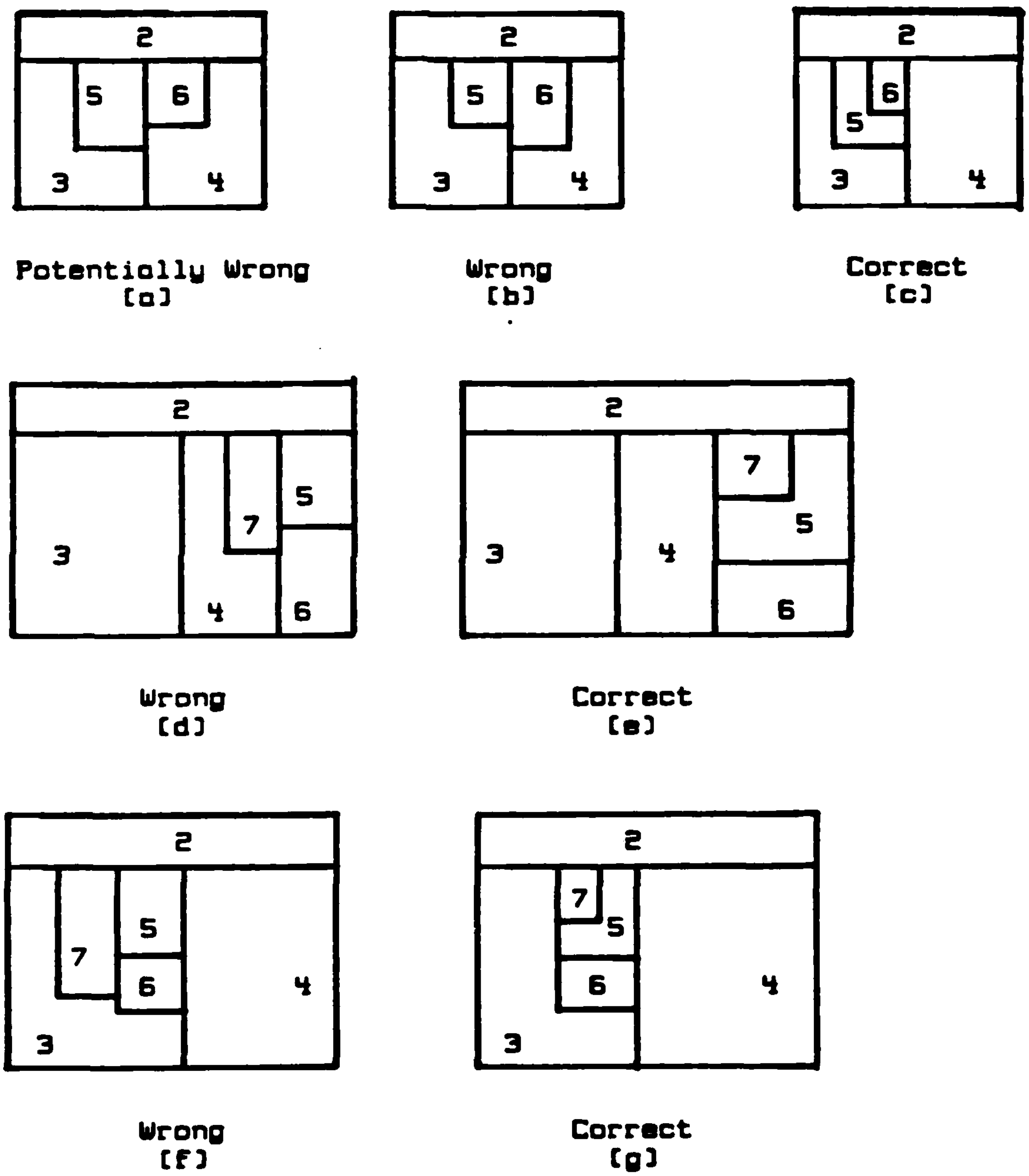


Figure 3.10. Correct boxing technique to prevent losing adjacencies

3.10[b]. In this case the block plan would not reflect the adjacencies required by the adjacency graph. The block plan that does reflect the required adjacencies regardless of areas is shown in figure 3.10[c].

Another example of inhibitors using the carve operation is illustrated in figures 3.10 [d] and [e]. Here a carve for facility 5 at 010204 is followed by a carve at 010405 for facility 6. With no inhibitors, the problem here is the addition of facility 7 at 020405 and the two options of box left down and box right down. As is seen in figure 3.10[d] the box left down destroys the adjacency between 4 and 5 and creates an adjacency between 6 and 7; however, at this stage facility 7 should only be adjacent to 2, 4, and 5. The box right down is appropriate here and figure 3.10[e] illustrates the block plan which the inhibitors require.

A final example is shown in figures 3.10 [f] and [g]. In this case facility 5 is added at 020304 followed by a carve for facility 6 at 030405. When facility 7 is added at 020305, the same problem presented in figure 3.11 arises again. With no inhibitors the block plan could end up as in figure 3.10[f], whereas inhibitors require the block plan in figure 3.10[g].

The initial choice of location for the inhibitors to the right of 020304 and 010304 is arbitrary. Placement of both on the left would perform just as well but it should be noted that they must both be on the same side or they would create the very problems they are designed to eliminate.

The results below follow from the operations as defined.

[6] _Theorem_1_. No more than one carve can be done within any facility. PROOF -- In order to carve there must be a corner towards which one carves. After one carve is done, there is no corner left in the original facility therefore the condition required to carve does not exist and no further carving can be done.

[7] _Theorem_2_. No more than three facilities may be placed within any given facility i. PROOF -- All facilities, with the exception of 2, begin as boxes. Even if a facility is added by a carve operation it contains one corner and therefore has the same structure as a box. As such, there are three nodes which can be expanded about to form new facilities. Each time a facility is added, due to the nature of the inhibitors, none of the new nodes created allow the addition of a facility within facility i. An illustration of this is given in Figure 3.11.

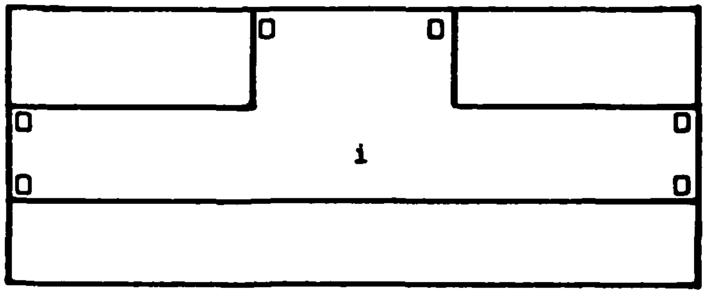


Figure 3.11. Location of inhibitors when no facilities may be added

[8] Corollary 2.1. If three facilities are added within facility i, two must be boxes and one a carve.

PROOF -- For a given node, if there is an opportunity to carve it will be done first. From theorem 1, one cannot carve again therefore the other facilities must be added by a box operation.

From Corollary 2.1, the worst shape a facility may have is a "T".

3-4.3 Creating The Block Plan

The block plan is nothing more than addition of areas to the dual. To accomplish this it is easiest to start with a "clean slate" rather than trying to adjust the existing dual. The inputs required for each facility i in this phase are the operation [OPER$[i]], the facility that it was placed in [PLIN$[i]], and the area [AREA[i]]. Each facility in the block plan is given by its coordinates within a square with sides of length one and where the coordinates represent percentages of the actual wall lengths. For example, consider two buildings each containing 10,000 square feet, with dimensions 100x100 for the first and 125x80 for the second (see figure 3.12.) A facility with dimensions [0,0], [0,0.5],[0.5,0], and [0.5,0.5] would have dimensions of 50x50 in the first case and 62.5x40 in the second however as one can see the areas are both equal to 2,500 sq. ft. This adds more flexibility to the actual site block plan since no restriction is made that the building be square.

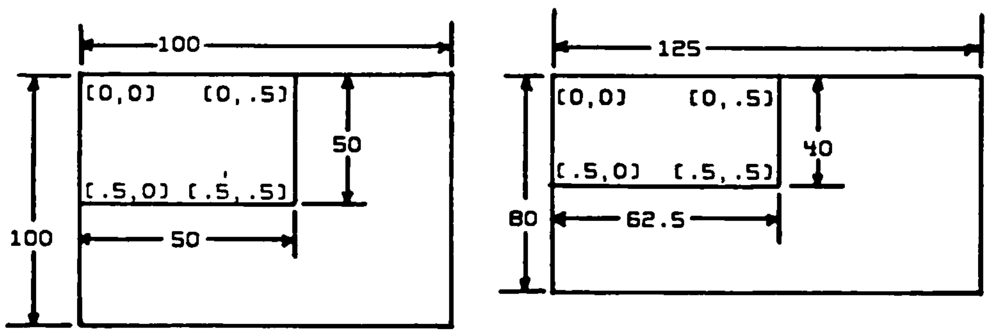


Figure 3.12. Coordinate/Area Relationship

(1) Computing the Initial Area Required for each Facility. The area required for a facility i when it is initially added into the block plan is not the area of facility i alone since subsequent facilities are added within the initial boundaries of facility i. The initial facility should contain the area required for all of the

facilities added within its initial boundaries at later stages. Using the PLIN vector, a cumulative area vector called AREAIN is calculated so that each value of AREAIN[i] is equal to the area of facility i plus the cumulative areas of all facilities subsequently added within the initial boundaries of facility i.

[2] Carving and The Addition of the First Two Facilities to the Block Plan. The entire square is defined as the initial boundary of facility 2, therefore its cumulative area [AREAIN[2]] is equal to the total area or AREATOT. Facility 3 is then placed within the initial boundary of facility 2. Since the initial facility 3 contains all facilities except 2 it can be viewed as a carve up from below. It is noted that both the carve left up and the carve right up look the same with the only difference being the node from which the carve took place. In the initial dual section this was an important distinction, however for the block plan it doesn't really matter since the shape for the block plan is all we are concerned with here [see figure 3.5.] Therefore in the block plan section only four carve routines are required since the left-up and right-up, the left-down and right-down, the down-right and up-right, and the down-left and up-left are equivalent. The carve operation at this stage involves basically cutting the

initial area of 2 into two parts that have the proper ratio of areas. Since the coordinates are in percentages of distance, the carve operation may be accomplished by simply relabeling the lower coordinates of facility 2 as the lower coordinates of the initial area of facility 3, redefining the lower two coordinates of facility 2 according to the ratio of cumulative areas, and also assigning these coordinates as the upper coordinates of the initial area of facility 3. The cumulative area of facility 3 [AREAIN[3]] is then subtracted from the cumulative area of 2 [AREAIN[2]] to get the new cumulative area of facility 2. The same type of operation is done for adding the initial area of facility 4 within facility 3 but a carve to the to left is used.

Up to this point there have been no problem specific facilities placed as facilities 2 through 4 always have the same initial location. From here on, the facilities are not necessarily added in the same sequence as they were in the insertion order; instead they are added according to the facility that they are placed in. For example, all facilities whose PLIN value is 3 are added to facility 3, then those with PLIN values of 4, etc. From Theorem 2 and its Corollary, at most three facilities may be placed in facility 1 and they must be a subset of two boxes and a carve. The PLIN vector is searched to find the three facilities, if they exist, that are placed in facility i. If a carve operation is present, it is done first. The carve method described above for the initialization of facilities 2 through 4 is used for subsequent carve additions.

[3] Box Additions to the Block Plan. When there are two boxes to be added to the block plan, the one with the largest cumulative area is chosen to be inserted first. Consider the addition of facility 5 at <2,3,4> within facility 3 as described above [see figure 3.2.] The upper right coordinates of facility 3 are relabeled as the upper right coordinates of facility 5. The lower right and upper left coordinates are calculated according to the square root of the ratio of cumulative areas. The lower left coordinate is the i coordinate of the lower right and the j coordinate of the upper left. The only change to the existing facility [3] is relabeling of the upper right coordinate which is the same as the lower left of the new facility.

The addition of a second box is done in the same manner as the first so long as there is sufficient space. If there is not a "correction" routine is entered. The definition of "sufficient space" is as follows. After one box has been added, an L shape exists. The coordinates for the rectangular portion of this existing L shape where the new box is to be added are used to determine the "effective" area of the existing facility. If the area of the box to be added is more than 95% of this effective area, there is not sufficient space. If this is the case, wall length of the first box in the offending direction is reduced with the adjacent wall being increased to maintain the specified area. When sufficient space is achieved, the second box is added along with the corrected first box. As with the dual construction, these operations are used repeatedly until the block plan is completed.

CHAPTER 4

EXAMPLE PROBLEMS

In this chapter DELTAPLAN solutions to three different problems are presented. The first example is a problem from Francis and White [1974] and comparisons with ALDEP and CORELAP solutions are given. The second example is also from Francis and White, and it includes the illustration of a possible extension to include changes to the adjacency graph made by the Improved Deltahedron Heuristic. The final example is a problem that is too large to be solved by the current version of DELTAPLAN, however a brief description of the variable reassignment required to construct the complete block plan is included.

4-1 Example I

The first example is a ten facility problem however, since the Deltahedron method requires the exterior to be included as facility 1, the problem shown has 11 facilities. The REL chart required as input by the Deltahedron method is given in figure 4.1.

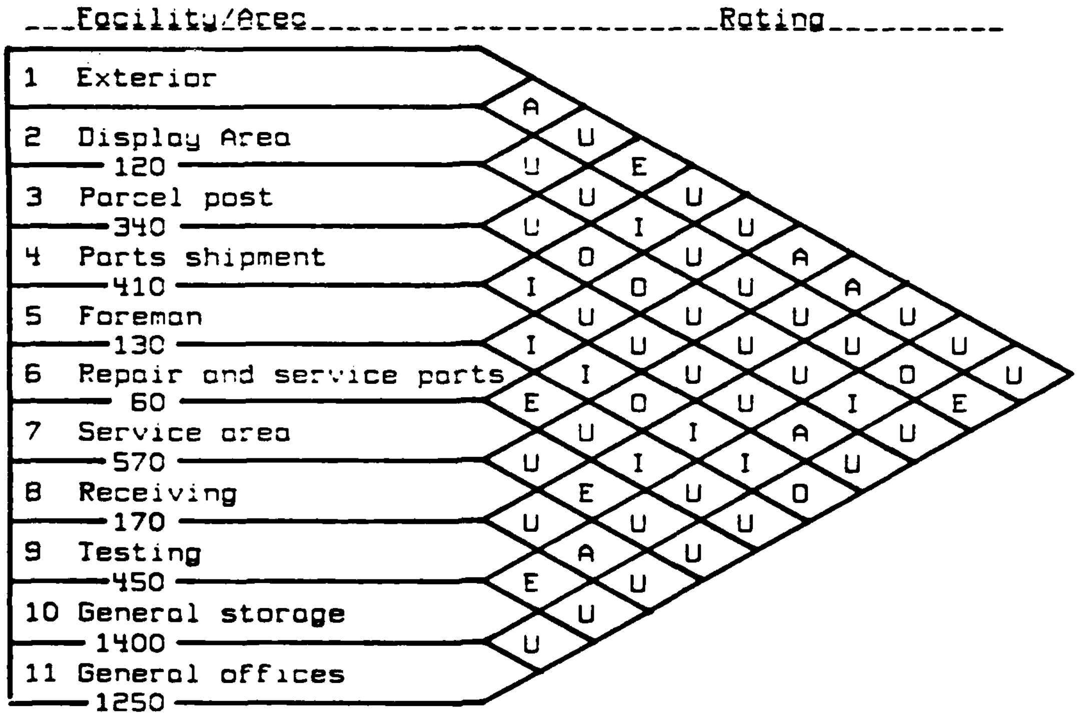


Figure 4.1. Example I REL Chart

The insertion order calculated using column sums is:

1 10 8 7 2 4 9 5 6 11 3

From the insertion order it can be seen that the initial tetrahedron is 1-10-8-7 and table 4.1 gives the remaining vertices and the triangles into which they were inserted.

Table 4.1 Example I Vertices and Insertion Triangles

Vertex	Triangle
2	< 1 8 7>
4	< 1 10 7>
9	<10 8 7>
5	<10 7 9>
6	< 5 7 9>
11	< 2 8 7>
3	<10 7 5>

Using the insertion order and triangle choices from the Deltahedron method, the DELTAPLAN procedure constructs the dual as illustrated in figure 4.2.

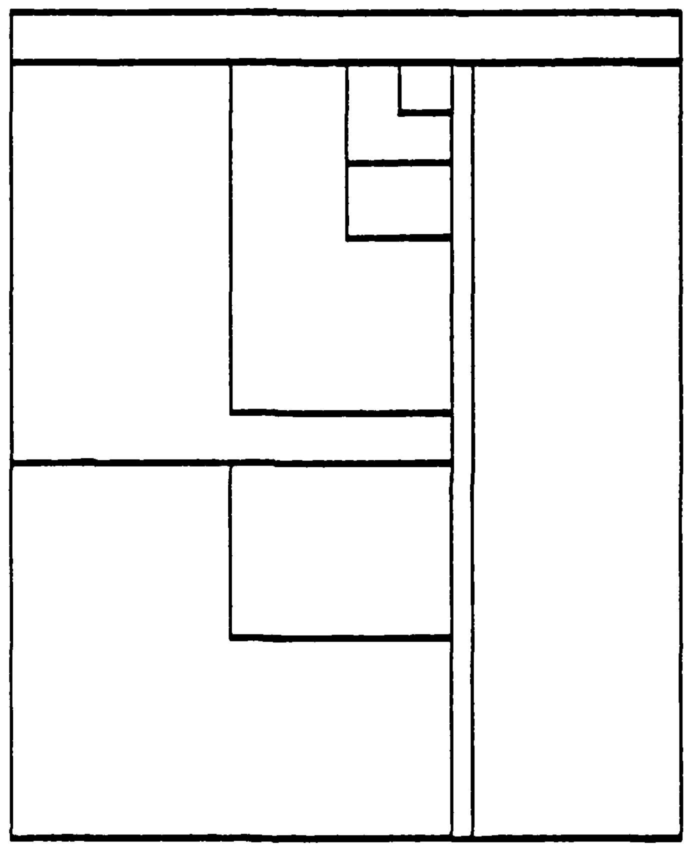


Figure 4.2. Example I Dual

The resulting block plan [rectangular geometric dual with areas] is shown in figure 4.3.

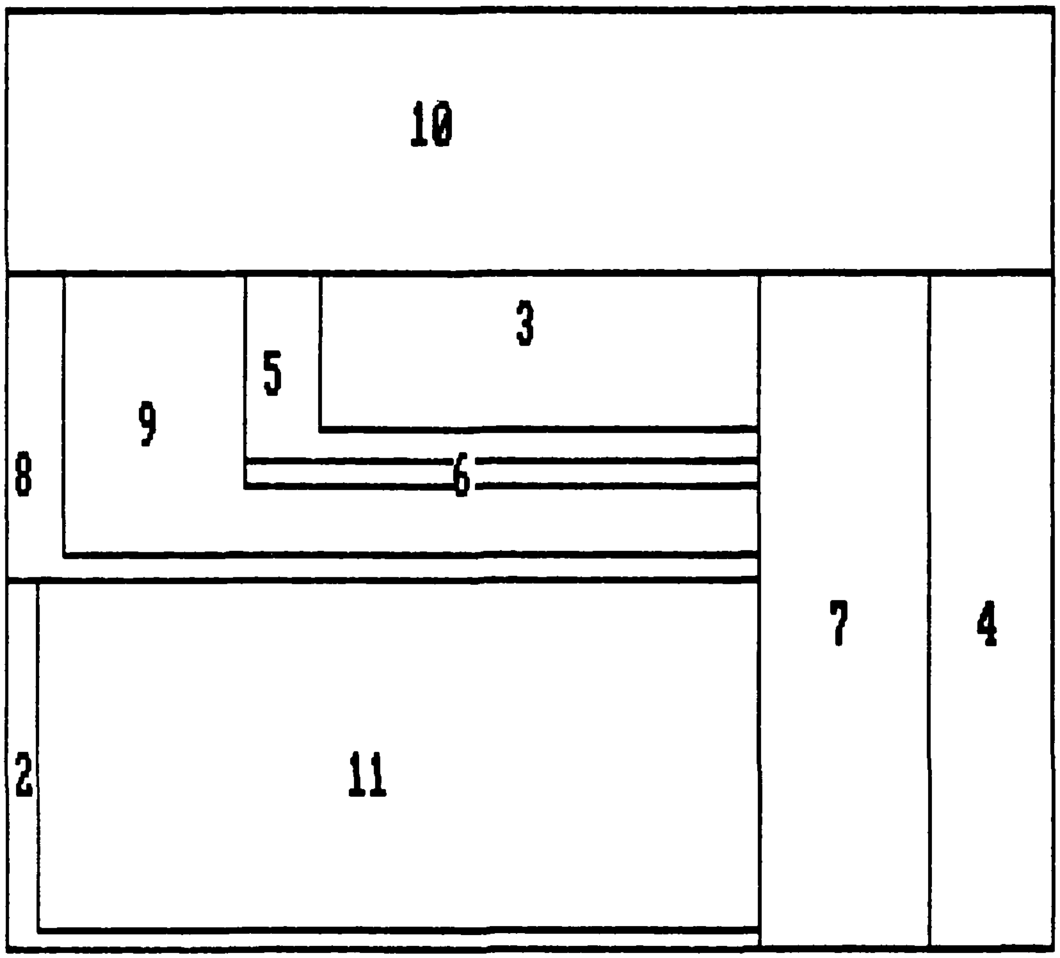


Figure 4.3. Example I Block Plan

The complete actual output from this example is given in the appendix. In addition to the output given here, the appendix includes the incidence matrix, a condensed version of the A$ matrix, the insertion order information, and the coordinates of the block plan. The incidence matrix is a duplicate of the original REL chart with the adjacencies not present in the adjacency graph replaced by dashes. The condensed A$ matrix uses numbers to represent the interior of facilities, dashes to represent the walls [including intersections], and O's to represent the inhibitors. The first line of the insertion order information gives the second, third, and fourth facilities inserted, and their areas. Each additional line gives the facility number, the area, the operation used to insert the facility in the dual, the triangle it was placed in [relabeled to correspond to the order of insertion], and the facility that the new facility was placed in [also relabeled]. The coordinates listed are in the same relative position on the page as in the block plan i.e. the upper left coordinate of each group of four is the coordinate of the upper left corner of the facility. In the case where a box has been placed in a facility and there are now six corners in the facility, the coordinate of the corner where the box was placed is the coordinate of the box that protrudes into the old facility [see figure 4.4].

