

19. a) Give an algorithm that is more efficient than the exhaustive approach for the problem described in Exercise 18.  
 b) Provide a big- $O$  estimate for your algorithm to prove that the algorithm is more efficient than the algorithm of Exercise 18.
20. Determine the largest value of  $n$  for which all of the Hamilton circuits of  $K_n$  may be generated in less than 10 minutes of computer time, assuming the computer requires  $10^{-4}$  seconds of computer time to generate one Hamilton circuit and compute its weight.

21. How many spanning trees does the complete bipartite graph  $K_{2,n}$  have?
22. Let  $K_n - e$  be the graph obtained by deleting the edge  $e$  from  $K_n$ . Show that the number of spanning trees of  $K_n - e$ , for any edge  $e$ , is  $(n-2)n^{n-3}$ .
- \*23. Let  $K_n$  have vertex set  $V = \{v_1, v_2, \dots, v_n\}$ . Show that the number of spanning trees of  $K_n$  such that vertex  $v_i$  has degree  $d_i$  in the spanning tree is

$$\frac{(n-2)!}{(d_1-1)!(d_2-1)! \cdots (d_n-1)!}$$

- \*24. Describe a method which generates the set of all spanning trees of  $K_n$  such that vertex  $v_i$  has degree  $d_i$ .
- \*25. Let  $K_{m,n}$  be the complete bipartite graph with vertices  $V = V_1 \cup V_2$ , where  $V_1 = \{u_1, u_2, \dots, u_m\}$  and  $V_2 = \{v_1, v_2, \dots, v_n\}$ . Show that the number of spanning trees of  $K_{m,n}$  such that vertex  $u_i$  has degree  $d_i$  and vertex  $v_j$  has degree  $f_j$  is

$$\frac{(m-1)!(n-1)!}{(d_1-1)! \cdots (d_m-1)!(f_1-1)! \cdots (f_n-1)!}$$

## Computer Projects

- Let  $K_{10}$  have vertex set  $V = \{v_1, v_2, \dots, v_{10}\}$ . Write a computer program that takes as input the weights of the edges of  $K_{10}$  and finds a path of length 3 of smallest possible weight that joins a given pair of vertices.
- Write a program that generates all the Hamilton circuits of  $K_6$ .
- Write a computer program that generates all the perfect matches of  $K_8$ .

# Traveling Salesman Problem

**Author:** Arthur M. Hobbs, Department of Mathematics, Texas A&M University.

**Prerequisites:** The prerequisites for this chapter are graphs and trees. See, for example, Chapters 7 and 8 of *Discrete Mathematics and Its Applications*, Second Edition, by Kenneth H. Rosen.

## Introduction

In the traveling salesman problem, we are given a list of cities including our own, and we are asked to find a route using existing roads that visits each city exactly once, returns to our home city, and is as short as possible. However, it is useful to formalize the problem, thus allowing other problems to be interpreted in terms of the traveling salesman problem. Thus we have the following definition.

**Definition 1** Given a graph  $G$  in which the edges may be directed, undirected, or some of each, and in which a weight is assigned to each edge, the *traveling salesman problem*, denoted *TSP*, is the problem of finding a Hamilton circuit in  $G$  with minimum total weight, where the weight of a circuit is the sum of the weights of the edges in the circuit. Depending on the application, the weights on the edges will be called *lengths* or *costs*.  $\square$

Notice that the Hamilton circuit problem, to determine whether or not a given graph has a Hamilton circuit and to find one if it exists, is the special case of the TSP in which each of the edge weights is 1. Also, the feature that most distinguishes the TSP from the shortest path problem solved in Section 7.6 of *Discrete Mathematics and Its Applications*, Second Edition, by Rosen is the requirement in the TSP that every vertex of the graph must be included in the solution to the TSP.

We want to emphasize that the TSP calls for a Hamilton circuit, not a Hamilton path. Some applications would be more naturally stated in terms of Hamilton paths, but they can be translated into circuit problems (and this is done in the examples). For theoretical purposes, it is much better to have the symmetry that a Hamilton circuit allows, rather than having a pair of special vertices serving as the ends of a Hamilton path. Also, there is a forward-looking reason for favoring Hamilton circuits. The TSP is not solved, and any future complete or partial solutions of the TSP will be stated in terms of Hamilton circuits, rather than Hamilton paths. Thus it is better that our work now should be stated in terms of Hamilton circuits, so that future solutions can be immediately applied to it.