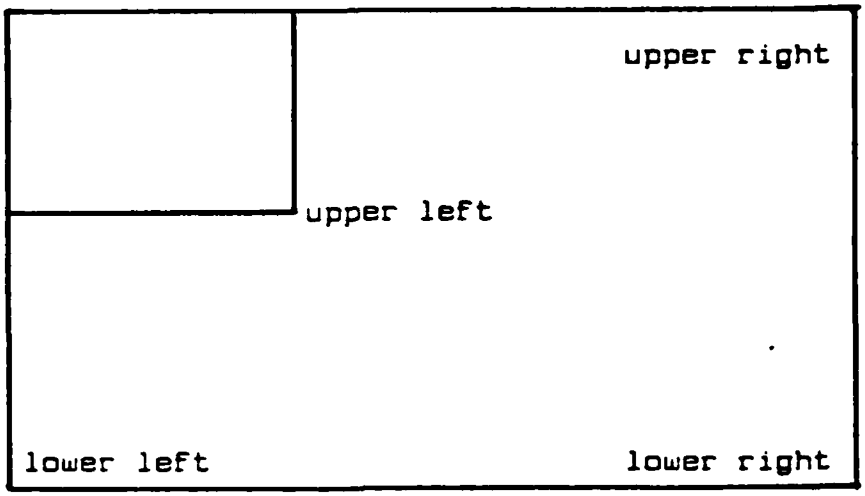


Figure 4.4. Coordinate location when a Box is placed within the facility

Figures 4.5 and 4.6 show the output from ALDEP and CORELAP for the same problem. For comparison, the scores for each are calculated using the scoring rules of the Deltahedron method. This is justified since the scoring for the ALDEP method is identical [this is true in this case since there are no facilities adjacent diagonally] and CORELAP includes maximization of adjacencies in its objective function. Scores for adjacencies to the exterior are not included since the input for ALDEP and CORELAP solutions did not include these adjacencies in their REL charts, therefore the scores for adjacency with the exterior are subtracted off the Deltahedron score.

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	8	8	10	10	10	10	7	7	7	7	11	11	11	11	0	0	1
1	8	8	10	10	10	10	7	7	7	7	11	11	11	11	0	0	1
1	8	8	10	10	10	10	7	7	7	7	11	11	11	11	0	0	1
1	8	8	10	10	10	10	7	7	7	7	11	11	11	11	0	0	1
1	8	8	10	10	10	10	7	7	7	7	11	11	11	11	0	0	1
1	8	8	10	10	10	10	7	7	7	7	11	11	11	11	0	0	1
1	8	8	10	10	10	10	7	9	7	7	11	11	11	11	0	0	1
1	8	8	10	10	10	10	9	9	7	7	11	11	11	11	0	0	1
1	8	10	10	10	10	10	9	9	7	7	11	11	11	11	0	0	1
1	10	10	10	10	10	10	9	9	7	7	2	11	11	11	0	0	1
1	10	10	10	10	10	10	9	9	7	7	2	2	11	11	0	0	1
1	10	10	10	10	10	10	9	9	7	7	2	2	11	11	11	11	1
1	10	10	10	10	10	10	9	9	7	7	2	2	11	11	11	11	1
1	10	10	10	10	10	10	9	9	7	7	2	2	11	11	11	11	1
1	10	10	10	10	10	4	9	9	7	7	2	2	11	11	11	11	1
1	10	10	10	10	4	4	9	9	7	7	3	2	11	11	11	11	1
1	10	10	10	10	4	4	9	9	7	7	3	3	11	11	11	11	1
1	10	10	10	10	4	4	9	9	7	7	3	3	11	11	11	11	1
1	10	10	10	10	4	4	9	9	7	7	3	3	11	11	11	11	1
1	10	10	10	10	4	4	9	9	7	7	3	3	11	11	11	11	1
1	10	10	10	10	4	4	9	9	7	7	3	3	11	11	11	11	1
1	10	10	10	10	4	4	9	9	7	7	3	3	11	11	11	11	1
1	10	10	10	10	4	4	9	9	6	6	3	3	11	11	11	11	1
1	10	10	10	10	4	4	9	9	6	6	3	3	11	11	11	11	1
1	10	10	10	10	4	4	9	9	6	6	3	3	11	11	11	11	1
1	10	10	10	10	4	4	9	9	5	5	3	3	11	11	11	11	1
1	10	10	10	10	4	4	9	9	5	5	3	3	11	11	11	11	1
1	10	10	10	10	4	4	9	9	5	5	3	3	11	11	11	11	1
1	10	10	10	10	4	4	9	9	5	5	3	3	11	11	11	11	1
1	10	10	10	10	4	4	4	4	5	5	3	3	11	11	11	11	1
1	10	10	10	10	4	4	4	4	5	5	3	3	11	11	11	11	1
1	10	10	10	10	4	4	4	4	5	3	3	3	11	11	11	11	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 4.5. ALDEP Layout for Example I

1	1	1	1	1	1	1	1
1	1	9	4	4	1	1	1
1	9	9	4	4	8	8	1
1	9	9	10	10	10	10	1
1	7	7	10	10	10	10	1
1	7	7	10	10	10	10	1
1	7	7	10	10	10	10	1
1	6	5	3	3	3	1	1
1	1	2	11	11	11	11	1
1	1	1	11	11	11	11	1
1	1	1	11	11	11	11	1
1	1	1	11	1	1	1	1
1	1	1	1	1	1	1	1

Figure 4.6. CORELAP Layout for Example I

Comparison shows that the Deltahedron method achieved the highest score with 217 followed by ALDEP with 211 and finally CORELAP with 210. It is noted that there are several narrow L shaped facilities in the DELTAPLAN block plan however some modifications described in the next example and in chapter 5 might help to create more rectangular or regular spaces.

4.2 Example II

The second example is also from Francis and White and as with example I, an exterior facility has been added resulting in a 12 facility problem. The REL chart for this problem has been rearranged so that the insertion order is simply increasing integers from 1 to 12. Figure 4.7 gives the REL chart used for input and table 4.2 gives the insertion vertices and triangles.

Since the REL chart has been rearranged, the initial tetrahdron is 1-2-3-4.

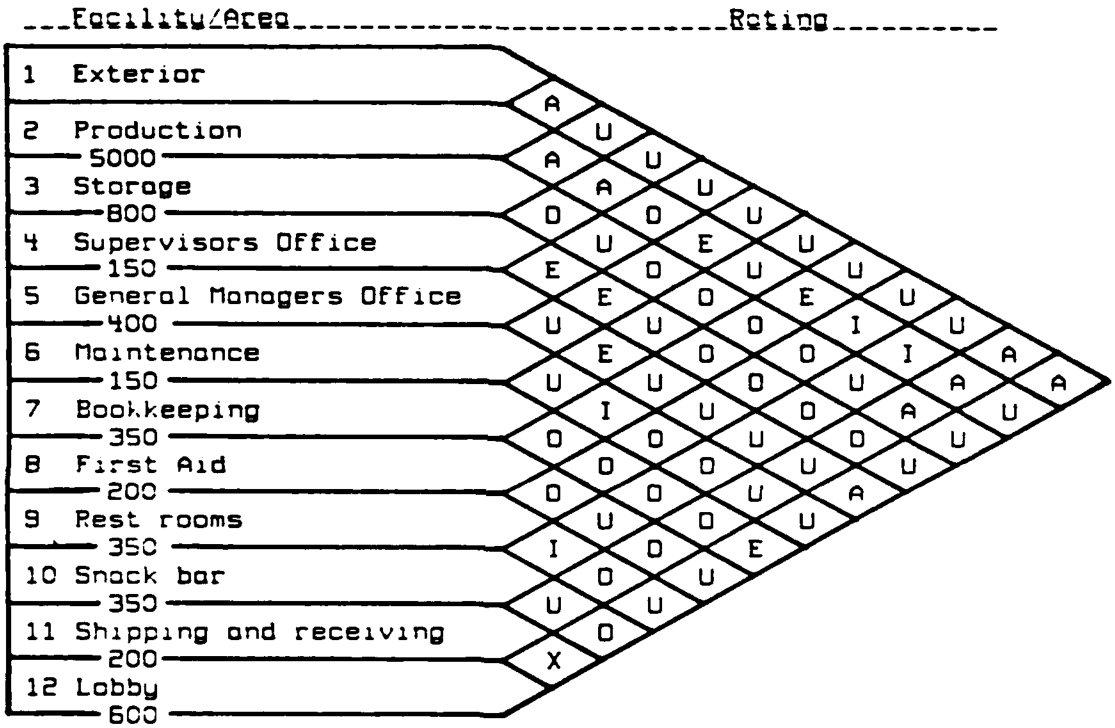


Figure 4.7. Example II REL Chart

Table 4.2 Example II Vertices and Insertion Triangles

Vertex	Triangle
5	<1 2 4>
6	<2 3 4>
7	<5 2 4>
8	<2 3 6>
9	<2 3 8>
10	<2 3 9>
11	<1 2 3>
12	<1 4 5>

As with example I, the complete computer output is given in the appendix. Figures 4.8 and 4.9 show the dual and the block plan respectively.

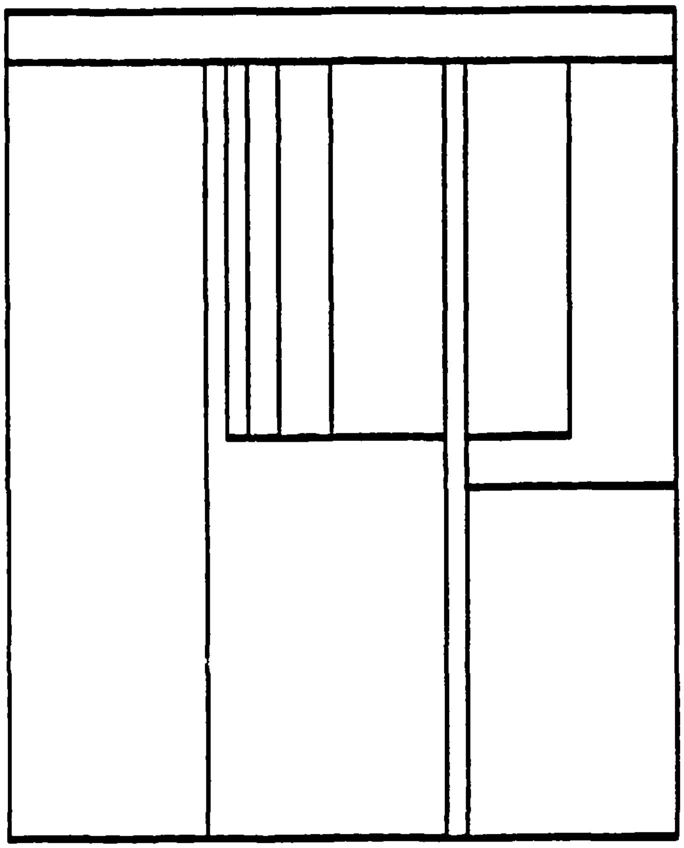


Figure 4.8. Example II Dual

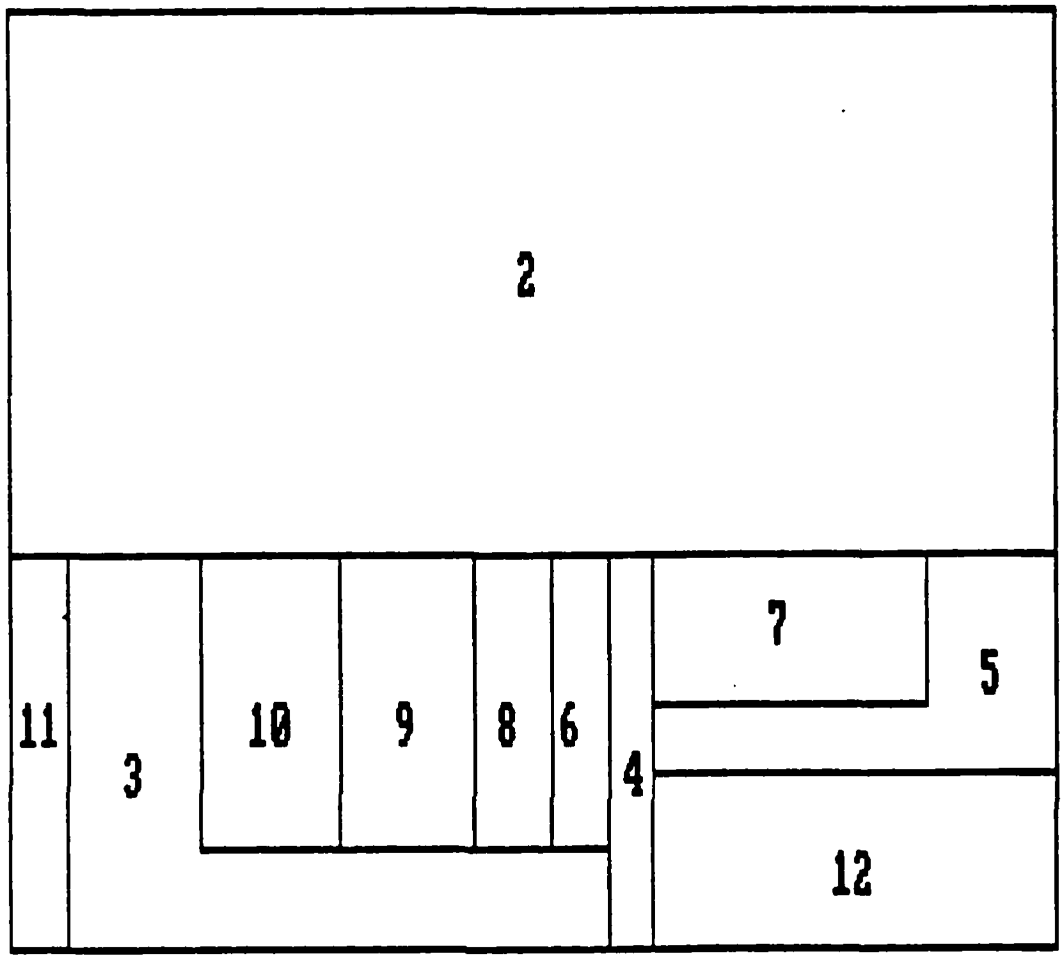
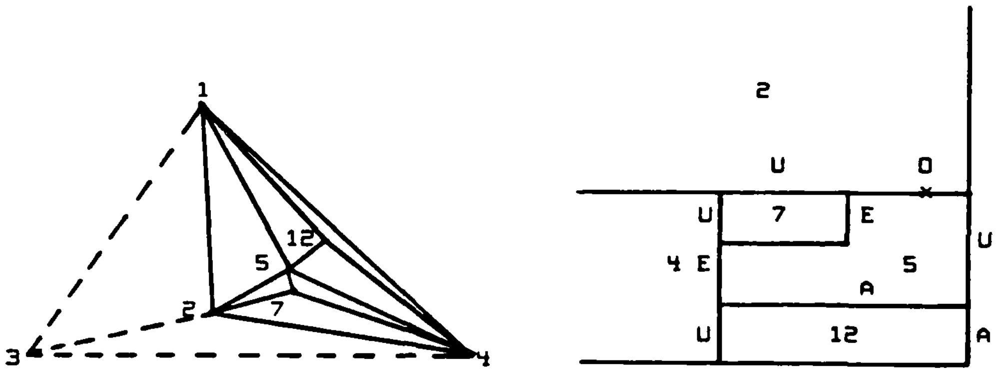
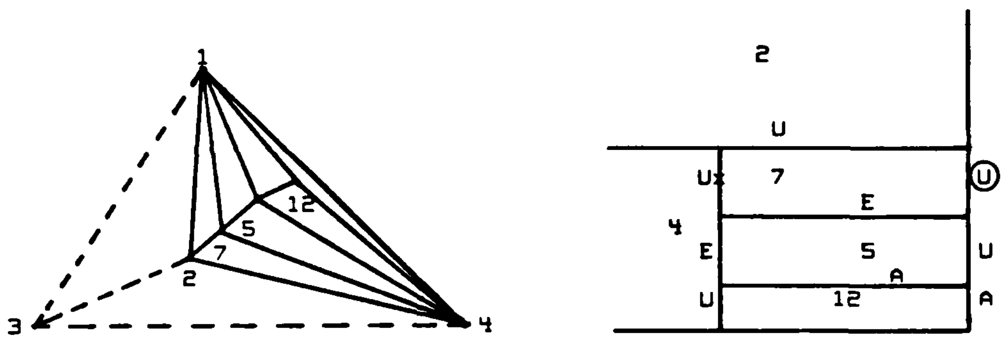


Figure 4.9. Example II Block Plan

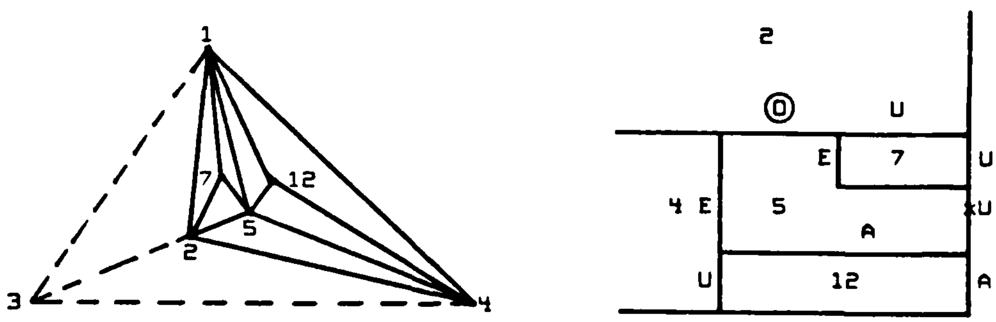
Consider the general manager's office (facility 5) and note that the L shape is not desirable. Consulting the REL chart it is also noted that the adjacency rating between facilities 7 and 12 is an E and the adjacency rating between 7 and 4 is only a U. With a series of edge swaps of the type described in the improved deltahedron, an increase in score can be achieved while also making the general manager's office a rectangle. The edge swaps and corresponding changes to the block plan are illustrated in figure 4.10.



Adjacency graph and dual after completion of original insertion order
(a)

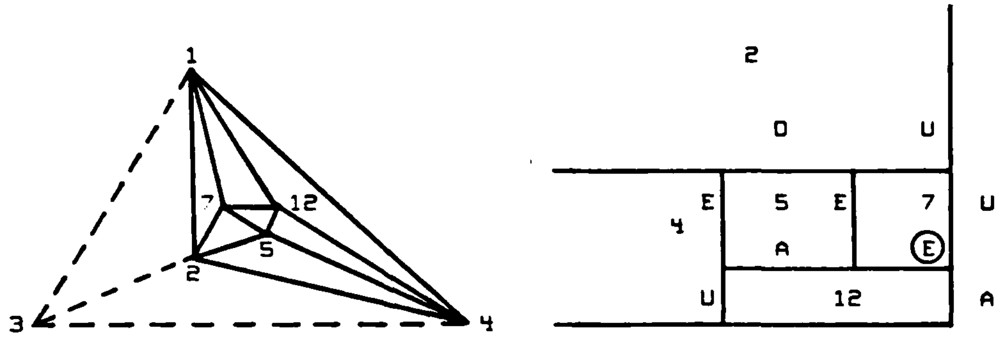


Adjacency graph and dual after one edge swap
(b)



Adjacency graph and dual after two edge swaps
(c)

Figure 4.10. Edge swap improvements to Example II



Adjacency graph and dual after three edge swaps
[d]

Figure 4.10--Continued.

It should be noted that since there is no vertex in the subgraph illustrated with degree three, there is no possible way to generate this graph using only the Deltahedron method as the last vertex inserted must have degree three. Additionally, the current Improved Deltahedron would not consider this sequence of changes since the first swap results in a lower score; therefore a look ahead procedure would be required. It is therefore proposed that every permissible edge swap can be characterized in the dual (and the block plan) as transforming a carve into a box or a box into a carve. It is further proposed that since every maximally planar graph can be constructed from an initial tetrahedron by a series of vertex insertions and edge swaps, [Giffin, 1984] if the sequence is known it is possible to construct the dual of all maximally planar adjacency graphs. The computer implementation of this procedure has not yet been done nor has the multiple step look ahead implementation of the Improved Deltahedron method. This is left for future research.

4-3 Example III

The final example is a real world problem and illustrates the degeneracy that often occurs in some larger actual problems. It also illustrates the outcome of a problem that is too large to be solved by the current program. Consider the REL chart illustrated in figure 4.11.

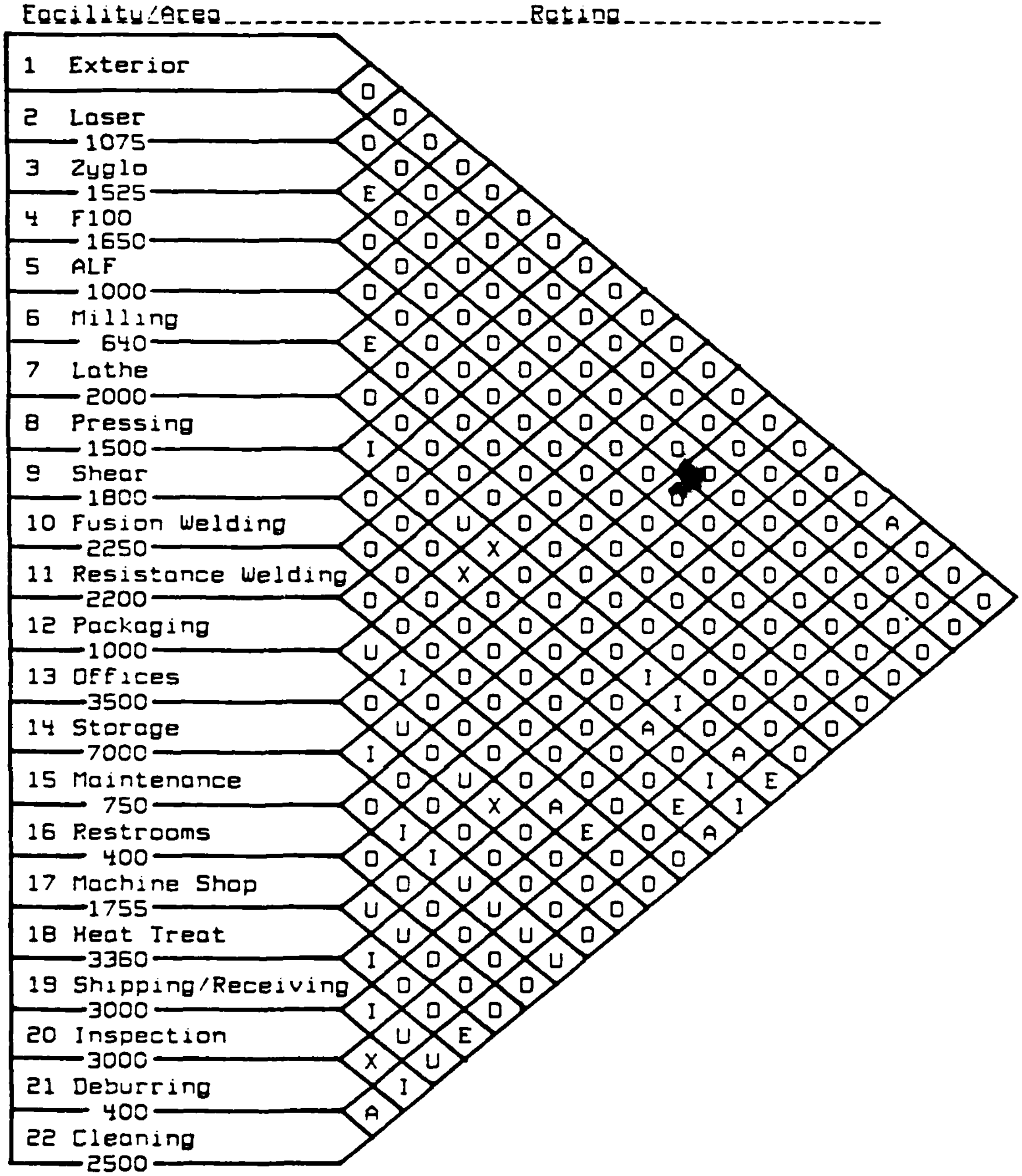


Figure 4.11. Example III REL Chart

It is clear from an inspection of the REL chart that facilities 2, 5, 11, and 16 have no rating other than O with respect to all other facilities. Therefore the problem is degenerate because when choosing among triangles for facilities 2, 5, 11, or 16, any triangle is as good as every other. Additionally, facilities 3 and 4 have an E only between themselves and an O with all others as do 6 and 7. It follows that as long as 3 and 4 are adjacent and 6 and 7 are adjacent, a block containing facilities 2, 3, 4, 5, 6, 7, 11, and 16 could be placed anywhere in the graph and result in the same score as placing it anywhere else. Because of this property, there are literally thousands of combinations that would result in the same score but have different adjacency graphs. One approach to this dilemma might be to group the 8 facilities into one large facility and thus reduce the size of the problem by more than a third. For the sake of demonstration however, the entire problem is run as given. This illustrates the problem encountered by the current program when the A$ matrix becomes too small to add all of the required facilities within it. The Deltahedron method runs without incident with the initial tetrahedron being facilities 1-19-22-21 and the insertion vertices and triangles are given in table 4.3. As with examples I and II, the complete output is contained in the appendix.

Table 4.3 Example III Vertices and Insertion Triangles

Vertex	Triangle
8	<19 22 21>
12	< 1 19 21>
10	< 8 22 21>
9	<19 21 8>
13	< 1 22 21>
18	<19 22 8>
20	< 1 19 12>
3	< 8 21 10>
4	< 8 21 3>
6	< 8 10 3>
7	< 6 10 3>
14	<20 19 12>
15	<18 22 8>
2	< 8 10 6>
5	< 8 6 2>
11	< 1 21 13>
16	< 8 3 4>
17	<15 22 8>

The resultant dual is given in figure 4.12 and it should be noted that there are only 19 facilities shown. Facilities 2, 5, and 16 were not able to be inserted since there was no room at the new location for an additional facility. The program can be continued normally from this point and output obtained, however the block plan will not contain the facilities left out of the dual (see figure 4.13).

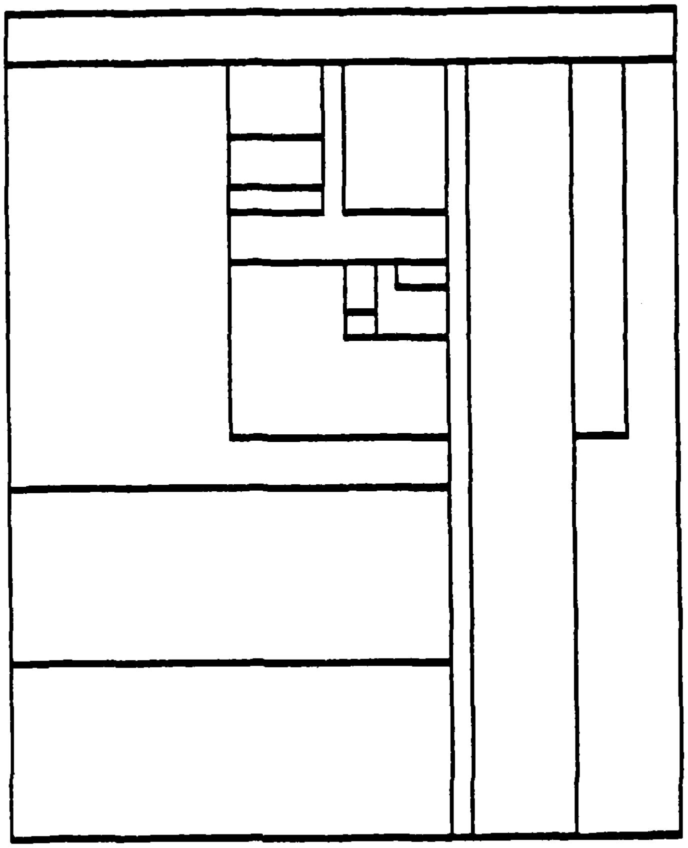


Figure 4.12. Example III Dual

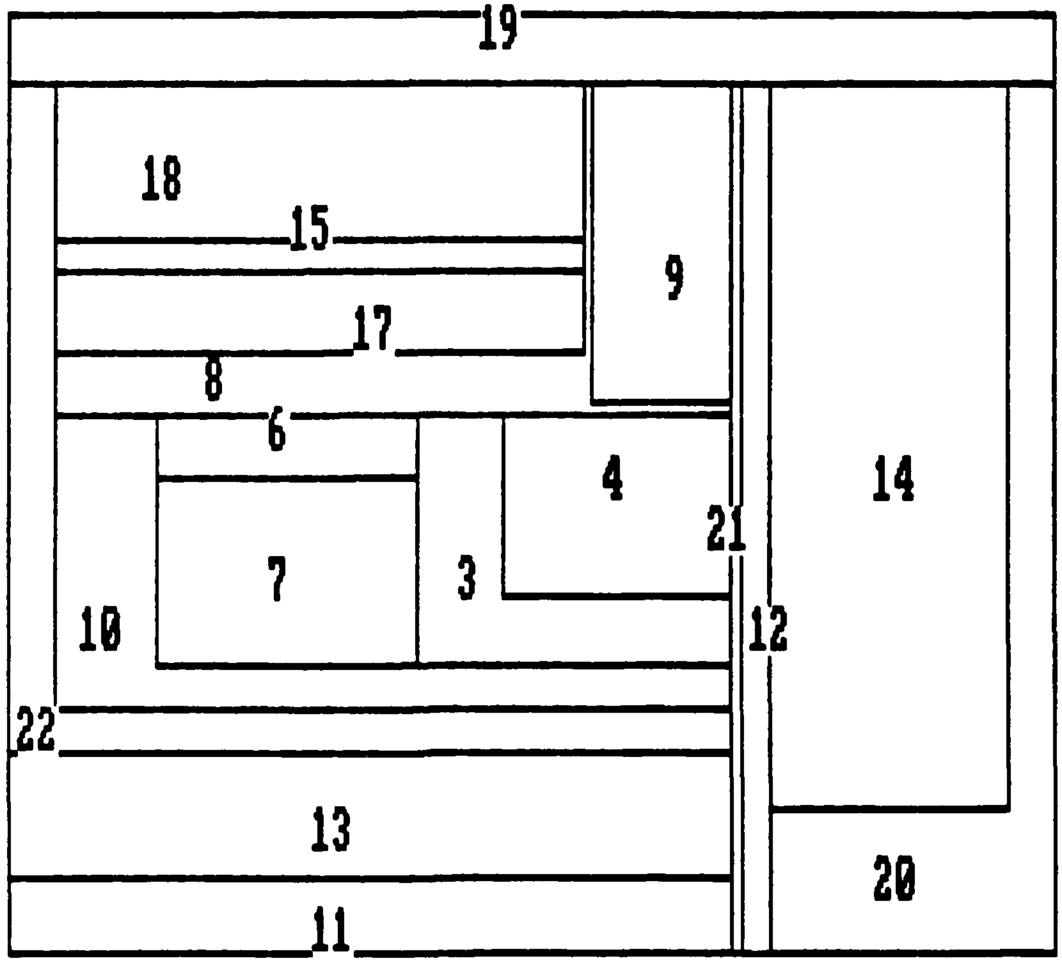


Figure 4.13. Example III Block Plan with 3 facilities not included

To provide a complete block plan, the BREAK feature of BASICA is used. Before continuing on to the construction of the block plan from the dual, the program execution is stopped with the BREAK key. When the program is halted in this manner, the variables defined up to this point remain in memory. The values not present for the complete construction of the block plan are the variables

OPER$ and PLIN for facilities 2, 5, and 16. An inspection of the condensed A$ matrix along with the insertion values displayed on the screen yield the necessary information to determine what the values would have been had the program had the necessary room. In this case the following variables were set to the values indicated below.

```
OPER$[18]="BRD"          PLIN[18]=14
OPER$[19]="CDL"          PLIN[19]=18
OPER$[21]="CDR"          PLIN[21]=13
```

After these values are set, execution is resumed and the result is given in figure 4.14. It should be noted that for different random seed values, DELTAPLAN will complete this problem with no variable redefinition required.

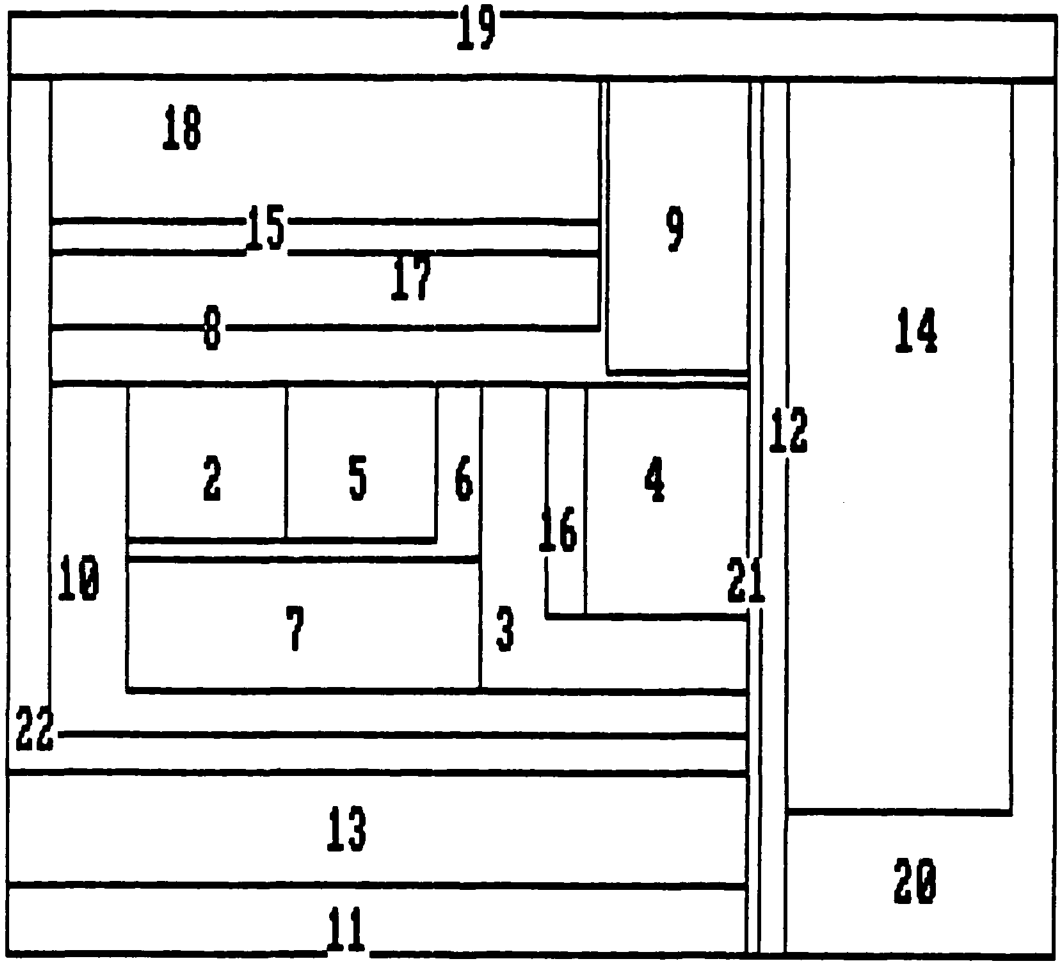


Figure 4.14. Example III Block Plan complete

A possible change to allow somewhat larger problems would be to redefine the data structure and to have an A$ matrix that starts out very small and expands out from the point where an additional facility is placed. This is opposite to the present method which starts with a given size and facilities are placed within its boundaries. Many times there is quite a lot of space remaining within the matrix; however, it is not where the new facility must be added.

CHAPTER 5

CONCLUSIONS

The method presented here has been shown to construct a rectangular geometric dual for the class of adjacency graphs developed using Deltahedron based heuristics. It has further demonstrated how areas can be incorporated to form a block plan. It should be remembered however, that all of the methods described in this thesis are analytical in nature and as stated in Francis and White [1974], "It should be realized that the analytical approach yields a solution to the model, but not necessarily the problem." For this reason, one should be cautious when selecting a block plan produced by any of the heuristics mentioned. Just because a particular plan has a higher adjacency score does not mean that it is a better plan. The maximally planar plans developed by DELTAPLAN sometimes have long narrow L or T shaped facilities which are most likely not very useful if included as shown in the block plan. The output of this as well as other methods is meant to be a starting place and guide for further planning. Alternatives that may not previously have been considered might surface with a computer method such as DELTAPLAN.

As a starting point, the plan and REL chart may be consulted to see if perhaps one of the adjacencies in a long narrow L or T shape is even worthwhile to have and as such it may become a candidate for deletion. If a graph is not very dense in highly weighted edges, perhaps a maximally planar block plan contains more adjacencies than are really necessary. In this case some adjacencies may be deleted to form a more regular plan and the adjacency score may not even be affected.

A very important fact is that the dual is not unique. There are many ways of arranging facilities with very slight changes to the rules of DELTAPLAN, that preserve all of the adjacencies required. One change might involve moving the initial inhibitors from the right side of the wall between facilities 3 and 4 to the left side. Another possibility would be to change the placement of facilities 2, 3, and 4 within the dual representation of the initial tetrahedron which would lead to six different orientations of the initial four facilities. These are either 2, 3, or 4 on the top and the remaining two facilities placed either on the left and right or the right and left. The point here is that the rules developed for DELTAPLAN continue to work for all of these orientations. If areas are not a factor or if it can be determined that no new facility might affect certain adjacencies after areas are added, changes to the inhibitors at a later stage can be invoked an result in still further alternatives.

As mentioned earlier, extensions can be made to include edge swaps of the type used in the Improved Deltahedron. Further extensions include the ability to develop the dual from any maximally planar adjacency graph once the series of vertex insertions and edge swaps required to form the adjacency graph from an initial tetrahedron are known. As yet it is not known how the process of enforcing a deltahedron like insertion and swapping procedure on an arbitrary adjacency graph should be performed efficiently. Another extension might be to develop the block plan in parallel with the Super Deltahedron method in order to have more accurate estimates of the distance between centroids for transportation cost estimates.

The implementation provided in this thesis should form an important subroutine to the realization of all of these extensions.

APPENDIX A

THE DELTAHEDRON HEURISTIC
PROGRAM LISTING

```
10      '---------------------------------------------------------------
20      '
30      '               The DELTAHEDRON HEURISTIC
40      '            using column sums insertion order
50      '
60      '                    by J. W. Giffin
70      '            with modifications by D. W. Keenan
80      '                     March 1, 1986
90      '---------------------------------------------------------------
100     '
110     DIM BEN [30,30],ORDER [30],BENSUM [30],TRIANG [64,3],SOLUTION [30,30]
120     DIM OTHERS [30],DEG [30],A [30],DEGCON [30],ROOTA [30]
130     DIM SPATH [30,30],HP [3],H [30],VALID[30],TRIANG$[30],
        AREA[30],P$[30,30]
140     CLS
150     RANDOMIZE
160     DEFINT I-N
170     INPUT "You will need to input the filename for the data you want to
        use.              Would you like a list of files on the disk
        [Y/N]";ANS$
180     IF ANS$="Y" OR ANS$="y" THEN FILES
190     INPUT"Enter any filename with .DAT for an extension";FILENAME$
200     IF RIGHT$(FILENAME$,4)<>".DAT" THEN FILENAME$=FILENAME$+".DAT"
210     PRINT
220     INPUT"If you need an X value other than -1024 enter it at the prompt,
if not press return." ;XVAL$
230     IF XVAL$="" THEN XVAL=-1024 ELSE XVAL=VAL(XVAL$)
240     '---------------------------------------------------------------
250     'Read data from data file and initialize NxN score matrix
260     '---------------------------------------------------------------
270     OPEN "I",#1,FILENAME$
280     INPUT #1, N
290     PRINT "NUMBER OF FACILITIES:"N
300     FLAG =0
310     K=7
320     FOR I=1 TO N
330             PRINT USING "##";I;:PRINT " : ";
340             PRINT TAB(K)
350             PRINT USING "##";I ;:PRINT " ";
360             FOR J=I+1 TO N
```

```
370                         INPUT #1, P$[I,J]
380                         PRINT P$[I,J] " ";
390                         IF P$[I,J]="U" THEN BEN [I,J]=0 : BEN [J,I]=0 :GOTO
                            450
400                         IF P$[I,J]="O" THEN BEN [I,J]=1 : BEN [J,I]=1 :GOTO
                            450
410                         IF P$[I,J]="I" THEN BEN [I,J]=4 : BEN [J,I]=4 :GOTO
                            450
420                         IF P$[I,J]="E" THEN BEN [I,J]=16 : BEN [J,I]=16 :GOTO
                            450
430                         IF P$[I,J]="A" THEN BEN [I,J]=64 : BEN [J,I]=64 :GOTO
                            450
440                         IF P$[I,J]="X" THEN BEN [I,J]=XVAL : BEN [J,I]=XVAL
                            :FLAG =1
450         NEXT J
460         K=K+2
470         PRINT
480         NEXT I
490         FOR I=2 TO N
500                 INPUT #1, AREA[I]
510         NEXT I
520         CLOSE
530         '-------------------------------------------------------------
540         'If an X is present, add a constant to all scores so they
550         'are all non-negative
560         '-------------------------------------------------------------
570         IF FLAG =0 GOTO 630
580         FOR I=1 TO N
590                 FOR J=1 TO N
600                         BEN [I,J]=BEN [I,J]-XVAL
610                 NEXT J
620         NEXT I
630         FOR I=1 TO N
640                 BEN [I,I]=0
650         NEXT I
660         GOSUB 1530
670         '-------------------------------------------------------------
680         'Initialize total score and add the score
690         'for the initial tetrahedron
700         '-------------------------------------------------------------
710         TOTBEN =0
720         FOR I=1 TO 4
730                 FOR J=I+1 TO 4
740                         TOTBEN =TOTBEN +BEN [ORDER [I],ORDER [J]]
750                 NEXT J
760         NEXT I
770         '-------------------------------------------------------------
780         'Determine best triangle for vertex insertion
790         '-------------------------------------------------------------
800         FOR I=5 TO N
810                 MAX=-1
```

```
820         X=ORDER [I]
830         CK=1+INT[RND*TRINO ]
840         FOR K=CK TO TRINO
850                 SUM =0
860                 FOR J=1 TO 3
870                         SUM =SUM +BEN [X ,TRIANG [K,J]]
880                 NEXT J
890                 IF SUM > MAX THEN MAX=SUM   : MAXTRI =K
900         NEXT K
910         FOR K=1 TO CK-1
920                 SUM =0
930                 FOR J=1 TO 3
940                         SUM =SUM +BEN [X ,TRIANG [K,J]]
950                 NEXT J
960                 IF SUM > MAX THEN MAX=SUM   : MAXTRI =K
970         NEXT K
980     '-----------------------------------------------------------------
990     'Print vertex and triangle it is inserted into
1000    '-----------------------------------------------------------------
1010        PRINT  "INSERTING VERTEX ";X ;" IN TRIANGLE ";
1020        FOR K= 1 TO I
1030                IF TRIANG [MAXTRI,1] = ORDER [K] THEN ELMNT1=K: GOTO
                    1050
1040        NEXT K
1050        PRINT TRIANG [MAXTRI ,1];
1060        FOR K= 1 TO I
1070                IF TRIANG [MAXTRI,2] = ORDER [K] THEN ELMNT2=K: GOTO
                    1090
1080        NEXT K
1090        PRINT TRIANG [MAXTRI ,2];
1100        FOR K= 1 TO I
1110                IF TRIANG [MAXTRI,3] = ORDER [K] THEN ELMNT3=K: GOTO
                    1130
1120        NEXT K
1130        PRINT TRIANG [MAXTRI ,3];
1140    '-----------------------------------------------------------------
1150    'Create character sting elements used as input for DELTAPLAN
1160    '-----------------------------------------------------------------
1170        IF ELMNT1<10 THEN ELMNT1$="0"+RIGHT$[STR$[ELMNT1],1] ELSE
            ELMNT1$=RIGHT$[STR$[ELMNT1],2]
1180        IF ELMNT2<10 THEN ELMNT2$="0"+RIGHT$[STR$[ELMNT2],1] ELSE
            ELMNT2$=RIGHT$[STR$[ELMNT2],2]
1190        IF ELMNT3<10 THEN ELMNT3$="0"+RIGHT$[STR$[ELMNT3],1] ELSE
            ELMNT3$=RIGHT$[STR$[ELMNT3],2]
1200        TRIANG$[I]=ELMNT1$+ELMNT2$+ELMNT3$
1210        PRINT TRIANG$[I]
1220        PRINT
1230        GOSUB 2130
1240        TOTBEN =TOTBEN +MAX
1250    NEXT I
1260    IF FLAG=1 THEN TOTBEN=TOTBEN + XVAL*[3*N-6]
```