**Example 1** Solve the TSP by finding all Hamilton circuits in the graph  $G_1$  of Figure 1.

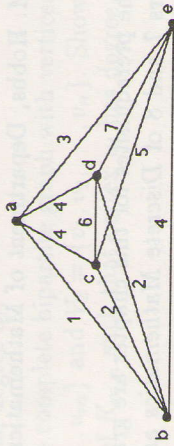


Figure 1. Graph  $G_1$ .

**Solution:** Since  $G_1$  is complete, any ordering of the vertices corresponds to a Hamilton circuit. By the nature of a circuit, we may suppose that all of the Hamilton circuits begin on the same vertex, e.g., vertex  $a$ . There are  $(5 - 1)! = 24$  permutations of the vertices starting with  $a$ . But each Hamilton circuit is described by two of these permutations (see Exercise 5). Thus, there are  $\frac{1}{2}(4!) = 12$  Hamilton circuits. These circuits and lengths are:

Permutation	Length	Permutation	Length
$a, b, c, d, e, a$	19	$a, d, e, c, b, a$	19
$a, b, d, c, e, a$	17	$a, b, d, e, c, a$	19
$a, d, c, e, b, a$	20	$a, b, e, d, c, a$	22
$a, c, b, d, e, a$	18	$a, c, b, e, d, a$	21
$a, c, d, b, e, a$	19	$a, c, e, b, d, a$	19
$a, d, b, c, e, a$	16	$a, d, c, b, e, a$	19

Thus the shortest Hamilton circuit, and hence the solution to the TSP in  $G_1$ , is given by the vertex sequence  $a, d, b, c, e, a$  with a total length of 16.  $\square$

A variant of this procedure can be used when not all of the possible edges are present or when some edges are directed and circuits are constrained to pass through them in only the given direction. We may examine all the permutations, casting out those which do not correspond to Hamilton circuits. This is useful if the graph is nearly complete. If many edges are not present, however, we may do a depth-first search on paths starting at  $a$ , looking for those that extend to Hamilton circuits. The following example illustrates the first possibility.

**Example 2** Solve the TSP for the graph of Figure 2.

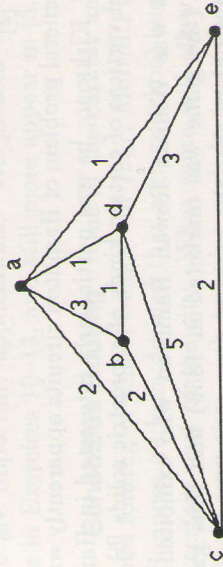


Figure 2. Graph  $G_2$ .

**Solution:** This graph has five vertices, but one edge of the complete graph is omitted. We again list all of the permutations of the vertices starting with  $a$ , but for some of the permutations, we note that the corresponding circuit does not exist in the graph:

Permutation	Length	Permutation	Length
$a, b, c, d, e, a$	14	$a, d, e, c, b, a$	11
$a, b, d, c, e, a$	12	$a, b, d, e, c, a$	11
$a, d, c, e, b, a$	does not exist	$a, b, e, d, c, a$	does not exist
$a, c, b, d, e, a$	9	$a, c, b, e, d, a$	does not exist
$a, c, d, b, e, a$	does not exist	$a, c, e, b, d, a$	does not exist
$a, d, b, c, e, a$	7	$a, d, c, b, e, a$	does not exist

Examining the six cases of sequences which do correspond to Hamilton circuits in  $G$ , we find the shortest is  $a, d, b, c, e, a$  with total length 7.  $\square$

These examples illustrate the most simple-minded algorithm for solving the traveling salesman problem: Just list all possible orderings of the vertices with one fixed beginning vertex, cast out orderings that fail to correspond to Hamilton circuits, and find the lengths of the rest, choosing the shortest. If there are  $n$  vertices, then there are  $\frac{1}{2}(n - 1)!$  orderings to examine. If no Hamilton circuit exists, the algorithm terminates, and the TSP has no solution in the graph. Otherwise, a shortest circuit is found, and its length is known. Because we must examine  $\frac{1}{2}(n - 1)!$  orderings of the vertices, this algorithm has complexity at least  $O((n - 1)!)$ ; such complexity is much worse than exponential complexity.