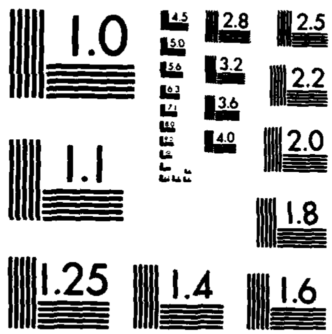
```
1270    PRINT
1280    PRINT "TOTAL DELTAHEDRON ADJACENCY SCORE IS" TOTBEN
1290    PRINT
1300    GOSUB 2280
1310    '-----------------------------------------------
1320    'Write output to data file
1330    '-----------------------------------------------
1340    OPEN "O",#1,"DATA1"
1350    WRITE #1,N
1360    FOR I=1 TO N
1370            WRITE #1,ORDER[I]
1380    NEXT I
1390    FOR I=5 TO N
1400            WRITE #1,TRIANG$[I]
1410    NEXT I
1420    FOR I=2 TO N
1430            WRITE #1,AREA[ORDER[I]]
1440    NEXT I
1450    CLOSE
1460    INPUT "WOULD YOU LIKE A LAYOUT DONE FROM THIS DATA [Y/N]";ANS$
1470    IF ANS$="N" OR ANS$="n" GOTO 1490
1480    CHAIN "DELTAPLN"
1490    END
1500    '-----------------------------------------------
1510    'Print NxN score matrix
1520    '-----------------------------------------------
1530    'FOR I=1 TO N
1540    '       FOR J=1 TO N
1550    '               PRINT BEN [I,J];
1560    '       NEXT J
1570    '       PRINT
1580    'NEXT I
1590    PRINT
1600    '-----------------------------------------------
1610    'Calculate column sums
1620    '-----------------------------------------------
1630    FOR J=1 TO N
1640            SUM =0
1650            FOR I=1 TO N
1660                    IF I<>J THEN SUM =SUM +BEN [I,J]
1670            NEXT I
1680            BENSUM [J]=SUM
1690    NEXT J
1700    FOR I=1 TO N
1710            VALID [I]=1
1720    NEXT I
1730    FOR I= 1 TO N
1740            ORDER [I]=I
1750    NEXT I
```

```
1760    '-----------------------------------------------------
1770    ' Sort vertices according to column sums
1780    ' Bubblesort array order according to BENSUM
1790    '-----------------------------------------------------
1800    FLIPS =1
1810    WHILE FLIPS =1
1820            FLIPS =0
1830            FOR I=2 TO N-1
1840                    IF BENSUM [ORDER [I]] < BENSUM [ORDER [I+1]] THEN SWAP
                        ORDER [I],ORDER [I+1] :FLIPS =1
1850            NEXT I
1860    WEND
1870    '-----------------------------------------------------
1880    'Print deltahedron insertion order
1890    '-----------------------------------------------------
1900    PRINT
1910    PRINT "DELTAHEDRON INSERTION ORDER"
1920    PRINT
1930    FOR I=1 TO N
1940            PRINT ORDER [I];
1950    NEXT I
1960    PRINT:PRINT
1970    '-----------------------------------------------------
1980    'Initialize triangles and incidence values for the
1990    'initial tetrahedron
2000    '-----------------------------------------------------
2010    FOR I=1 TO 4
2020            X =ORDER [I]
2030            FOR J=1 TO 4
2040                    Y =ORDER [J]
2050                    IF J<I THEN TRIANG [I,J]=Y ELSE IF J>I THEN TRIANG
                        [I,J-1]=Y : SOLUTION [X,Y]=1:SOLUTION [Y,X]=1
2060            NEXT J
2070    NEXT I
2080    TRINO =4
2090    RETURN
2100    '-----------------------------------------------------
2110    '<<< Relabel deleted triangle and add two more >>>
2120    '-----------------------------------------------------
2130    FOR J=1 TO 3
2140            SOLUTION [X ,TRIANG [MAXTRI,J]]=1
2150            SOLUTION [TRIANG [MAXTRI,J],X ]=1
2160    NEXT J
2170    TRINO =TRINO +1
2180    TRIANG [TRINO ,1]=TRIANG [MAXTRI,1]
2190    TRIANG [TRINO ,2]=TRIANG [MAXTRI,2]
2200    TRIANG [TRINO ,3]=X
2210    TRINO =TRINO +1
2220    TRIANG [TRINO ,1]=TRIANG [MAXTRI,1]
2230    TRIANG [TRINO ,2]=TRIANG [MAXTRI,3]
2240    TRIANG [TRINO ,3]=X
```

```
2250    TRIANG [MAXTRI,1]=X
2260    RETURN
2270    '-----------------------------------------------------
2280    '<<< Print matrix of adjacencies present >>>
2290    '-----------------------------------------------------
2300    PRINT "INCIDENCE MATRIX:"
2310    PRINT
2320    K=7
2330    FOR I=1 TO N
2340            PRINT I;
2350            PRINT TAB[K]
2360            FOR J=I+1 TO N
2370                    IF SOLUTION [I,J]=1 THEN PRINT P$[I,J];" "; ELSE PRINT
                        "- ";
2380            NEXT J
2390            K=K+2
2400            PRINT
2410    NEXT I
2420    PRINT
2430    RETURN
```

AD-A171 339 BLOCK PLAN CONSTRUCTION FROM A DELTAHEDRON BASED ADJACENCY GRAPH(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH D W KEENAN 1986 2/2

UNCLASSIFIED AFIT/CI/NR-86-128T F/G 12/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

APPENDIX B

DELTAPLAN
PROGRAM LISTING

```
10      '-----------------------------------------------------------------
20      '
30      '                       DELTAPLAN
40      'A Procedure to Construct a Rectangular Geometric Dual and
50      '   a Block Plan from a Deltahedron Based Adjacency Graph
60      '
70      '                    by David W. Keenan
80      '                     March 1, 1986
90      '-----------------------------------------------------------------
100     '
110     KEY OFF
120     SCREEN 1
130     WIDTH 80
140     CLS
150     '-----------------------------------------------------------------
160     '--[I,J] ARE THE INSERTION TRIANGLE COORDINATES
170     '--[I1,J1] ARE THE OPPOSITE CORNER COORDINATES OF THE NEW FACILITY
180     '--L IS THE HORIZONTAL COORDINATE OF THE NEXT ADJACENCY TRIANGLE TO
190     'THE LEFT
200     '--R IS THE HORIZONTAL COORDINATE OF THE NEXT ADJACENCY TRIANGLE TO
210     'THE RIGHT
220     '--D IS THE VERTICAL COORDINATE OF THE NEXT ADJACENCY TRIANGLE BELOW
230     '--U IS THE VERTICAL COORDINATE OF THE NEXT ADJACENCY TRIANGLE ABOVE
240     '--_FLAG0=0 INDICATES THE DIRECTION IS USABLE
250     '--_FLAG0=1 INDICATES THE DIRECTION IS NOT USABLE
260     '--_FLAG1 INDICATES A CORNER OR THE ABSENCE THEREOF -- 0=NO CORNER,
270     '1=LOWER LEFT CORNER, 2=LOWER RIGHT CORNER, 3=UPPER RIGHT CORNER,
280     '4=UPPER LEFT CORNER
290     '-----------------------------------------------------------------
300     DIM A$[35,66],R1$[200],R2[200],R3[200],PLIN[50],PLINS$[50],OPERS$[50],
        TRIANGLES$[50],AREA[50],AREAIN[50],ORDER[50]
310     PLACE = 3
320     CT=4
330     BL$=STRING$(79,32)
340     '>>> INITIALIZE MATRIX
350     GOSUB 1040
360     '>>> GET INPUT AND DETERMINE ITS COORDINATES
370     FOR FAC=5 TO NUMFAC
380             L=0
390             R=0
```

```
400           D=0
410           U=0
420           I=0
430           J=0
440           IF FAC < 10 THEN FAC$="0"+RIGHT$[STR$[FAC],1] ELSE
              FAC$=RIGHT$[STR$[FAC],2]
450           GOSUB 5430 '>>> CLEAR FIRST LINE
460           PRINT "Inserting Facility";ORDER[FAC];
470           LOCATE 2,1
480           PRINT BL$
490           AREAIN[FAC]=AREA[FAC]
500       '>>> SORT THE INPUT TO INSURE PROPER CHARACTER SEQUENCE
510           SORTA$=TRIANGLE$[FAC] : GOSUB 5070 : TRIANGLE$[FAC]=SORTA$
520           FOR X=1 TO CT
530                   IF TRIANGLE$[FAC]=R1$[X] THEN I=R2[X]:J=R3[X] :R2[X]=0
                      :R3[X]=0 :GOTO 550
540           NEXT X
550           IF I=0 THEN LOCATE 2,1 : PRINT "THIS TRIANGLE CAN'T BE FOUND
              AS LISTED -- TRY AGAIN ":GOTO 880
560       '>>> BEGIN SEARCH
570       '>>> SEARCH LEFT
580           GOSUB 1870
590       '>>> SEARCH RIGHT
600           GOSUB 2010
610       '>>> SEARCH DOWN
620           GOSUB 2150
630       '>>> SEARCH UP
640           GOSUB 2290
650       '>>> CHECK FOR CORNERS AND CARVE IF POSSIBLE
660           IF LFLAG1=1 AND LFLAG0=0 AND UFLAG0=0 THEN GOSUB 2430 :GOTO
              820
670           IF LFLAG1=4 AND LFLAG0=0 AND DFLAG0=0 THEN GOSUB 2640 :GOTO
              820
680           IF RFLAG1=2 AND RFLAG0=0 AND UFLAG0=0 THEN GOSUB 2850 :GOTO
              820
690           IF RFLAG1=3 AND RFLAG0=0 AND DFLAG0=0 THEN GOSUB 3060 :GOTO
              820
700           IF DFLAG1=1 AND DFLAG0=0 AND RFLAG0=0 THEN GOSUB 3270 :GOTO
              820
710           IF DFLAG1=2 AND DFLAG0=0 AND LFLAG0=0 THEN GOSUB 3480 :GOTO
              820
720           IF UFLAG1=3 AND UFLAG0=0 AND LFLAG0=0 THEN GOSUB 3690 :GOTO
              820
730           IF UFLAG1=4 AND UFLAG0=0 AND RFLAG0=0 THEN GOSUB 3900 :GOTO
              820
740       '>>> CHECK FOR INTERSECTIONS AND BOX IF POSSIBLE
750           IF LFLAG0=0 AND DFLAG0=0 THEN GOSUB 4110 :GOTO 820
760           IF LFLAG0=0 AND UFLAG0=0 THEN GOSUB 4350 :GOTO 820
770           IF RFLAG0=0 AND DFLAG0=0 THEN GOSUB 4590 :GOTO 820
780           IF RFLAG0=0 AND UFLAG0=0 THEN GOSUB 4830 :GOTO 820
790           LOCATE 2,1
```

```
800            PRINT "This triangle cannot be inserted here, try another
               location"
810            GOTO 880
820            'CONTINUE
830            LOCATE PLACE,50
840            PLIN[FAC]=VAL[PLIN$[FAC]]
850            PRINT ORDER[FAC];AREA[FAC];" ";OPER$[FAC];" ";TRIANGLE$[FAC];"
               ";PLIN[FAC]
860            IF PLACE >= 23 THEN PLACE = 4 ELSE PLACE = PLACE +1
870            LINE [J*4+28,I*4+50]-[J1*4+28,I1*4+50],,B
880     NEXT FAC
890     'FOR U=1 TO CT
900     '      PRINT R1$[U];R2[U];R3[U]
910     'NEXT U
920     'PRINT CT
930     GOSUB 5430 '>>> CLEAR FIRST LINE
940     INPUT "Would you like a layout copy printed [Y/N]";ANS$
950     IF ANS$ = "Y" OR ANS$ = "y" THEN GOSUB 1770
960     GOSUB 5430 '>>> CLEAR FIRST LINE
970     INPUT "Would you like an insertion order copy printed [Y/N]";ANS$
980     IF ANS$ = "Y" OR ANS$ = "y" THEN GOSUB 5370
990     GOSUB 5430 '>>> CLEAR FIRST LINE
1000    INPUT "Would you like to see the layout with areas [Y/N]";ANS$
1010    IF ANS$="Y" OR ANS$="y" THEN GOSUB 5480
1020    CHAIN "DELTASUM"
1030  END '--------------------------------------------------------
1040    '>>> INITIALIZE MATRIX
1050    OPEN "I",#1,"DATA1"
1060    INPUT #1,NUMFAC
1070    FOR I=1 TO NUMFAC
1080           INPUT #1,ORDER[I]
1090    NEXT I
1100    FOR I=5 TO NUMFAC
1110           INPUT #1,TRIANGLE$[I]
1120    NEXT I
1130    FOR I=2 TO NUMFAC
1140           INPUT #1,AREA[I]
1150    NEXT I
1160    CLOSE
1170    LOCATE 10,5
1180    PRINT "Please wait a few moments while things are being
        initialized...."
1190    FOR J=0 TO 66
1200           A$[0,J]="01"
1210           A$[1,J]="-"
1220           A$[3,J]="-"
1230           A$[34,J]="-"
1240           A$[35,J]="01"
1250    NEXT J
1260    FOR I=1 TO 34
1270           A$[I,0]="01"
```

```
1280            A$[I,1]="-"
1290            A$[I,43]="-"
1300            A$[I,65]="-"
1310            A$[I,66]="01"
1320    NEXT I
1330    FOR J=2 TO 64
1340            A$[2,J]="02"
1350    NEXT J
1360    FOR I=4 TO 33
1370            FOR J1=2 TO 42
1380                    A$[I,J1]="03"
1390            NEXT J1
1400            FOR J2=44 TO 64
1410                    A$[I,J2]="04"
1420            NEXT J2
1430    NEXT I
1440    A$[1,1]="000102"
1450    A$[1,65]="000102"
1460    A$[3,1]="010203"
1470    A$[3,43]="020304"
1480    A$[3,65]="010204"
1490    A$[34,1]="AA0103"
1500    A$[34,43]="010304"
1510    A$[34,65]="BB0104"
1520    A$[3,44]="000000"
1530    A$[34,44]="000000"
1540    CLS
1550    LOCATE PLACE,50
1560    PRINT ORDER[2];":";AREA[2];ORDER[3];":";AREA[3];ORDER[4];":";AREA[4]
1570    PLACE=PLACE+1
1580    LINE [32,54]-[288,186],,B
1590    LINE [32,62]-[200,186],,B
1600    LINE [200,62]-[288,186],,B
1610    R1$[1]="010203"
1620    R2[1]=3
1630    R3[1]=1
1640    R1$[2]="020304"
1650    R2[2]=3
1660    R3[2]=43
1670    R1$[3]="010204"
1680    R2[3]=3
1690    R3[3]=65
1700    R1$[4]="010304"
1710    R2[4]=34
1720    R3[4]=43
1730    AREAIN[2]=AREA[2]
1740    AREAIN[3]=AREA[3]
1750    AREAIN[4]=AREA[4]
1760     RETURN
1770    '>>> <<< PRINT A$ MATRIX
1780    FOR J=66 TO 0 STEP -1
```

```
1790            FOR I=0 TO 35
1800                    PRINT A$[I,J] " ";
1810                    IF LEN[A$[I,J]]=6 AND A$[I,J]<>"000000" THEN AA$="-"
                        ELSE AA$=RIGHT$[A$[I,J],2]
1820                    IF AA$="-" THEN LPRINT " -"; ELSE LPRINT USING
                        "##";ORDER[VAL[AA$]];
1830            NEXT I
1840            LPRINT
1850    NEXT J
1860    RETURN
1870    '>>> <<< SEARCH LEFT
1880    LFLAG0=0
1890    LFLAG1=0
1900    LFLAG2=0
1910    L=J
1920    L=L-1
1930    IF L<1 THEN LFLAG0=1: GOTO 2000
1940    AVAL$=LEFT$[A$[I,L],1]
1950    IF AVAL$="-" THEN GOTO 1920
1960    IF AVAL$="A" THEN LFLAG1=1
1970    IF AVAL$="D" THEN LFLAG1=4
1980    IF A$[I,L]="000000" THEN LFLAG2=1
1990    IF L=J-1 OR L=J-2 THEN LFLAG0=1
2000    RETURN
2010    '>>> <<< SEARCH RIGHT
2020    RFLAG0=0
2030    RFLAG1=0
2040    RFLAG2=0
2050    R=J
2060    R=R+1
2070    IF R>65 THEN RFLAG0=1: GOTO 2140
2080    AVAL$=LEFT$[A$[I,R],1]
2090    IF AVAL$="-" THEN GOTO 2060
2100    IF AVAL$="B" THEN RFLAG1=2
2110    IF AVAL$="C" THEN RFLAG1=3
2120    IF A$[I,R]="000000" THEN RFLAG2=1
2130    IF R=J+1 OR R=J+2 THEN RFLAG0=1
2140    RETURN
2150    '>>> <<< SEARCH DOWN
2160    DFLAG0=0
2170    DFLAG1=0
2180    DFLAG2=0
2190    D=I
2200    D=D+1
2210    IF D>34 THEN DFLAG0=1: GOTO 2280
2220    AVAL$=LEFT$[A$[D,J],1]
2230    IF AVAL$="-" THEN GOTO 2200
2240    IF AVAL$="A" THEN DFLAG1=1
2250    IF AVAL$="B" THEN DFLAG1=2
2260    IF A$[D,J]="000000" THEN DFLAG2=1
2270    IF D=I+1 OR D=I+2 THEN DFLAG0=1
```

```
2280    RETURN
2290    '>>> <<< SEARCH UP
2300    UFLAG0=0
2310    UFLAG1=0
2320    UFLAG2=0
2330    U=I
2340    U=U-1
2350    IF U<1 THEN UFLAG0=1: GOTO 2420
2360    AVAL$=LEFT$[A$[U,J],1]
2370    IF AVAL$="-" THEN GOTO 2340
2380    IF AVAL$="C" THEN UFLAG1=3
2390    IF AVAL$="D" THEN UFLAG1=4
2400    IF A$[U,J]="000000" THEN UFLAG2=1
2410    IF U=I-1 OR U=I-2 THEN UFLAG0=1
2420    RETURN
2430    '>>> <<< CARVE LEFT-UP
2440    IF UFLAG2=1 THEN I1=U+1 ELSE I1=I-[INT[ABS[  J]/2]]
2450    J1=L
2460    L$=A$[I1+1,J1-1]
2470    U$=A$[I1-1,J1+1]
2480    R$=A$[I1+1,J+1]
2490    SORTA$=L$+U$+FAC$ : GOSUB 5070 :
        A$[I1,J1]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I1:R3[CT]=J1
2500    SORTA$=R$+U$+FAC$ : GOSUB 5070 :
        A$[I1,J]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I1:R3[CT]=J
2510    SORTA$=L$+R$+FAC$ : GOSUB 5070 :
        A$[I,J]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I:R3[CT]=J
2520    SORTA$="AA"+L$+FAC$ : GOSUB 5070 : A$[I,J1]=SORTA$
2530    A$[I1-1,J]="000000"
2540    A$[I1-1,J1]="000000"
2550    OPER$[FAC]="CLU"
2560    PLIN$[FAC]=U$
2570    FOR J2=J1+1 TO J-1
2580            A$[I1,J2]="-"
2590            FOR I2=I1+1 TO I-1
2600                    A$[I2,J2]=FAC$
2610            NEXT I2
2620    NEXT J2
2630    RETURN
2640    '>>> <<< CARVE LEFT-DOWN
2650    IF DFLAG2=1 THEN I1=D-1 ELSE I1=I+[INT[ABS[I-D]/2]]
2660    J1=L
2670    L$=A$[I1-1,J1-1]
2680    D$=A$[I1+1,J1+1]
2690    R$=A$[I1-1,J+1]
2700    SORTA$=L$+D$+FAC$ : GOSUB 5070 :
        A$[I1,J1]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I1:R3[CT]=J1
2710    SORTA$=R$+D$+FAC$ : GOSUB 5070 :
        A$[I1,J]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I1:R3[CT]=J
2720    SORTA$=L$+R$+FAC$ : GOSUB 5070 :
        A$[I,J]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I:R3[CT]=J
```

```
2730    SORTA$="DD"+L$+FAC$ : GOSUB 5070 : A$[I,J1]=SORTA$
2740    A$[I1+1,J]="000000"
2750    A$[I1+1,J1]="000000"
2760    OPER$[FAC]="CLD"
2770    PLIN$[FAC]=D$
2780    FOR J2=J1+1 TO J-1
2790            A$[I1,J2]="-"
2800            FOR I2=I+1 TO I1-1
2810                    A$[I2,J2]=FAC$
2820            NEXT I2
2830    NEXT J2
2840    RETURN
2850    '>>> <<< CARVE RIGHT-UP
2860    IF UFLAG2=1 THEN I1=U+1 ELSE I1=I-[INT[ABS[I-U]/2]]
2870    J1=R
2880    L$=A$[I1+1,J-1]
2890    U$=A$[I1-1,J1-1]
2900    R$=A$[I1+1,J1+1]
2910    SORTA$=R$+U$+FAC$ : GOSUB 5070 :
        A$[I1,J1]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I1:R3[CT]=J1
2920    SORTA$=L$+U$+FAC$ : GOSUB 5070 :
        A$[I1,J]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I1:R3[CT]=J
2930    SORTA$=L$+R$+FAC$ : GOSUB 5070 :
        A$[I,J]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I:R3[CT]=J
2940    SORTA$="BB"+R$+FAC$ : GOSUB 5070 : A$[I,J1]=SORTA$
2950    A$[I1-1,J]="000000"
2960    A$[I1-1,J1]="000000"
2970    OPER$[FAC]="CRU"
2980    PLIN$[FAC]=U$
2990    FOR J2=J+1 TO J1-1
3000            A$[I1,J2]="-"
3010            FOR I2=I1+1 TO I-1
3020                    A$[I2,J2]=FAC$
3030            NEXT I2
3040    NEXT J2
3050    RETURN
3060    '>>> <<< CARVE RIGHT-DOWN
3070    IF DFLAG2=1 THEN I1=D-1 ELSE I1=I+[INT[ABS[I-D]/2]]
3080    J1=R
3090    L$=A$[I1-1,J-1]
3100    D$=A$[I1+1,J1-1]
3110    R$=A$[I1-1,J1+1]
3120    SORTA$=R$+D$+FAC$ : GOSUB 5070 :
        A$[I1,J1]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I1:R3[CT]=J1
3130    SORTA$=L$+D$+FAC$ : GOSUB 5070 :
        A$[I1,J]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I1:R3[CT]=J
3140    SORTA$=L$+R$+FAC$ : GOSUB 5070 :
        A$[I,J]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I:R3[CT]=J
3150    SORTA$="CC"+R$+FAC$ : GOSUB 5070 : A$[I,J1]=SORTA$
3160    A$[I1+1,J]="000000"
3170    A$[I1+1,J1]="000000"
```

```
3180    OPER$[FAC]="CRD"
3190    PLIN$[FAC]=D$
3200    FOR J2=J+1 TO J1-1
3210            A$[I1,J2]="-"
3220            FOR I2=I+1 TO I1-1
3230                    A$[I2,J2]=FAC$
3240            NEXT I2
3250    NEXT J2
3260    RETURN
3270    '>>> <<< CARVE DOWN-RIGHT
3280    I1=D
3290    IF RFLAG2=1 THEN J1=R-1 ELSE J1=J+[INT[ABS[J-R]/2]]
3300    U$=A$[I-1,J1-1]
3310    R$=A$[I1-1,J1+1]
3320    D$=A$[I1+1,J1-1]
3330    SORTA$=R$+D$+FAC$ : GOSUB 5070 :
        A$[I1,J1]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I1:R3[CT]=J1
3340    SORTA$="AA"+D$+FAC$ : GOSUB 5070 : A$[I1,J]=SORTA$
3350    SORTA$=U$+D$+FAC$ : GOSUB 5070 :
        A$[I,J]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I:R3[CT]=J
3360    SORTA$=R$+U$+FAC$ : GOSUB 5070 :
        A$[I,J1]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I:R3[CT]=J1
3370    A$[I,J1+1]="000000"
3380    A$[I1,J1+1]="000000"
3390    OPER$[FAC]="CDR"
3400    PLIN$[FAC]=R$
3410    FOR I2=I+1 TO I1-1
3420            A$[I2,J1]="-"
3430            FOR J2=J+1 TO J1-1
3440                    A$[I2,J2]=FAC$
3450            NEXT J2
3460    NEXT I2
3470    RETURN
3480    '>>> <<< CARVE DOWN-LEFT
3490    I1=D
3500    IF LFLAG2=1 THEN J1=L+1 ELSE J1=J-[INT[ABS[J-L]/2]]
3510    U$=A$[I-1,J1+1]
3520    L$=A$[I1-1,J1-1]
3530    D$=A$[I1+1,J1+1]
3540    SORTA$=L$+D$+FAC$ : GOSUB 5070 :
        A$[I1,J1]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I1:R3[CT]=J1
3550    SORTA$="BB"+D$+FAC$ : GOSUB 5070 : A$[I1,J]=SORTA$
3560    SORTA$=U$+D$+FAC$ : GOSUB 5070 :
        A$[I,J]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I:R3[CT]=J
3570    SORTA$=L$+U$+FAC$ : GOSUB 5070 :
        A$[I,J1]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I:R3[CT]=J1
3580    A$[I,J1-1]="000000"
3590    A$[I1,J1-1]="000000"
3600    OPER$[FAC]="CDL"
3610    PLIN$[FAC]=L$
3620    FOR I2=I+1 TO I1-1
```

```
3630            A$[I2,J1]="-"
3640            FOR J2=J1+1 TO J-1
3650                    A$[I2,J2]=FAC$
3660            NEXT J2
3670    NEXT I2
3680    RETURN
3690    '>>> <<< CARVE UP-LEFT
3700    I1=U
3710    IF LFLAG2=1 THEN J1=L+1 ELSE J1=J-[INT[ABS[J-L]/2]]
3720    U$=A$[I1-1,J1+1]
3730    L$=A$[I1+1,J1-1]
3740    D$=A$[I+1,J1+1]
3750    SORTA$=L$+U$+FAC$ : GOSUB 5070 :
        A$[I1,J1]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I1:R3[CT]=J1
3760    SORTA$="CC"+U$+FAC$ : GOSUB 5070 : A$[I1,J]=SORTA$
3770    SORTA$=U$+D$+FAC$ : GOSUB 5070 :
        A$[I,J]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I:R3[CT]=J
3780    SORTA$=L$+D$+FAC$ : GOSUB 5070 :
        A$[I,J1]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I:R3[CT]=J1
3790    A$[I,J1-1]="000000"
3800    A$[I1,J1-1]="000000"
3810    OPER$[FAC]="CUL"
3820    PLIN$[FAC]=L$
3830    FOR I2=I1+1 TO I-1
3840            A$[I2,J1]="-"
3850            FOR J2=J1+1 TO J-1
3860                    A$[I2,J2]=FAC$
3870            NEXT J2
3880    NEXT I2
3890    RETURN
3900    '>>> <<< CARVE UP-RIGHT
3910    I1=U
3920    IF RFLAG2=1 THEN J1=R-1 ELSE J1=J+[INT[ABS[J-R]/2]]
3930    U$=A$[I1-1,J1-1]
3940    R$=A$[I1+1,J1+1]
3950    D$=A$[I+1,J1-1]
3960    SORTA$=R$+U$+FAC$ : GOSUB 5070 :
        A$[I1,J1]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I1:R3[CT]=J1
3970    SORTA$="DD"+U$+FAC$ : GOSUB 5070 : A$[I1,J]=SORTA$
3980    SORTA$=U$+D$+FAC$ : GOSUB 5070 :
        A$[I,J]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I:R3[CT]=J
3990    SORTA$=R$+D$+FAC$ : GOSUB 5070 :
        A$[I,J1]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I:R3[CT]=J1
4000    A$[I,J1+1]="000000"
4010    A$[I1,J1+1]="000000"
4020    OPER$[FAC]="CUR"
4030    PLIN$[FAC]=R$
4040    FOR I2=I1+1 TO I-1
4050            A$[I2,J1]="-"
4060            FOR J2=J+1 TO J1-1
4070                    A$[I2,J2]=FAC$
```

```
4080            NEXT J2
4090    NEXT I2
4100    RETURN
4110    '>>> <<< BOX LEFT-DOWN
4120    IF DFLAG2=1 THEN I1=D-1 ELSE I1=I+[INT[ABS[I-D]/2]]
4130    IF LFLAG2=1 THEN J1=L+1 ELSE J1=J-[INT[ABS[J-L]/2]]
4140    L$=A$[I+1,J1-1]
4150    U$=A$[I-1,J1+1]
4160    R$=A$[I+1,J+1]
4170    SORTA$="AA"+L$+FAC$ : GOSUB 5070 : A$[I1,J1]=SORTA$
4180    SORTA$=R$+L$+FAC$ : GOSUB 5070 :
        A$[I1,J]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I1:R3[CT]=J
4190    SORTA$=U$+R$+FAC$ : GOSUB 5070 :
        A$[I,J]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I:R3[CT]=J
4200    SORTA$=U$+L$+FAC$ : GOSUB 5070 :
        A$[I,J1]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I:R3[CT]=J1
4210    A$[I1+1,J]="000000"
4220    A$[I,J1-1]="000000"
4230    OPER$[FAC]="BLD"
4240    PLIN$[FAC]=L$
4250    FOR I2=I+1 TO I1-1
4260            A$[I2,J1]="-"
4270    NEXT I2
4280    FOR J2=J1+1 TO J-1
4290            A$[I1,J2]="-"
4300            FOR I2=I+1 TO I1-1
4310                    A$[I2,J2]=FAC$
4320            NEXT I2
4330    NEXT J2
4340    RETURN
4350    '>>> <<< BOX LEFT-UP
4360    IF UFLAG2=1 THEN I1=U+1 ELSE I1=I-[INT[ABS[I-U]/2]]
4370    IF LFLAG2=1 THEN J1=L+1 ELSE J1=J-[INT[ABS[J-L]/2]]
4380    L$=A$[I-1,J1-1]
4390    D$=A$[I+1,J1+1]
4400    R$=A$[I-1,J+1]
4410    SORTA$="DD"+L$+FAC$ : GOSUB 5070 : A$[I1,J1]=SORTA$
4420    SORTA$=R$+L$+FAC$ : GOSUB 5070 :
        A$[I1,J]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I1:R3[CT]=J
4430    SORTA$=D$+R$+FAC$ : GOSUB 5070 :
        A$[I,J]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I:R3[CT]=J
4440    SORTA$=D$+L$+FAC$ : GOSUB 5070 :
        A$[I,J1]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I:R3[CT]=J1
4450    A$[I1-1,J]="000000"
4460    A$[I,J1-1]="000000"
4470    OPER$[FAC]="BLU"
4480    PLIN$[FAC]=L$
4490    FOR I2=I1+1 TO I-1
4500            A$[I2,J1]="-"
4510    NEXT I2
4520    FOR J2=J1+1 TO J-1
```

```
4530            A$[I1,J2]="-"
4540            FOR I2=I1+1 TO I-1
4550                    A$[I2,J2]=FAC$
4560            NEXT I2
4570    NEXT J2
4580    RETURN
4590    '>>> <<< BOX RIGHT-DOWN
4600    IF DFLAG2=1 THEN I1=D-1 ELSE I1=I+[INT[ABS[I-D]/2]]
4610    IF RFLAG2=1 THEN J1=R-1 ELSE J1=J+[INT[ABS[J-R]/2]]
4620    L$=A$[I+1,J-1]
4630    U$=A$[I-1,J1-1]
4640    R$=A$[I+1,J1+1]
4650    SORTA$="BB"+R$+FAC$ : GOSUB 5070 : A$[I1,J1]=SORTA$
4660    SORTA$=R$+L$+FAC$ : GOSUB 5070 :
        A$[I1,J]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I1:R3[CT]=J
4670    SORTA$=U$+L$+FAC$ : GOSUB 5070 :
        A$[I,J]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I:R3[CT]=J
4680    SORTA$=U$+R$+FAC$ : GOSUB 5070 :
        A$[I,J1]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I:R3[CT]=J1
4690    A$[I1+1,J]="000000"
4700    A$[I,J1+1]="000000"
4710    OPER$[FAC]="BRD"
4720    PLIN$[FAC]=R$
4730    FOR I2=I+1 TO I1-1
4740            A$[I2,J1]="-"
4750    NEXT I2
4760    FOR J2=J+1 TO J1-1
4770            A$[I1,J2]="-"
4780            FOR I2=I+1 TO I1-1
4790                    A$[I2,J2]=FAC$
4800            NEXT I2
4810    NEXT J2
4820    RETURN
4830    '>>> <<< BOX RIGHT-UP
4840    IF UFLAG2=1 THEN I1=U+1 ELSE I1=I-[INT[ABS[I-U]/2]]
4850    IF RFLAG2=1 THEN J1=R-1 ELSE J1=J+[INT[ABS[J-R]/2]]
4860    L$=A$[I-1,J-1]
4870    D$=A$[I+1,J1-1]
4880    R$=A$[I-1,J1+1]
4890    SORTA$="CC"+R$+FAC$ : GOSUB 5070 : A$[I1,J1]=SORTA$
4900    SORTA$=R$+L$+FAC$ : GOSUB 5070 :
        A$[I1,J]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I1:R3[CT]=J
4910    SORTA$=D$+L$+FAC$ : GOSUB 5070 :
        A$[I,J]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I:R3[CT]=J
4920    SORTA$=D$+R$+FAC$ : GOSUB 5070 :
        A$[I,J1]=SORTA$:CT=CT+1:R1$[CT]=SORTA$:R2[CT]=I:R3[CT]=J1
4930    A$[I1-1,J]="000000"
4940    A$[I,J1+1]="000000"
4950    OPER$[FAC]="BRU"
4960    PLIN$[FAC]=R$
4970    FOR I2=I1+1 TO I-1
```

```
4980            A$[I2,J1]="-"
4990    NEXT I2
5000    FOR J2=J+1 TO J1-1
5010            A$[I1,J2]="-"
5020            FOR I2=I1+1 TO I-1
5030                    A$[I2,J2]=FAC$
5040            NEXT I2
5050    NEXT J2
5060    RETURN
5070    '>>> <<< SORT ROUTINE
5080    'SORTA$ IS THE ELEMENT TO BE PUT IN NUMERICAL ORDER
5090    N1$=MID$[SORTA$,1,2]
5100    N2$=MID$[SORTA$,3,2]
5110    N3$=MID$[SORTA$,5,2]
5120    IF N1$="AA" OR N1$="BB" OR N1$="CC" OR N1$="DD" GOTO 5270
5130    N[1]=VAL[N1$]
5140    N[2]=VAL[N2$]
5150    N[3]=VAL[N3$]
5160    FOR X=1 TO 2
5170            FOR Y=X+1 TO 3
5180                    IF N[X] <= N[Y] GOTO 5220
5190                    H=N[X]
5200                    N[X]=N[Y]
5210                    N[Y]=H
5220            NEXT Y
5230    NEXT X
5240    IF N[1]<10 THEN N1$="0"+RIGHT$[STR$[N[1]],1] ELSE
        N1$=RIGHT$[STR$[N[1]],2]
5250    GOTO 5330
5260    GOTO 5360
5270    N[2]=VAL[N2$]
5280    N[3]=VAL[N3$]
5290    IF N[2] <= N[3] GOTO 5330
5300    H=N[2]
5310    N[2]=N[3]
5320    N[3]=H
5330    IF N[2]<10 THEN N2$="0"+RIGHT$[STR$[N[2]],1] ELSE
        N2$=RIGHT$[STR$[N[2]],2]
5340    IF N[3]<10 THEN N3$="0"+RIGHT$[STR$[N[3]],1] ELSE
        N3$=RIGHT$[STR$[N[3]],2]
5350    SORTA$=N1$+N2$+N3$
5360    RETURN
5370    '>>> <<< PRINT INSERTION ORDER
5380    LPRINT ORDER[2];":";AREA[2];ORDER[3];":";AREA[3];ORDER[4];":";AREA[4]
5390    FOR I=5 TO FAC-1
5400            LPRINT ORDER[I];AREA[I];" ";OPERS[I];" ";TRIANGLES$[I];"
                ";PLIN[I]
5410    NEXT I
5420    RETURN
5430    '>>> <<< CLEAR FIRST LINE
5440    LOCATE 1,1
```

```
5450    PRINT BL$
5460    LOCATE 1,1
5470    RETURN
5480   '>>> <<< AREA CALCULATIONS
5490    DIM ULI[50],ULJ[50],URI[50],URJ[50],LLI[50],LLJ[50],LRI[50],LRJ[50]
5500    CLS
5510    FAC=FAC-1
5520    PLIN[3]=2
5530    PLIN[4]=3
5540    FOR I=FAC TO 2 STEP -1
5550            AREAIN[PLIN[I]]=AREAIN[PLIN[I]]+AREAIN[I]
5560    NEXT I
5570    AREATOT=AREAIN[2]
5580    ULI[2]=0:ULJ[2]=0
5590    URI[2]=0:URJ[2]=1
5600    LLI[2]=1:LLJ[2]=0
5610    LRI[2]=1:LRJ[2]=1
5620    DRW=2
5630    GOSUB 8550
5640    CARVE=3 :I=2
5650    GOSUB 6110
5660    CARVE=4 :I=3
5670    GOSUB 6590
5680    FOR I=3 TO FAC
5690            CARVE=0
5700            BOX1=0
5710            BOX2=0
5720            FOR I1=I TO FAC
5730                    IF PLIN[I1]=I AND LEFT$[OPER$[I1],1]="C" THEN CARVE=I1
5740                    IF PLIN[I1]=I AND LEFT$[OPER$[I1],1]="B" THEN
                        BOX1=I1:GOTO 5770
5750            NEXT I1
5760            GOTO 5810
5770            FOR I2=I1+1 TO FAC
5780                    IF PLIN[I2]=I AND LEFT$[OPER$[I2],1]="C" THEN CARVE=I2
5790                    IF PLIN[I2]=I AND LEFT$[OPER$[I2],1]="B" THEN
                        BOX2=I2:GOTO 5810
5800            NEXT I2
5810            'CONTINUE
5820            IF AREA[BOX2]>AREA[BOX1] THEN SWAP BOX1,BOX2
5830            IF CARVE=0 GOTO 5880
5840            IF OPER$[CARVE]="CLU" OR OPER$[CARVE]="CRU" THEN GOSUB
                6110 :GOTO 5880
5850            IF OPER$[CARVE]="CLD" OR OPER$[CARVE]="CRD" THEN GOSUB 6270
                :GOTO 5880
5860            IF OPER$[CARVE]="CDR" OR OPER$[CARVE]="CUR" THEN GOSUB 6430
                :GOTO 5880
5870            IF OPER$[CARVE]="CDL" OR OPER$[CARVE]="CUL" THEN GOSUB 6590
5880            IF BOX1=0 GOTO 6060
5890            BOX=BOX1
5900            IF OPER$[BOX1]="BLD" THEN GOSUB 6750:GOTO 5940
```

```
5910          IF OPER$[BOX1]="BLU" THEN GOSUB 6920:GOTO 5940
5920          IF OPER$[BOX1]="BRD" THEN GOSUB 7090:GOTO 5940
5930          IF OPER$[BOX1]="BRU" THEN GOSUB 7260
5940          IF BOX2=0 GOTO 6060
5950          FLAGFIT=0
5960          IF OPER$[BOX2]="BLD" THEN GOSUB 7430:GOTO 6000
5970          IF OPER$[BOX2]="BLU" THEN GOSUB 7710:GOTO 6000
5980          IF OPER$[BOX2]="BRD" THEN GOSUB 7990:GOTO 6000
5990          IF OPER$[BOX2]="BRU" THEN GOSUB 8270
6000          IF FLAGFIT=1 GOTO 6060
6010          BOX=BOX2
6020          IF OPER$[BOX2]="BLD" THEN GOSUB 6750:GOTO 6060
6030          IF OPER$[BOX2]="BLU" THEN GOSUB 6920:GOTO 6060
6040          IF OPER$[BOX2]="BRD" THEN GOSUB 7090:GOTO 6060
6050          IF OPER$[BOX2]="BRU" THEN GOSUB 7260
6060   NEXT I
6070   LOCATE 23,1
6080   INPUT "Would you like a list of coordinates printed [Y/N]";ANS$
6090   IF ANS$="Y" OR ANS$="y" THEN GOSUB 8580
6100 RETURN
6110 '>>> <<< CLU OR CRU
6120   DISTUP=AREAIN[CARVE]/AREAIN[I]*[LLI[I]-ULI[I]]
6130   LLI[CARVE]=LLI[I]
6140   LRI[CARVE]=LRI[I]
6150   LLI[I]=LLI[I]-DISTUP
6160   LRI[I]=LRI[I]-DISTUP
6170   ULI[CARVE]=LLI[I]
6180   URI[CARVE]=LRI[I]
6190   ULJ[CARVE]=ULJ[I]
6200   URJ[CARVE]=URJ[I]
6210   LLJ[CARVE]=LLJ[I]
6220   LRJ[CARVE]=LRJ[I]
6230   DRW=CARVE
6240   GOSUB 8550
6250   AREAIN[I]=AREAIN[I]-AREAIN[CARVE]
6260 RETURN
6270 '>>> <<< CLD OR CRD
6280   DISTDWN=AREAIN[CARVE]/AREAIN[I]*[LLI[I]-ULI[I]]
6290   ULI[CARVE]=ULI[I]
6300   URI[CARVE]=URI[I]
6310   ULI[I]=ULI[I]+DISTDWN
6320   URI[I]=URI[I]+DISTDWN
6330   LLI[CARVE]=ULI[I]
6340   LRI[CARVE]=URI[I]
6350   ULJ[CARVE]=ULJ[I]
6360   URJ[CARVE]=URJ[I]
6370   LLJ[CARVE]=LLJ[I]
6380   LRJ[CARVE]=LRJ[I]
6390   DRW=CARVE
6400   GOSUB 8550
6410   AREAIN[I]=AREAIN[I]-AREAIN[CARVE]
```

```
6420  RETURN
6430  '>>> <<< CDR OR CUR
6440    DISTRT=AREAIN[CARVE]/AREAIN[I]*[URJ[I]-ULJ[I]]
6450    ULJ[CARVE]=ULJ[I]
6460    LLJ[CARVE]=LLJ[I]
6470    ULJ[I]=ULJ[I]+DISTRT
6480    LLJ[I]=LLJ[I]+DISTRT
6490    URJ[CARVE]=ULJ[I]
6500    LRJ[CARVE]=LLJ[I]
6510    ULI[CARVE]=ULI[I]
6520    URI[CARVE]=URI[I]
6530    LLI[CARVE]=LLI[I]
6540    LRI[CARVE]=LRI[I]
6550    DRW=CARVE
6560    GOSUB 8550
6570    AREAIN[I]=AREAIN[I]-AREAIN[CARVE]
6580  RETURN
6590  '>>> <<< CDL OR CUL
6600    DISTLT=AREAIN[CARVE]/AREAIN[I]*[URJ[I]-ULJ[I]]
6610    URJ[CARVE]=URJ[I]
6620    LRJ[CARVE]=LRJ[I]
6630    URJ[I]=URJ[I]-DISTLT
6640    LRJ[I]=LRJ[I]-DISTLT
6650    ULJ[CARVE]=URJ[I]
6660    LLJ[CARVE]=LRJ[I]
6670    ULI[CARVE]=ULI[I]
6680    URI[CARVE]=URI[I]
6690    LLI[CARVE]=LLI[I]
6700    LRI[CARVE]=LRI[I]
6710    DRW=CARVE
6720    GOSUB 8550
6730    AREAIN[I]=AREAIN[I]-AREAIN[CARVE]
6740  RETURN
6750  '>>> <<< BLD
6760    DISTLT=[AREAIN[BOX]/AREAIN[I]]^.5*[URJ[I]-ULJ[I]]
6770    DISTDWN=[AREAIN[BOX]/AREAIN[I]]^.5*[LRI[I]-URI[I]]
6780    URI[BOX]=URI[I]
6790    URJ[BOX]=URJ[I]
6800    URI[I]=URI[I]+DISTDWN
6810    URJ[I]=URJ[I]-DISTLT
6820    LLI[BOX]=URI[I]
6830    LLJ[BOX]=URJ[I]
6840    ULI[BOX]=ULI[I]
6850    ULJ[BOX]=URJ[I]
6860    LRI[BOX]=URI[I]
6870    LRJ[BOX]=LRJ[I]
6880    DRW=BOX
6890    GOSUB 8550
6900    AREAIN[I]=AREAIN[I]-AREAIN[BOX]
6910  RETURN
6920  '>>> <<< BLU
```

```
6930    DISTLT=[AREAIN[BOX]/AREAIN[I]]^.5*[LRJ[I]-LLJ[I]]
6940    DISTUP=[AREAIN[BOX]/AREAIN[I]]^.5*[LRI[I]-URI[I]]
6950    LRI[BOX]=LRI[I]
6960    LRJ[BOX]=LRJ[I]
6970    LRI[I]=LRI[I]-DISTUP
6980    LRJ[I]=LRJ[I]-DISTLT
6990    ULI[BOX]=LRI[I]
7000    ULJ[BOX]=LRJ[I]
7010    LLI[BOX]=LLI[I]
7020    LLJ[BOX]=LRJ[I]
7030    URI[BOX]=LRI[I]
7040    URJ[BOX]=URJ[I]
7050    DRW=BOX
7060    GOSUB 8550
7070    AREAIN[I]=AREAIN[I]-AREAIN[BOX]
7080  RETURN
7090  '>>> <<< BRD
7100    DISTRT=[AREAIN[BOX]/AREAIN[I]]^.5*[URJ[I]-ULJ[I]]
7110    DISTDWN=[AREAIN[BOX]/AREAIN[I]]^.5*[LLI[I]-ULI[I]]
7120    ULI[BOX]=ULI[I]
7130    ULJ[BOX]=ULJ[I]
7140    ULI[I]=ULI[I]+DISTDWN
7150    ULJ[I]=ULJ[I]+DISTRT
7160    LRI[BOX]=ULI[I]
7170    LRJ[BOX]=ULJ[I]
7180    URI[BOX]=URI[I]
7190    URJ[BOX]=ULJ[I]
7200    LLI[BOX]=ULI[I]
7210    LLJ[BOX]=LLJ[I]
7220    DRW=BOX
7230    GOSUB 8550
7240    AREAIN[I]=AREAIN[I]-AREAIN[BOX]
7250  RETURN
7260  '>>> <<< BRU
7270    DISTRT=[AREAIN[BOX]/AREAIN[I]]^.5*[LRJ[I]-LLJ[I]]
7280    DISTUP=[AREAIN[BOX]/AREAIN[I]]^.5*[LLI[I]-ULI[I]]
7290    LLI[BOX]=LLI[I]
7300    LLJ[BOX]=LLJ[I]
7310    LLI[I]=LLI[I]-DISTUP
7320    LLJ[I]=LLJ[I]+DISTRT
7330    URI[BOX]=LLI[I]
7340    URJ[BOX]=LLJ[I]
7350    LRI[BOX]=LRI[I]
7360    LRJ[BOX]=LLJ[I]
7370    ULI[BOX]=LLI[I]
7380    ULJ[BOX]=ULJ[I]
7390    DRW=BOX
7400    GOSUB 8550
7410    AREAIN[I]=AREAIN[I]-AREAIN[BOX]
7420  RETURN
7430  '>>> <<< BLD CORRECTIONS
```

```
7440    IF OPER$[BOX1]="BLU" GOTO 7580  'IF NOT IT"S BRD
7450    LLJ[I]=ULJ[I]
7460    ULI[I]=URI[I]
7470    AREAIN[I]=[LRJ[I]-LLJ[I]]*[LLI[I]-ULI[I]]*AREATOT
7480    IF AREAIN[I]*.98<AREAIN[BOX2] THEN DRW=BOX1:GOSUB 8860:GOSUB 8700 ELSE
        GOTO 7700
7490    LLJ[I]=ULJ[I]
7500    URJ[BOX1]=ULJ[I]
7510    LLI[BOX1]=[AREAIN[BOX1]/AREATOT/[URJ[BOX1]-ULJ[BOX1]]]+ULI[BOX1]
7520    LRJ[BOX1]=URJ[BOX1]
7530    LRI[BOX1]=LLI[BOX1]
7540    DRW=BOX1
7550    GOSUB 8550
7560    LINE [INT[URJ[BOX1]*400]+60,INT[URI[BOX1]*150]+25] -
        [INT[URJ[I]*400]+60,INT[URI[I]*150]+25]
7570    GOTO 7700
7580    LRJ[I]=URJ[I]
7590    LLI[I]=LRI[I]
7600    AREAIN[I]=[LRJ[I]-LLJ[I]]*[LLI[I]-ULI[I]]*AREATOT
7610    IF AREAIN[I]*.98<AREAIN[BOX2] THEN DRW=BOX1:GOSUB 8860:GOSUB 8740 ELSE
        GOTO 7700
7620    LRI[I]=LLI[I]
7630    URI[BOX1]=LLI[I]
7640    LLJ[BOX1]=-[AREAIN[BOX1]/AREATOT/[LRI[BOX1]-URI[BOX1]]]+LRJ[BOX1]
7650    ULI[BOX1]=URI[BOX1]
7660    ULJ[BOX1]=LLJ[BOX1]
7670    DRW=BOX1
7680    GOSUB 8550
7690    LINE [INT[URJ[BOX1]*400]+60,INT[URI[BOX1]*150]+25] -
        [INT[URJ[I]*400]+60,INT[URI[I]*150]+25]
7700  RETURN
7710  '>>> <<< BLU CORRECTIONS
7720    IF OPER$[BOX1]="BRU" GOTO 7860  'IF NOT IT"S BLD
7730    URJ[I]=LRJ[I]
7740    ULI[I]=URI[I]
7750    AREAIN[I]=[LRJ[I]-LLJ[I]]*[LLI[I]-ULI[I]]*AREATOT
7760    IF AREAIN[I]*.98<AREAIN[BOX2] THEN DRW=BOX1:GOSUB 8860:GOSUB 8820 ELSE
        GOTO 7980
7770    URI[I]=ULI[I]
7780    LRI[BOX1]=ULI[I]
7790    ULJ[BOX1]=-[AREAIN[BOX1]/AREATOT/[LRI[BOX1]-URI[BOX1]]]+URJ[BOX1]
7800    LLI[BOX1]=LRI[BOX1]
7810    LLJ[BOX1]=ULJ[BOX1]
7820    DRW=BOX1
7830    GOSUB 8550
7840    LINE [INT[LRJ[BOX1]*400]+60,INT[LRI[BOX1]*150]+25] -
        [INT[LRJ[I]*400]+60,INT[LRI[I]*150]+25]
7850    GOTO 7980
7860    ULJ[I]=LLJ[I]
7870    LLI[I]=LRI[I]
7880    AREAIN[I]=[LRJ[I]-LLJ[I]]*[LLI[I]-ULI[I]]*AREATOT
```

```
7890    IF AREAIN[I]*.98<AREAIN[BOX2] THEN DRW=BOX1:GOSUB 8860:GOSUB 8700 ELSE
        GOTO 7980
7900    LLJ[I]=ULJ[I]
7910    LRJ[BOX1]=ULJ[I]
7920    ULI[BOX1]=-[AREAIN[BOX1]/AREATOT/[LRJ[BOX1]-LLJ[BOX1]]]+LLI[BOX1]
7930    URJ[BOX1]=LRJ[BOX1]
7940    URI[BOX1]=ULI[BOX1]
7950    DRW=BOX1
7960    GOSUB 8550
7970    LINE [INT[LRJ[BOX1]*400]+60,INT[LRI[BOX1]*150]+25] -
        [INT[LRJ[I]*400]+60,INT[LRI[I]*150]+25]
7980  RETURN
7990  '>>> <<< BRD CORRECTIONS
8000    IF OPER$[BOX1]="BLD" GOTO 8140  'IF NOT IT"S BRU
8010    LLJ[I]=ULJ[I]
8020    LRI[I]=LLI[I]
8030    AREAIN[I]=[LRJ[I]-LLJ[I]]*[LLI[I]-ULI[I]]*AREATOT
8040    IF AREAIN[I]*.98<AREAIN[BOX2] THEN DRW=BOX1:GOSUB 8860:GOSUB 8740 ELSE
        GOTO 8260
8050    LRI[I]=LLI[I]
8060    ULI[BOX1]=LLI[I]
8070    LRJ[BOX1]=[AREAIN[BOX1]/AREATOT/[LLI[BOX1]-ULI[BOX1]]]+LLJ[BOX1]
8080    URI[BOX1]=ULI[BOX1]
8090    URJ[BOX1]=LRJ[BOX1]
8100    DRW=BOX1
8110    GOSUB 8550
8120    LINE [INT[ULJ[BOX1]*400]+60,INT[ULI[BOX1]*150]+25] -
        [INT[ULJ[I]*400]+60,INT[ULI[I]*150]+25]
8130    GOTO 8260
8140    LRJ[I]=URJ[I]
8150    URI[I]=ULI[I]
8160    AREAIN[I]=[LRJ[I]-LLJ[I]]*[LLI[I]-ULI[I]]*AREATOT
8170    IF AREAIN[I]*.98<AREAIN[BOX2] THEN DRW=BOX1:GOSUB 8860:GOSUB 8780 ELSE
        GOTO 8260
8180    LRJ[I]=URJ[I]
8190    ULJ[BOX1]=URJ[I]
8200    LRI[BOX1]=[AREAIN[BOX1]/AREATOT/[URJ[BOX1]-ULJ[BOX1]]]+URI[BOX1]
8210    LLJ[BOX1]=ULJ[BOX1]
8220    LLI[BOX1]=LRI[BOX1]
8230    DRW=BOX1
8240    GOSUB 8550
8250    LINE [INT[ULJ[BOX1]*400]+60,INT[ULI[BOX1]*150]+25] -
        [INT[ULJ[I]*400]+60,INT[ULI[I]*150]+25]
8260  RETURN
8270  '>>> <<< BRU CORRECTIONS
8280    IF OPER$[BOX1]="BRD" GOTO 8420  'IF NOT IT"S BLU
8290    URJ[I]=LRJ[I]
8300    LRI[I]=LLI[I]
8310    AREAIN[I]=[LRJ[I]-LLJ[I]]*[LLI[I]-ULI[I]]*AREATOT
8320    IF AREAIN[I]*.98<AREAIN[BOX2] THEN DRW=BOX1:GOSUB 8860:GOSUB 8780 ELSE
        GOTO 8540
```

```
8330    LRJ[I]=URJ[I]
8340    LLJ[BOX1]=URJ[I]
8350    URI[BOX1]=-[AREAIN[BOX1]/AREATOT/[LRJ[BOX1]-LLJ[BOX1]]]+LRI[BOX1]
8360    ULJ[BOX1]=LLJ[BOX1]
8370    ULI[BOX1]=URI[BOX1]
8380    DRW=BOX1
8390    GOSUB 8550
8400    LINE [INT[LLJ[BOX1]*400]+60,INT[LLI[BOX1]*150]+25] -
        [INT[LLJ[I]*400]+60,INT[LLI[I]*150]+25]
8410    GOTO 8540
8420    ULJ[I]=LLJ[I]
8430    URI[I]=ULI[I]
8440    AREAIN[I]=[LRJ[I]-LLJ[I]]*[LLI[I]-ULI[I]]*AREATOT
8450    IF AREAIN[I]*.98<AREAIN[BOX2] THEN DRW=BOX1:GOSUB 8860:GOSUB 8820 ELSE
        GOTO 8540
8460    URI[I]=ULI[I]
8470    LLI[BOX1]=ULI[I]
8480    URJ[BOX1]=[AREAIN[BOX1]/AREATOT/[LLI[BOX1]-ULI[BOX1]]]+ULJ[BOX1]
8490    LRI[BOX1]=LLI[BOX1]
8500    LRJ[BOX1]=URJ[BOX1]
8510    DRW=BOX1
8520    GOSUB 8550
8530    LINE [INT[LLJ[BOX1]*400]+60,INT[LLI[BOX1]*150]+25] -
        [INT[LLJ[I]*400]+60,INT[LLI[I]*150]+25]
8540  RETURN
8550  '>>> <<< BOX
8560    LINE[INT[ULJ[DRW]*400]+60,INT[ULI[DRW]*150]+25] -
        [INT[LRJ[DRW]*400]+60,INT[LRI[DRW]*150]+25],,B
8570  RETURN
8580  '>>> <<< PRINT COORDINATES
8590    FOR I=2 TO FAC
8600            LPRINT ORDER[I]
8610            LPRINT USING "###.####";ULI[I];ULJ[I];
8620            LPRINT "      ";
8630            LPRINT USING "###.####";URI[I];URJ[I]
8640            LPRINT USING "###.####";LLI[I];LLJ[I];
8650            LPRINT "      ";
8660            LPRINT USING "###.####";LRI[I];LRJ[I]
8670            LPRINT
8680    NEXT I
8690  RETURN
8700  '>>> <<< PUSH LEFT
8710    ULJ[I]=ULJ[I]-.01
8720    AREAIN[I]=[URJ[I]-ULJ[I]]*[LLI[I]-ULI[I]]*AREATOT
8730    IF AREAIN[I]*.98<AREAIN[BOX2] THEN GOTO 8710:ELSE RETURN
8740  '>>> <<< PUSH DOWN
8750    LLI[I]=LLI[I]+.01
8760    AREAIN[I]=[URJ[I]-ULJ[I]]*[LLI[I]-ULI[I]]*AREATOT
8770    IF AREAIN[I]*.98<AREAIN[BOX2] THEN GOTO 8750:ELSE RETURN
8780  '>>> <<< PUSH RIGHT
8790    URJ[I]=URJ[I]+.01
```

```
8800   AREAIN[I]=(URJ[I]-ULJ[I])*(LLI[I]-ULI[I])*AREATOT
8810   IF AREAIN[I]*.98<AREAIN[BOX2] THEN GOTO 8790:ELSE RETURN
8820  '>>> <<< PUSH UP
8830   ULI[I]=ULI[I]-.01
8840   AREAIN[I]=(URJ[I]-ULJ[I])*(LLI[I]-ULI[I])*AREATOT
8850   IF AREAIN[I]*.98<AREAIN[BOX2] THEN GOTO 8830:ELSE RETURN
8860  '>>> <<< UNBOX
8870   LINE(INT(ULJ[DRW]*400)+60,INT(ULI[DRW]*150)+25) -
       (INT(LRJ[DRW]*400)+60,INT(LRI[DRW]*150)+25),0,B
8880  RETURN
```