## History

The roots of the traveling salesman problem are in the studies of knight's tours in chess and of Hamilton circuits. A knight's tour is a sequence of moves taken by a knight on a chessboard, that begins and ends on a fixed square  $a$  and visits all other squares, each exactly once. This can be seen as a Hamilton circuit in a graph in which each square of the board is a vertex and two vertices are joined by an edge if and only if a knight's move connects the corresponding squares. A solution of the knight's tour problem was given by Leonhard Euler [2].

The more general problem of Hamilton circuits apparently was first studied in 1856 by T. P. Kirkman; he was particularly interested in Hamilton circuits on the edges and vertices of certain kinds of geometric solids [5].

However, it was William Rowan Hamilton who exhibited "The Icosian Game" in 1857 (the game was marketed in 1859) and thus gained so much publicity for the problem that the problem was named for him. "The Icosian Game" provided a 20-vertex graph drawn on a board and 20 numbered pegs to place at the vertices; the object was to place the pegs in the order of a Hamilton circuit through the vertices of the graph. As a game "The Icosian Game" failed, but as publicity for a mathematical problem, however unintentionally, it was very effective. (One of the sources of mathematical interest in this game was that it serves as a model of a non-commutative algebra ("the Icosian calculus") and thus can be viewed as part of "the origin of group theory" [1], an important part of modern mathematics.)

The traveling salesman problem appears to have been first described sometime in the 1930s.\* The problem became important during the late 1930s, just as the modern explosive growth of interest in combinatorics began. It was popularized by Merrill Flood of the RAND Corporation during the next two decades. The first important paper on the subject appeared in 1954 [1]; in it the authors George Dantzig, Ray Fulkerson, and Selmer Johnson of the RAND Corp. showed "that a certain tour of 49 cities, one in each of the [contiguous] 48 states and Washington, D. C., has the shortest road distance" [1]. The work was carried out using an ingenious combination of linear programming, graph theory, and map readings.

The study of the TSP has grown enormously since then; a monograph published in 1985 summarized the subject in 465 pages [6]. The literature on the problem is still growing rapidly.

\* This was perhaps done in a seminar conducted by Hassler Whitney in 1934 [1,3], although he did not remember the event [1].

## Applications

Even if there were no applications for a solution to the TSP, this problem would be important. It is an archetypical combinatorial problem: Other difficult combinatorial problems, such as the problem of finding the size of a smallest set  $S$  of vertices in a graph  $G$  such that every vertex in  $G$  is adjacent to a vertex in  $S$ , would be already solved if we could find a solution to the TSP [7]. There are literally hundreds of such problems [4], and their study has become a huge mathematical industry.

But there are important applications. We present three of them here. Others appear in Exercises 10–13, and still more appear in [6, Chapter 2].

**Example 3 Schoolbus Routing** One of the earliest applications of the TSP was to the routing of school busses. In 1937, Merrill Flood studied this problem [1]: Suppose we have decided that a given school bus will pick up the children in a certain part of the city. What route will allow the bus to visit each child's home just once each morning and do so as cheaply as possible? This is just a restatement of the TSP in terms of a school bus, with children's homes for vertices and roads for edges. Its solution may save a school district thousands of dollars per year.  $\square$

**Example 4 Electronics** In electrical circuit design, it is common for several components to be connected electrically to the same terminal (for example, to the ground terminal). Further, it is common for the components (memory chips, cpu's, sockets, etc.) to be placed before the wiring diagram is completed. For example, memory chips on a computer's motherboard are generally neatly aligned in rows and columns on a certain area of the board. Once such components are placed on the board, we have a subset of the pins of these components that must be electrically connected. Now, electricity flows easily in both directions through a wire, so a tree of wires will serve for the connections, the pins acting as vertices and the wires as edges. Because the pins on the components are small and because of the limited space available for printed wires, at most two printed wires can be connected to each pin. But then the degree of each vertex is at most two; thus the tree will be a path. Circuit boards are crowded with wires, so minimizing the total length of wire is necessary both to allow all of the wires to fit and to keep down the signal transfer time. Thus, in the complete graph on the pins to be connected together, we wish to find a Hamilton path of shortest length.