APPENDIX C

OUTPUT FROM EXAMPLE I

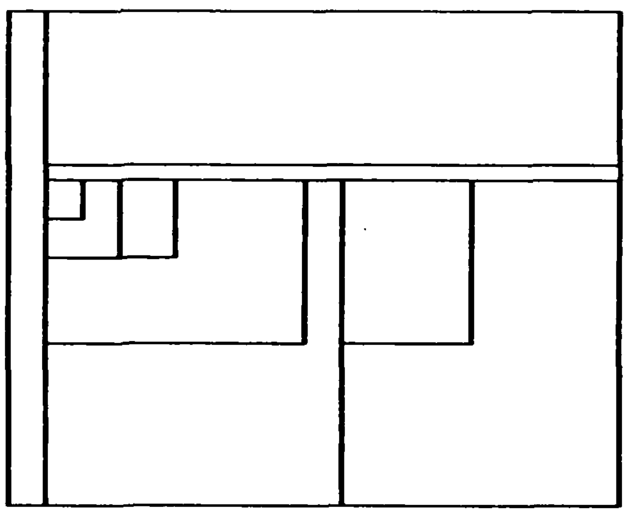
```
RUN
Random number seed [-32768 to 32767]? 1
You will need to input the filename for the data you want to use.
Would you like a list of files on the disk [Y/N]? N
Enter any filename with .DAT for an extension? FUCDPLAP

If you need an X value other than -1024 enter it at the prompt.
if not press return.?
NUMBER OF FACILITIES: 11
 1 :   1 A U E U U A A U U U
 2 :     2 U U I U U U U U E
 3 :       3 U O O U U U I O
 4 :         4 I U U U U A U
 5 :           5 I I O I I O
 6 :             6 E U I U U
 7 :               7 U E U U
 8 :                 8 U A U
 9 :                   9 E U
10 :                    10 U
11 :                      11


DELTAHEDRON INSERTION ORDER

 1  10  8  7  2  4  9  5  6  11  3

INSERTING VERTEX  2   IN TRIANGLE  1   8   7 010304

INSERTING VERTEX  4   IN TRIANGLE  1   10  7 010204

INSERTING VERTEX  9   IN TRIANGLE  10  8   7 020304

INSERTING VERTEX  5   IN TRIANGLE  10  7   9 020407

INSERTING VERTEX  6   IN TRIANGLE  5   7   9 080407

INSERTING VERTEX  11  IN TRIANGLE  2   8   7 050304

INSERTING VERTEX  3   IN TRIANGLE  10  7   5 020408

TOTAL DELTAHEDRON ADJACENCY SCORE IS 425

INCIDENCE MATRIX:

 1     A - E - - A A - U -
 2       - - - - U U - - E
 3         - O - U - - I -
 4           - - U - - A -
 5             I I - I I -
 6               E - I - -
 7                 U E U U
 8                   U A U
 9                     E -
10                       -
11
```

Example I Deltahedron Heuristic Output

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - 1
1 -10 - 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 - 1
1 -10 - 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 - 1
1 -10 - 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 - 1
1 -10 - 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 - 1
1 -10 - 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 - 1
1 -10 - 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 - 1
1 -10 - 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 - 1
1 -10 - 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 - 1
1 -10 - 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 - 1
1 -10 - 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 - 1
1 -10 - 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 - 1
1 -10 - 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 - 1
1 -10 - 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 - 1
1 -10 - 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 - 1
1 -10 - 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 - 1
1 -10 - 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 - 1
1 -10 - 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 - 1
1 -10 - 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 - 1
1 -10 - 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 - 1
1 -10 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - 1
1 -10 0 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 0 1
1 -10 - - - 0 - - - - 0 - - - - - - 0 - - - - - - - - 0 - - - - - - - 1
1 -10 - 3 - 5 - 6 6 - 9 9 9 9 9 9 - 8 -111111111111 - 2 2 2 2 2 2 2 - 1
1 -10 - 3 - 5 - 6 6 - 9 9 9 9 9 9 - 8 -111111111111 - 2 2 2 2 2 2 2 - 1
1 -10 - 3 - 5 - 6 6 - 9 9 9 9 9 9 - 8 -111111111111 - 2 2 2 2 2 2 2 - 1
1 -10 - 3 - 5 - 6 6 - 9 9 9 9 9 9 - 8 -111111111111 - 2 2 2 2 2 2 2 - 1
1 -10 - - - 5 - 6 6 - 9 9 9 9 9 9 - 8 -111111111111 - 2 2 2 2 2 2 2 - 1
1 -10 0 5 5 5 - 6 6 - 9 9 9 9 9 9 - 8 -111111111111 - 2 2 2 2 2 2 2 - 1
1 -10 - 5 5 5 - 6 6 - 9 9 9 9 9 9 - 8 -111111111111 - 2 2 2 2 2 2 2 - 1
1 -10 - 5 5 5 - 6 6 - 9 9 9 9 9 9 - 8 -111111111111 - 2 2 2 2 2 2 2 - 1
1 -10 - 5 5 5 - 6 6 - 9 9 9 9 9 9 - 8 -111111111111 - 2 2 2 2 2 2 2 - 1
1 -10 - - - 0 - - - - 9 9 9 9 9 9 - 8 -111111111111 - 2 2 2 2 2 2 2 - 1
1 -10 0 9 9 9 9 9 9 9 9 9 9 9 9 9 - 8 -111111111111 - 2 2 2 2 2 2 2 - 1
1 -10 - 9 9 9 9 9 9 9 9 9 9 9 9 9 - 8 -111111111111 - 2 2 2 2 2 2 2 - 1
1 -10 - 9 9 9 9 9 9 9 9 9 9 9 9 9 - 8 -111111111111 - 2 2 2 2 2 2 2 - 1
1 -10 - 9 9 9 9 9 9 9 9 9 9 9 9 9 - 8 -111111111111 - 2 2 2 2 2 2 2 - 1
1 -10 - 9 9 9 9 9 9 9 9 9 9 9 9 9 - 8 -111111111111 - 2 2 2 2 2 2 2 - 1
1 -10 - 9 9 9 9 9 9 9 9 9 9 9 9 9 - 8 -111111111111 - 2 2 2 2 2 2 2 - 1
1 -10 - 9 9 9 9 9 9 9 9 9 9 9 9 9 - 8 -111111111111 - 2 2 2 2 2 2 2 - 1
1 -10 - 9 9 9 9 9 9 9 9 9 9 9 9 9 - 8 -111111111111 - 2 2 2 2 2 2 2 - 1
1 -10 - 9 9 9 9 9 9 9 9 9 9 9 9 9 - 8 -111111111111 - 2 2 2 2 2 2 2 - 1
1 -10 - 9 9 9 9 9 9 9 9 9 9 9 9 9 - 8 -111111111111 - 2 2 2 2 2 2 2 - 1
1 -10 - - - - - - - - - - - - - - - 8 - - - - - - - - 2 2 2 2 2 2 2 - 1
1 -10 0 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 - 1
1 -10 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 2 2 2 2 2 2 2 2 2 2 2 2 2 2 - 1
1 -10 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 2 2 2 2 2 2 2 2 2 2 2 2 2 2 - 1
1 -10 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 2 2 2 2 2 2 2 2 2 2 2 2 2 2 - 1
1 -10 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 2 2 2 2 2 2 2 2 2 2 2 2 2 2 - 1
1 -10 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 2 2 2 2 2 2 2 2 2 2 2 2 2 2 - 1
1 -10 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 2 2 2 2 2 2 2 2 2 2 2 2 2 2 - 1
1 -10 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 2 2 2 2 2 2 2 2 2 2 2 2 2 2 - 1
1 -10 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 2 2 2 2 2 2 2 2 2 2 2 2 2 2 - 1
1 -10 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 2 2 2 2 2 2 2 2 2 2 2 2 2 2 - 1
1 -10 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 2 2 2 2 2 2 2 2 2 2 2 2 2 2 - 1
1 -10 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 2 2 2 2 2 2 2 2 2 2 2 2 2 2 - 1
1 -10 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 2 2 2 2 2 2 2 2 2 2 2 2 2 2 - 1
1 -10 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 2 2 2 2 2 2 2 2 2 2 2 2 2 2 - 1
1 -10 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 2 2 2 2 2 2 2 2 2 2 2 2 2 2 - 1
1 -10 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 2 2 2 2 2 2 2 2 2 2 2 2 2 2 - 1
1 -10 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 2 2 2 2 2 2 2 2 2 2 2 2 2 2 - 1
1 -10 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 2 2 2 2 2 2 2 2 2 2 2 2 2 2 - 1
1 -10 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 2 2 2 2 2 2 2 2 2 2 2 2 2 2 - 1
1 -10 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 2 2 2 2 2 2 2 2 2 2 2 2 2 2 - 1
1 - - - - - - - - - - - - - - - - - 0 - - - - - - - - - - - - - - - - 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Example I Condensed AS Matrix

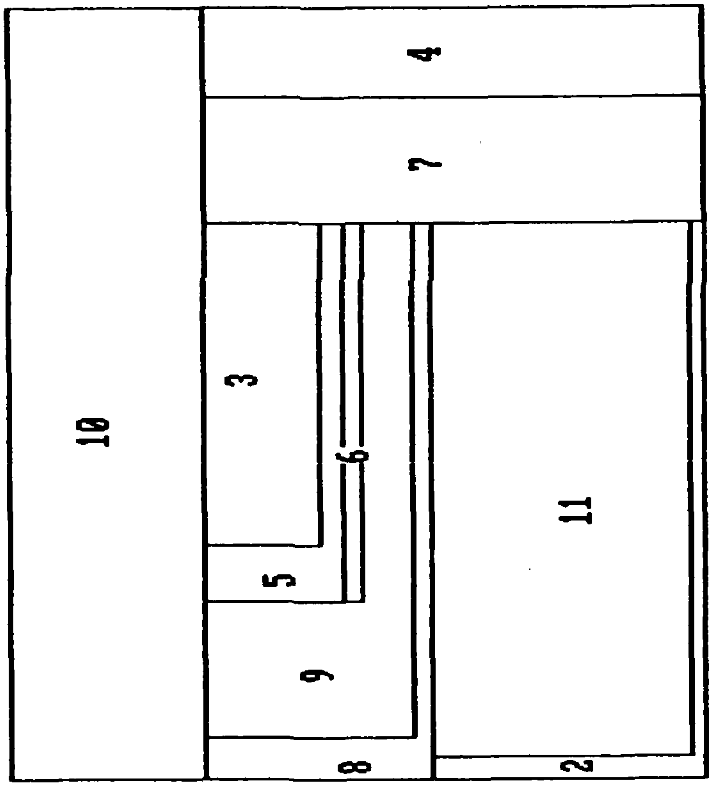
```
10 : 1400  8 : 170  7 : 570
2  120  CLU 010304  3
4  410  CDL 010204  4
9  450  BLD 020304  3
5  130  BLD 020407  7
6  60  CLU 040708  8
11  1250  BLD 030405  5
3  340  BLD 020408  8
```



Exomple I Screen Print of Duol

10 : 1400 8 : 170 7 : 570
2 120 CLU 010304 3
4 410 CDL 010204 4
9 450 BLD 020304 3
5 130 BLD 020407 7
6 60 CLU 040708 8
11 1250 BLD 030405 5
3 340 BLD 020408 8

Example I Insertion Information



Example I Screen Print of Block Plan

10

0.0000	0.0000	0.0000	1.0000
0.2857	0.0000	0.2857	1.0000

8

0.2857	0.0000	0.5866	0.0553
0.6117	0.0000	0.6117	0.7200

7

0.2857	0.7200	0.2857	0.8829
1.0000	0.7200	1.0000	0.8829

2

0.6117	0.0000	0.9826	0.0323
1.0000	0.0000	1.0000	0.7200

4

0.2857	0.8829	0.2857	1.0000
1.0000	0.8829	1.0000	1.0000

9

0.2857	0.0553	0.5070	0.2312
0.5866	0.0553	0.5866	0.7200

5

0.2857	0.2312	0.4526	0.3043
0.4820	0.2312	0.4820	0.7200

6

0.4820	0.2312	0.4820	0.7200
0.5070	0.2312	0.5070	0.7200

11

0.6117	0.0323	0.6117	0.7200
0.9826	0.0323	0.9826	0.7200

3

0.2857	0.3043	0.2857	0.7200
0.4526	0.3043	0.4526	0.7200

Example I Block Plan Coordinates

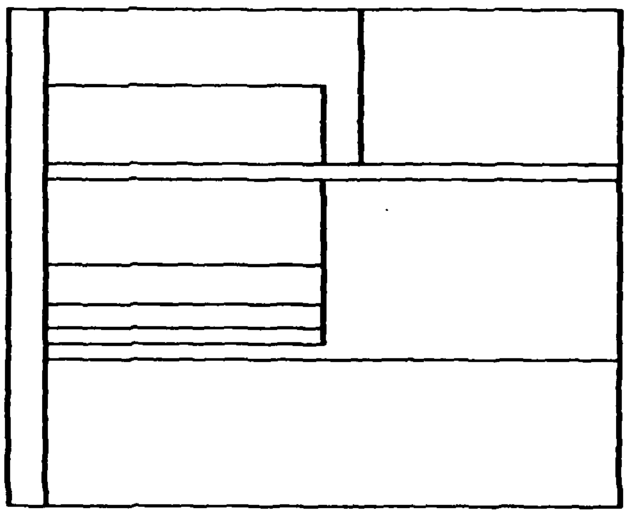
APPENDIX D

OUTPUT FROM EXAMPLE II

```
RUN
Random number seed (-32768 to 32767)? 2
You will need to input the filename for the data you want to use.
Would you like a list of files on the disk (Y/N)? N
Enter any filename with .DAT for an extension? 3-12R

If you need an X value other than -1024 enter it at the prompt,
if not press return.?
NUMBER OF FACILITIES: 12
 1 :   1 A U U U U U U U U A A
 2 :     2 A A O E U E I I A U
 3 :       3 O U O O O O U A U
 4 :         4 E E U O O O O U
 5 :           5 U E U U U U A
 6 :             6 U I O O U U
 7 :               7 O O O O E
 8 :                 8 O U O U
 9 :                   9 I O U
10 :                    10 U O
11 :                      11 X
12 :                        12

DELTAHEDRON INSERTION ORDER

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

INSERTING VERTEX  5   IN TRIANGLE  1  2  4 010204

INSERTING VERTEX  6   IN TRIANGLE  2  3  4 020304

INSERTING VERTEX  7   IN TRIANGLE  5  2  4 050204

INSERTING VERTEX  8   IN TRIANGLE  2  3  6 020306

INSERTING VERTEX  9   IN TRIANGLE  2  3  8 020308

INSERTING VERTEX  10  IN TRIANGLE  2  3  9 020309

INSERTING VERTEX  11  IN TRIANGLE  1  2  3 010203

INSERTING VERTEX  12  IN TRIANGLE  1  4  5 010405

TOTAL DELTAHEDRON ADJACENCY SCORE IS 614

INCIDENCE MATRIX:

 1     A U U U - - - - - A A
 2       A A O E U E I I A -
 3         O - O - O O U A -
 4           E E U - - - - U
 5             - E - - - - A
 6               - I - - - -
 7                 - - - - -
 8                   O - - -
 9                     I - -
 10                      - -
 11                        -
 12
```

Example II Deltahedron Heuristic Output

```
2 : 5000  3 : 800  4 : 150
5  400  CDL 010204  4
6  150  BLD 020304  3
7  350  BRD 020405  5
8  200  CDR 020306  6
9  350  CDR 020308  8
10  350  CDR 020309  9
11  200  CDR 010203  3
12  600  CRU 010405  5
```



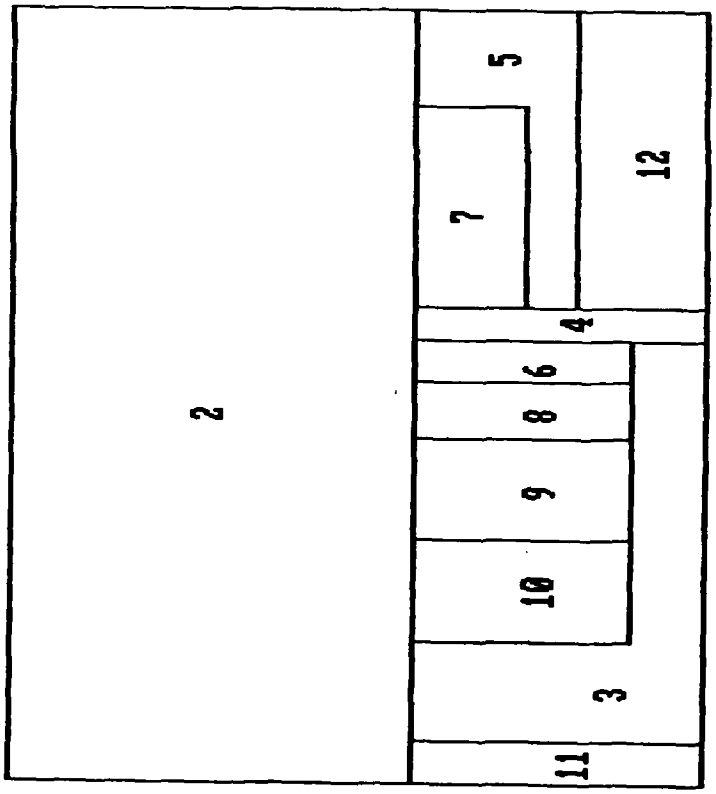
Example II Screen Print of Dual

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 - - - - - - - - - - - - - - - - - - 0 - -                           1
1 - 2 - 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 -12121212121212121212121212 - 1
1 - 2 - 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 -12121212121212121212121212 - 1
1 - 2 - 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 -12121212121212121212121212 - 1
1 - 2 - 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 -12121212121212121212121212 - 1
1 - 2 - 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 -12121212121212121212121212 - 1
1 - 2 - 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 -12121212121212121212121212 - 1
1 - 2 - 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 -12121212121212121212121212 - 1
1 - 2 - 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 -12121212121212121212121212 - 1
1 - 2 0 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 -12121212121212121212121212 - 1
1 - 2 - - - - - - - - - - - - - - - - 5 -12121212121212121212121212 - 1
1 - 2 - 7 7 7 7 7 7 7 7 7 7 7 7 7 7 - 5 -12121212121212121212121212 - 1
1 - 2 - 7 7 7 7 7 7 7 7 7 7 7 7 7 7 - 5 -12121212121212121212121212 - 1
1 - 2 - 7 7 7 7 7 7 7 7 7 7 7 7 7 7 - 5 -12121212121212121212121212 - 1
1 - 2 - 7 7 7 7 7 7 7 7 7 7 7 7 7 7 - 5 -12121212121212121212121212 - 1
1 - 2 - 7 7 7 7 7 7 7 7 7 7 7 7 7 7 - 5 -12121212121212121212121212 - 1
1 - 2 - 7 7 7 7 7 7 7 7 7 7 7 7 7 7 - 5 -12121212121212121212121212 - 1
1 - 2 - 7 7 7 7 7 7 7 7 7 7 7 7 7 7 - 5 -12121212121212121212121212 - 1
1 - 2 - 7 7 7 7 7 7 7 7 7 7 7 7 7 7 - 5 -12121212121212121212121212 - 1
1 - 2 - 7 7 7 7 7 7 7 7 7 7 7 7 7 7 - 5 -12121212121212121212121212 - 1
1 - 2 - - - - - - - - - - - - - - - - 0 - - - - - - - - - - - - - - 1
1 - 2 0 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 0 1
1 - 2 - - - - - - - - - - - - - - - - 0 - - - - - - - - - - - - - - 1
1 - 2 - 6 6 6 6 6 6 6 6 6 6 6 6 6 6 - 3 3 3 3 3 3 3 3 3 3 3 3 3 3 - 1
1 - 2 - 6 6 6 6 6 6 6 6 6 6 6 6 6 6 - 3 3 3 3 3 3 3 3 3 3 3 3 3 3 - 1
1 - 2 - 6 6 6 6 6 6 6 6 6 6 6 6 6 6 - 3 3 3 3 3 3 3 3 3 3 3 3 3 3 - 1
1 - 2 - 6 6 6 6 6 6 6 6 6 6 6 6 6 6 - 3 3 3 3 3 3 3 3 3 3 3 3 3 3 - 1
1 - 2 - 6 6 6 6 6 6 6 6 6 6 6 6 6 6 - 3 3 3 3 3 3 3 3 3 3 3 3 3 3 - 1
1 - 2 - 6 6 6 6 6 6 6 6 6 6 6 6 6 6 - 3 3 3 3 3 3 3 3 3 3 3 3 3 3 - 1
1 - 2 - 6 6 6 6 6 6 6 6 6 6 6 6 6 6 - 3 3 3 3 3 3 3 3 3 3 3 3 3 3 - 1
1 - 2 - 6 6 6 6 6 6 6 6 6 6 6 6 6 6 - 3 3 3 3 3 3 3 3 3 3 3 3 3 3 - 1
1 - 2 - 6 6 6 6 6 6 6 6 6 6 6 6 6 6 - 3 3 3 3 3 3 3 3 3 3 3 3 3 3 - 1
1 - 2 0 6 6 6 6 6 6 6 6 6 6 6 6 6 6 0 3 3 3 3 3 3 3 3 3 3 3 3 3 3 - 1
1 - 2 - - - - - - - - - - - - - - - - 3 3 3 3 3 3 3 3 3 3 3 3 3 3 - 1
1 - 2 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 3 3 3 3 3 3 3 3 3 3 3 3 3 3 - 1
1 - 2 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 3 3 3 3 3 3 3 3 3 3 3 3 3 3 - 1
1 - 2 - 8 8 8 8 8 8 8 8 8 8 8 8 8 8 - 3 3 3 3 3 3 3 3 3 3 3 3 3 3 - 1
1 - 2 0 8 8 8 8 8 8 8 8 8 8 8 8 8 8 0 3 3 3 3 3 3 3 3 3 3 3 3 3 3 - 1
1 - 2 - - - - - - - - - - - - - - - - 3 3 3 3 3 3 3 3 3 3 3 3 3 3 - 1
1 - 2 - 9 9 9 9 9 9 9 9 9 9 9 9 9 9 - 3 3 3 3 3 3 3 3 3 3 3 3 3 3 - 1
1 - 2 0 9 9 9 9 9 9 9 9 9 9 9 9 9 9 0 3 3 3 3 3 3 3 3 3 3 3 3 3 3 - 1
1 - 2 - - - - - - - - - - - - - - - - 3 3 3 3 3 3 3 3 3 3 3 3 3 3 - 1
1 - 2 -1010101010101010101010101010 - 3 3 3 3 3 3 3 3 3 3 3 3 3 3 - 1
1 - 2 - - - - - - - - - - - - - - - - 3 3 3 3 3 3 3 3 3 3 3 3 3 3 - 1
1 - 2 0 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 0 1
1 - 2 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - 1
1 - 2 -1111111111111111111111111111111111111111111111111111111111 - 1
1 - 2 -1111111111111111111111111111111111111111111111111111111111 - 1
1 - 2 -1111111111111111111111111111111111111111111111111111111111 - 1
1 - 2 -1111111111111111111111111111111111111111111111111111111111 - 1
1 - 2 -1111111111111111111111111111111111111111111111111111111111 - 1
1 - 2 -1111111111111111111111111111111111111111111111111111111111 - 1
1 - 2 -1111111111111111111111111111111111111111111111111111111111 - 1
1 - 2 -1111111111111111111111111111111111111111111111111111111111 - 1
1 - 2 -1111111111111111111111111111111111111111111111111111111111 - 1
1 - 2 -1111111111111111111111111111111111111111111111111111111111 - 1
1 - 2 -1111111111111111111111111111111111111111111111111111111111 - 1
1 - 2 -1111111111111111111111111111111111111111111111111111111111 - 1
1 - 2 -1111111111111111111111111111111111111111111111111111111111 - 1
1 - 2 -1111111111111111111111111111111111111111111111111111111111 - 1
1 - 2 -1111111111111111111111111111111111111111111111111111111111 - 1
1 - 2 -1111111111111111111111111111111111111111111111111111111111 - 1
1 - 2 -1111111111111111111111111111111111111111111111111111111111 - 1
1 - 2 -1111111111111111111111111111111111111111111111111111111111 - 1
1 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Example II Condensed AS Matrix

2 : 5000	3 : 800	4 : 150		
5	400	CDL	010204	4
6	150	BLD	020304	3
7	350	BRD	020405	5
8	200	CDR	020306	6
9	350	CDR	020308	8
10	350	CDR	020309	9
11	200	CDR	010203	3
12	600	CRU	010405	5

Example II Insertion Information



Example II Screen Print of Block Plan

2			
0.0000	0.0000	0.0000	1.0000
0.5848	0.0000	0.5848	1.0000
3			
0.5848	0.0563	0.8976	0.1849
1.0000	0.0563	1.0000	0.5775
4			
0.5848	0.5775	0.5848	0.6197
1.0000	0.5775	1.0000	0.6197
5			
0.7424	0.8795	0.5848	1.0000
0.8155	0.6197	0.8155	1.0000
6			
0.5848	0.5214	0.5848	0.5775
0.8976	0.5214	0.8976	0.5775
7			
0.5848	0.6197	0.5848	0.8795
0.7424	0.6197	0.7424	0.8795
8			
0.5848	0.4466	0.5848	0.5214
0.8976	0.4466	0.8976	0.5214
9			
0.5848	0.3157	0.5848	0.4466
0.8976	0.3157	0.8976	0.4466
10			
0.5848	0.1849	0.5848	0.3157
0.8976	0.1849	0.8976	0.3157
11			
0.5848	0.0000	0.5848	0.0563
1.0000	0.0000	1.0000	0.0563
12			
0.8155	0.6197	0.8155	1.0000
1.0000	0.6197	1.0000	1.0000

Example II Block Plan Coordinates

APPENDIX E

OUTPUT FROM EXAMPLE III

```
RUN
Random number seed [-32768 to 32767]? 1
You will need to input the filename for the data you want to use.
Would you like a list of files on the disk [Y/N]? n
Enter any filename with .DAT for an extension? FOULDS

If you need an X value other than -1024 enter it at the prompt,
if not press return.? -1
NUMBER OF FACILITIES: 22
 1 :    1 O O O O O O O O O O O A O O O O O A O O O
 2 :      2 O O O O O O O O O O O O O O O O O O O O
 3 :        3 E O O O O O O O O O O O O O O O O O O
 4 :          4 O O O O O O O O O O O O O O O O O O
 5 :            5 O O O O O O O O O O O O O O O O O
 6 :              6 E O O O O O O O O O O O O O O O
 7 :                7 O O O O O O O O O O O O O O O
 8 :                  8 I O O U X O O O O I I O A E
 9 :                    9 O O O X O O O O O A O I I
10 :                     10 O O O O O O O O O O E A
11 :                       11 O O O O O O O O O O O
12 :                         12 U I O O O O A E O O
13 :                           13 O U O U X O O O O
14 :                             14 I O O O O O O O
15 :                               15 O I I U U U U
16 :                                 16 O O O O O O
17 :                                   17 U U O O O
18 :                                     18 I O O E
19 :                                       19 I U U
20 :                                         20 X I
21 :                                           21 A
22 :                                             22

DELTAHEDRON INSERTION ORDER

 1  19  22  21  8  12  10  9  13  18  20  3  4  6  7  14  15  2  5  11  16  17

INSERTING VERTEX  8  IN TRIANGLE  19  22  21 020304

INSERTING VERTEX  12  IN TRIANGLE  1  19  21 010204

INSERTING VERTEX  10  IN TRIANGLE  8  22  21 050304

INSERTING VERTEX  9  IN TRIANGLE  19  21  8 020405

INSERTING VERTEX  13  IN TRIANGLE  1  22  21 010304

INSERTING VERTEX  18  IN TRIANGLE  19  22  8 020305

INSERTING VERTEX  20  IN TRIANGLE  1  19  12 010206

INSERTING VERTEX  3  IN TRIANGLE  8  21  10 050407

INSERTING VERTEX  4  IN TRIANGLE  8  21  3 050412

INSERTING VERTEX  6  IN TRIANGLE  8  10  3 050712
```

INSERTING VERTEX 7 IN TRIANGLE 6 10 3 140712

INSERTING VERTEX 14 IN TRIANGLE 20 19 12 110206

INSERTING VERTEX 15 IN TRIANGLE 18 22 8 100305

INSERTING VERTEX 2 IN TRIANGLE 8 10 6 050714

INSERTING VERTEX 5 IN TRIANGLE 8 6 2 051418

INSERTING VERTEX 11 IN TRIANGLE 1 21 13 010409

INSERTING VERTEX 16 IN TRIANGLE 8 3 4 051213

INSERTING VERTEX 17 IN TRIANGLE 15 22 8 170305

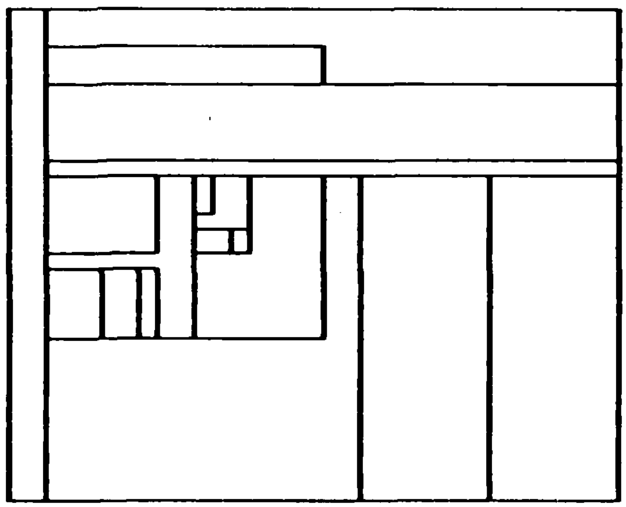
TOTAL DELTAHEDRON ADJACENCY SCORE IS 615

INCIDENCE MATRIX:

```
 1    - - - - - - - - - O O A - - - - - A O O O
 2      - - O O - O - O - - - - - - - - - - - -
 3        E - O O O - O - - - - - O - - - - O -
 4          - - - O - - - - - - - O - - - - O -
 5            O - O - - - - - - - - - - - - - -
 6              E O - O - - - - - - - - - - - -
 7                - - O - - - - - - - - - - - -
 8                  I O - - - - O O O I I - A E
 9                    - - - - - - - - - A - I -
10                      - - - - - - - - - - E A
11                        - O - - - - - - - O -
12                          - I - - - - A E O -
13                            - - - - - - - O O
14                              - - - - O O - -
15                                - I I - - - U
16                                  - - - - - -
17                                    - - - - O
18                                      I - - E
19                                        I U U
20                                          - -
21                                            A
22
```

E.ample III Deltahedron Heuristic Output

```
19 : 3000  22 : 2500  21 : 400
    8  1500  BLD 020304  3
    12  1000  CDL 010204  4
    10  2250  CLU 030405  5
    9  1800  BLD 020405  5
    13  3500  CLU 010304  3
    18  3360  BRD 020305  5
    20  3000  CDL 010206  6
    3  1525  BLD 040507  7
    4  1650  BLD 040512  12
    6  640  CDR 050712  12
    7  2000  CLU 071214  14
    14  7000  BRD 020611  11
    15  750  CRU 030510  10
    11  2200  CLU 010409  9
    17  1755  CRU 030517  17
```



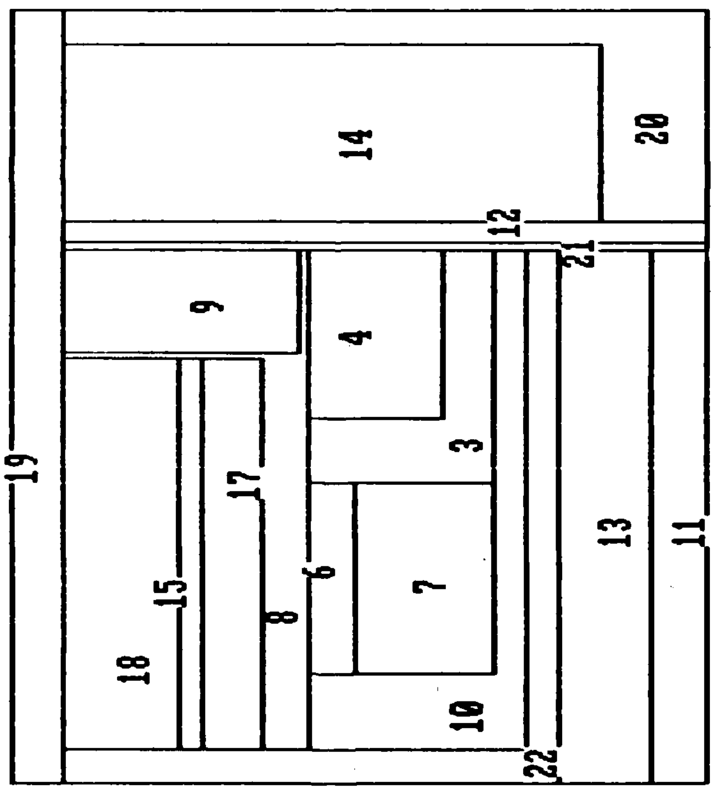
Excmple III Screen Print of Dual

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 - - - - - - - - - - - - - - - - - - - - - - -                           1
1 -19 -20202020202020202020202020202020202020202020202020202020202020202020 - 1
1 -19 -20202020202020202020202020202020202020202020202020202020202020202020 - 1
1 -19 -20202020202020202020202020202020202020202020202020202020202020202020 - 1
1 -19 020202020202020202020202020202020202020202020202020202020202020202020 - 1
1 -19 - - - - - - - - - - - - - - - - - -2020202020202020202020202020202020 - 1
1 -19 -14141414141414141414141414141414 -2020202020202020202020202020202020 - 1
1 -19 -14141414141414141414141414141414 -2020202020202020202020202020202020 - 1
1 -19 -14141414141414141414141414141414 -2020202020202020202020202020202020 - 1
1 -19 -14141414141414141414141414141414 -2020202020202020202020202020202020 - 1
1 -19 - - - - - - - - - - - - - - - - - - 0 - - - - - - - - - - - - - - - - 1
1 -19 012121212121212121212121212121212121212121212121212121212121212121212 0 1
1 -19 -12121212121212121212121212121212121212121212121212121212121212121212 - 1
1 -19 -12121212121212121212121212121212121212121212121212121212121212121212 - 1
1 -19 -12121212121212121212121212121212121212121212121212121212121212121212 - 1
1 -19 -12121212121212121212121212121212121212121212121212121212121212121212 - 1
1 -19 -12121212121212121212121212121212121212121212121212121212121212121212 - 1
1 -19 -12121212121212121212121212121212121212121212121212121212121212121212 - 1
1 -19 -12121212121212121212121212121212121212121212121212121212121212121212 - 1
1 -19 -12121212121212121212121212121212121212121212121212121212121212121212 - 1
1 -19 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - 1
1 -19 021212121212121212121212121212121212121212121212121212121212121212121 0 1
1 -19 - - - - - - - 0 - - 0 - 0 - - - 0 - - - - - - - 0 - - - - - - - - - 1
1 -19 - 9 9 9 9 9 - 8 - - 3 -101010 -22 -131313131313 -111111111111 - 1
1 -19 - 9 9 9 9 9 - 8 - - 3 -101010 -22 -131313131313 -111111111111 - 1
1 -19 - 9 9 9 9 9 - 8 - - 3 -101010 -22 -131313131313 -111111111111 - 1
1 -19 - 9 9 9 9 9 - 8 - - 3 -101010 -22 -131313131313 -111111111111 - 1
1 -19 - 9 9 9 9 9 - 8 - - 3 -101010 -22 -131313131313 -111111111111 - 1
1 -19 - 9 9 9 9 9 - 8 0 3 3 0101010 -22 -131313131313 -111111111111 - 1
1 -19 - 9 9 9 9 9 - 8 - 0 0 -101010 -22 -131313131313 -111111111111 - 1
1 -19 - 9 9 9 9 9 - 8 - 6 - -101010 -22 -131313131313 -111111111111 - 1
1 -19 - 9 9 9 9 9 - 8 - 6 - -101010 -22 -131313131313 -111111111111 - 1
1 -19 - - - - - - - 8 - 0 - -101010 -22 -131313131313 -111111111111 - 1
1 -19 0 8 8 8 8 8 8 8 0101010101010 -22 -131313131313 -111111111111 - 1
1 -19 - - 0 - 0 - - 8 -101010101010 -22 -131313131313 -111111111111 - 1
1 -19 -1818 -15 - - 8 -101010101010 -22 -131313131313 -111111111111 - 1
1 -19 -1818 -15 - - 8 -101010101010 -22 -131313131313 -111111111111 - 1
1 -19 -1818 -15 - - 8 -101010101010 -22 -131313131313 -111111111111 - 1
1 -19 -1818 -15 - - 8 -101010101010 -22 -131313131313 -111111111111 - 1
1 -19 -1818 -15 - - 8 -101010101010 -22 -131313131313 -111111111111 - 1
1 -19 -1818 -15 - - 8 -101010101010 -22 -131313131313 -111111111111 - 1
1 -19 -1818 -15 - - 8 -101010101010 -22 -131313131313 -111111111111 - 1
1 -19 -1818 -15 - - 8 -101010101010 -22 -131313131313 -111111111111 - 1
1 -19 - - 0 - 0 - - 0 - - - - - - - - -22 -131313131313 -111111111111 - 1
1 -19 0222222222222222222222222222222222 -131313131313 -111111111111 - 1
1 -19 -222222222222222222222222222222222 -131313131313 -111111111111 - 1
1 -19 -222222222222222222222222222222222 -131313131313 -111111111111 - 1
1 -19 -222222222222222222222222222222222 -131313131313 -111111111111 - 1
1 -19 -222222222222222222222222222222222 -131313131313 -111111111111 - 1
1 -19 -222222222222222222222222222222222 -131313131313 -111111111111 - 1
1 -19 -222222222222222222222222222222222 -131313131313 -111111111111 - 1
1 -19 -222222222222222222222222222222222 -131313131313 -111111111111 - 1
1 -19 -222222222222222222222222222222222 -131313131313 -111111111111 - 1
1 -19 -222222222222222222222222222222222 -131313131313 -111111111111 - 1
1 -19 -222222222222222222222222222222222 -131313131313 -111111111111 - 1
1 -19 -222222222222222222222222222222222 -131313131313 -111111111111 - 1
1 -19 -222222222222222222222222222222222 -131313131313 -111111111111 - 1
1 -19 -222222222222222222222222222222222 -131313131313 -111111111111 - 1
1 -19 -222222222222222222222222222222222 -131313131313 -111111111111 - 1
1 -19 -222222222222222222222222222222222 -131313131313 -111111111111 - 1
1 -19 -222222222222222222222222222222222 -131313131313 -111111111111 - 1
1 -19 -222222222222222222222222222222222 -131313131313 -111111111111 - 1
1 -19 -222222222222222222222222222222222 -131313131313 -111111111111 - 1
1 -19 -222222222222222222222222222222222 -131313131313 -111111111111 - 1
1 - - - - - - - - - - - - - - - - - - - 0 - - - - - - - 0 - - - - - - - - - 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Example III Condensed AS Matrix

```
19 : 3000  22 : 2500  21 : 400
8  1500  BLD 020304  3
12  1000  CDL 010204  4
10  2250  CLU 030405  5
9  1800  BLD 020405  5
13  3500  CLU 010304  3
18  3360  BRD 020305  5
20  3000  CDL 010206  6
3  1525  BLD 040507  7
4  1650  BLD 040512  12
6  640  CDR 050712  12
7  2000  CLU 071214  14
14  7000  BRD 020611  11
15  750  CRU 030510  10
2  1075  050714  0
5  1000  051418  0
11  2200  CLU 010409  9
16  400  051213  0
17  1755  CRU 030517  17
```

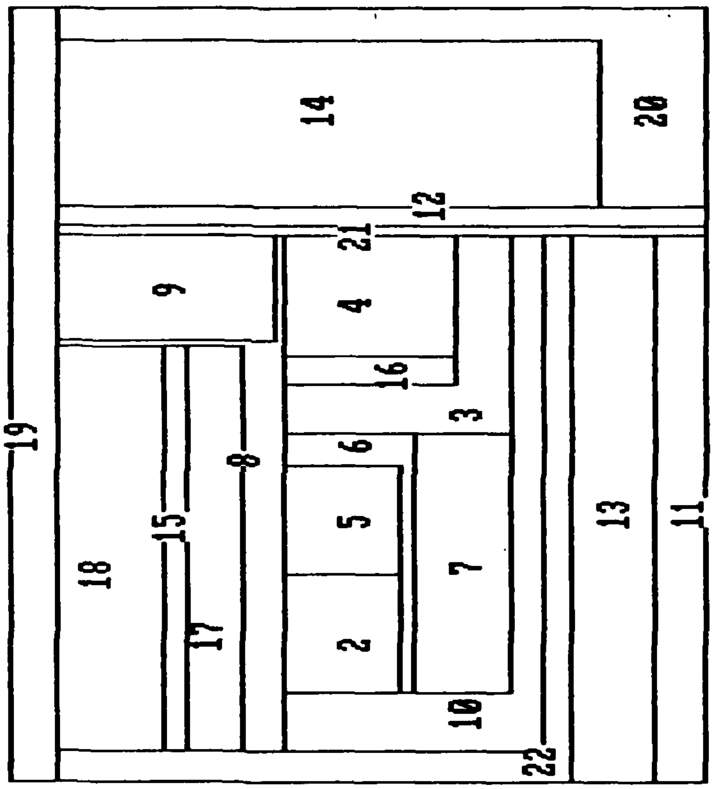
Example III Insertion Information



Example III Screen Print of Block Plan with three facilities not included

```
19
 0.0000  0.0000         0.0000  1.0000
 0.0753  0.0000         0.0753  1.0000

22
 0.0753  0.0000         0.7457  0.0452
 0.7927  0.0000         0.7927  0.6905

21
 0.0753  0.6905         0.0753  0.7013
 1.0000  0.6905         1.0000  0.7013

8
 0.0753  0.5514         0.4157  0.5577
 0.4319  0.5514         0.4319  0.6905

12
 0.0753  0.7013         0.0753  0.7285
 1.0000  0.7013         1.0000  0.7285

10
 0.4319  0.0452         0.6984  0.1426
 0.7457  0.0452         0.7457  0.6905

9
 0.0753  0.5577         0.0753  0.6905
 0.4157  0.5577         0.4157  0.6905

13
 0.7927  0.0000         0.7927  0.6905
 0.9200  0.0000         0.9200  0.6905

18
 0.0753  0.0452         0.0753  0.5514
 0.2420  0.0452         0.2420  0.5514

20
 0.8490  0.9557         0.0753  1.0000
 1.0000  0.7285         1.0000  1.0000

3
 0.4319  0.3913         0.6240  0.4748
 0.6984  0.3913         0.6984  0.6905

4
 0.4319  0.4748         0.4319  0.6905
 0.6240  0.4748         0.6240  0.6905

6
 0.4319  0.1426         0.4319  0.3913
 0.4965  0.1426         0.4965  0.3913

7
 0.4965  0.1426         0.4965  0.3913
 0.6984  0.1426         0.6984  0.3913

14
 0.0753  0.7285         0.0753  0.9557
 0.8490  0.7285         0.8490  0.9557

15
 0.2420  0.0452         0.2420  0.5514
 0.2792  0.0452         0.2792  0.5514

2
 0.0000  0.0000         0.0000  0.0000
 0.0000  0.0000         0.0000  0.0000

5
 0.0000  0.0000         0.0000  0.0000
 0.0000  0.0000         0.0000  0.0000

11
 0.9200  0.0000         0.9200  0.6905
 1.0000  0.0000         1.0000  0.6905

16
 0.0000  0.0000         0.0000  0.0000
 0.0000  0.0000         0.0000  0.0000

17
 0.2792  0.0452         0.2792  0.5514
 0.3662  0.0452         0.3662  0.5514
```

Example III Block Plan Coordinates with three Facilities not included



Example III Screen Print of Complete Block Plan

19				
	0.0000	0.0000	0.0000	1.0000
	0.0709	0.0000	0.0709	1.0000
22				
	0.0709	0.0000	0.7674	0.0412
	0.8102	0.0000	0.8102	0.7100
21				
	0.0709	0.7100	0.0709	0.7201
	1.0000	0.7100	1.0000	0.7201
8				
	0.0709	0.5662	0.3805	0.5725
	0.3948	0.5662	0.3948	0.7100
12				
	0.0709	0.7201	0.0709	0.7456
	1.0000	0.7201	1.0000	0.7456
10				
	0.3948	0.0412	0.7252	0.1168
	0.7674	0.0412	0.7674	0.7100
9				
	0.0709	0.5725	0.0709	0.7100
	0.3805	0.5725	0.3805	0.7100
13				
	0.8102	0.0000	0.8102	0.7100
	0.9268	0.0000	0.9268	0.7100
18				
	0.0709	0.0412	0.0709	0.5662
	0.2222	0.0412	0.2222	0.5662
20				
	0.8482	0.9584	0.0709	1.0000
	1.0000	0.7456	1.0000	1.0000
3				
	0.3948	0.4542	0.6450	0.5163
	0.7252	0.4542	0.7252	0.7100
4				
	0.3948	0.5541	0.3948	0.7100
	0.6450	0.5541	0.6450	0.7100
6				
	0.5611	0.4117	0.3948	0.4542
	0.5851	0.1168	0.5851	0.4542
7				
	0.5851	0.1168	0.5851	0.4542
	0.7252	0.1168	0.7252	0.4542
14				
	0.0709	0.7456	0.0709	0.9584
	0.8482	0.7456	0.8482	0.9584
15				
	0.2222	0.0412	0.2222	0.5662
	0.2560	0.0412	0.2560	0.5662
2				
	0.3948	0.1168	0.3948	0.2696
	0.5611	0.1168	0.5611	0.2696
5				
	0.3948	0.2696	0.3948	0.4117
	0.5611	0.2696	0.5611	0.4117
11				
	0.9268	0.0000	0.9268	0.7100
	1.0000	0.0000	1.0000	0.7100
16				
	0.3948	0.5163	0.3948	0.5541
	0.6450	0.5163	0.6450	0.5541
17				
	0.2560	0.0412	0.2560	0.5662
	0.3350	0.0412	0.3350	0.5662

Example III Complete Block Plan Coordinates

REFERENCES

Armour, G. C., and Buffa, E. S., A Heuristic Algorithm and Simulation Approach to Relative Location of Facilities, Management Science, Vol. 9, No. 1, 1963, pp.294-309.

Eades, P., Foulds, L. R., and Giffin, J. W., An Efficient Heuristic for Identifying a Maximum Weight Planar Subgraph, Combinatorial Mathematics IX, Lecture Notes in Mat. No. 952, [Springer-Verlag, Berlin], 1982, pp.239-251.

Euler, L., Demonstratio nonnullarum insignium proprietatum quibus solida hedris planis inclusa sunt praedita, Novi Comm. Acad. Sci. Imp. Petropol, Vol.4, 1752, pp.140-160 1752 [Opera Omnia, Vol. 26, pp. 94-108].

Foulds, L. R., Gibbons, P. B., and Giffin, J. W., Facilities Layout Adjacency Determination: An Experimental Comparison of Three Graph Theoretic Heuristics, Operations Research, Vol. 33, No. 5, 1985, pp. 1091-1106.

Foulds, L. R. and Robinson, D. F., Graph Theoretic Heuristics for the Plant Layout Problem, International Journal of Production Research, Vol. 16, 1978, pp. 27-37

Francis, R. L., and White, J. A., Facility Layout and Location: An Analytical Approach, Prentice-Hall Inc., Englewood, Ca., 1974.

Giffin, J. W., Graph Theoretic Techniques for Facilities Layout, Doctoral Dissertation, University of Canterbury, Christchurch, New Zealand, 1984

Giffin, J. W. and Foulds, L. R., A Graph-Theoretic Heuristic for Minimizing Total Transportation Cost in Facilities Layout, International Journal of Production Research, Vol. 23, No. 6, 1985, pp. 1247-1257.

Giffin, J. W. and Foulds, L. R., Improved Graph Theoretic Techniques for Facilities Layout, European Journal of Operations Research, [to appear], 1986.

Hopcroft, J. E. and Tarjan, R. E., Efficient Planarity Testing, J. A. C. M., Vol. 21, 1974, 544-568.

Koopmans, T. C., and Beckmann, M., Assignment problems and the location of economic facilities, Econometrica, Vol. 25, 1957, pp. 52-76.

Lee,R. C., and Moore, J. M., CORELAP-Computerized Relationship Layout Planning, Journal of Industrial Engineering, Vol. 18, No. 3, 1967, pp. 195-200.

Muther, R., Systematic Layout Planning, Industrial Education Institute, Boston, Mass., 1961.

Roth, J., Hashimshony, R., and Wachman, A., Turning a Graph into a Rectangular Floor Plan, Building and Environment, Vol. 17, No. 3, 1982, pp. 163-173.

Seehof, J. M., and Evans, W. O., Automated Layout Design Program, The Journal of Industrial Engineering, Vol. 18, No. 12, 1967, pp. 690-695.

Whitney, H., A Theorem on Graphs, Ann. Math., Vol. 32, 1931, pp. 378-390.

END

10-86

DTIC