This problem can easily be converted into the form of a TSP. Add one new vertex to the graph, and join it to every other vertex by an edge of length 0. Then the Hamilton circuits in this augmented graph correspond one-to-one with Hamilton paths in the complete graph, and a Hamilton circuit in the augmented graph is shortest if and only if the corresponding Hamilton path in the complete graph is shortest. (For further information, see Section 2.2 of [6].)  $\square$

## Array Clustering

Our third example involves a much more subtle use of the traveling salesman problem. But we must develop the problem quite far before we can get to the TSP. (See Section 2.5 of [6] for a more detailed treatment.)

Suppose that both Joe's Garage and Melissa's Shop rebuild both engines and carburetors but do no upholstery work, while Sam's Repairs and Anne's Annex reupholster auto seats and replace headliners but do no engine work. If we let the first capital letter of each of these companies stand for the company, and let  $a$  stand for engine rebuilding,  $b$  for carburetor rebuilding,  $c$  for reupholstering auto seats, and  $d$  for replacing headliners, we can represent the activities of these companies in the form of a matrix whose entries are 0s and 1s: Each row corresponds to a company, and each column to a possible activity of the company. A 1 is in a given row and column if the company carries out that activity; otherwise a 0 appears there. Then the activities of these companies are represented by both matrices  $\mathbf{X}_1$  and  $\mathbf{X}_2$ , using different orderings of the rows and columns.

$$\mathbf{X}_1 = \begin{matrix} & \begin{matrix} a & b & c & d \end{matrix} \\ \begin{matrix} J \\ M \\ S \\ A \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

$$\mathbf{X}_2 = \begin{matrix} & \begin{matrix} a & c & b & d \end{matrix} \\ \begin{matrix} J \\ S \\ M \\ A \end{matrix} & \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \end{matrix}$$

Clearly the representation by matrix  $\mathbf{X}_1$  is better in that it groups the companies by their activities, or by industry. But matrix  $\mathbf{X}_2$  is obtained from  $\mathbf{X}_1$  by merely permuting the rows and columns of  $\mathbf{X}_1$ . If we are to use a matrix representation such as that shown here, our problem is that of ordering the companies and the activities they might engage in so that we get a matrix like  $\mathbf{X}_1$  instead of one like  $\mathbf{X}_2$ . This is not hard if there are only a few companies and activities involved, like we have here. But what if there are hundreds of each? That is the situation an economist studying the industries of the United States would face.

In general, let  $\mathbf{A} = [a_{ij}]$  be an  $m \times n$  matrix, where each row represents a company, each column represents a possible activity of a company,  $a_{ij} = 1$  if company  $i$  has experience carrying out activity  $j$ , and  $a_{ij} = 0$  if the company does not have such experience. The matrix  $\mathbf{A}$  is called a **relationship matrix**.

In a complex economy, we may set up a relationship matrix and yet find that, like matrix  $\mathbf{X}_2$ , it shows very little of the natural clusters, or industries,

that are present. Could we introduce a function which would give a value to such matrices and which would show, for example, that  $\mathbf{X}_1$  is better than  $\mathbf{X}_2$ ? The answer is "yes." In matrices  $\mathbf{X}_1$  and  $\mathbf{X}_2$ , notice that if we go along each of rows 1, 2, and 3, multiply each entry by the entry immediately below it, and add all the products, then in  $\mathbf{X}_1$  we get a sum of 4 while in  $\mathbf{X}_2$  we get a sum of 0. The reason we get a larger value from  $\mathbf{X}_1$  is that the 1s are bunched together there. The same thing happens if we go down columns 1, 2, and 3, multiply each entry by the entry to its immediate right, and add the products.

For a general 0, 1-matrix  $\mathbf{A} = [a_{ij}]$  with  $m$  rows and  $n$  columns, the observations of the previous paragraph lead us to the function  $f(\mathbf{A})$  given by

$$f(\mathbf{A}) = \sum_{i=1}^{m-1} \sum_{j=1}^n a_{ij} a_{i+1,j} + \sum_{j=1}^{n-1} \sum_{i=1}^m a_{ij} a_{i,j+1}.$$

Notice that  $f(\mathbf{X}_1) = [1 \cdot 1 + 1 \cdot 1 + 0 \cdot 0 + 0 \cdot 0 + 1 \cdot 0 + 1 \cdot 0 + 1 \cdot 0 + 1 \cdot 0 + 0 \cdot 0 + 1 \cdot 1 + 1 \cdot 1] + [1 \cdot 1 + 1 \cdot 1 + 0 \cdot 0 + 0 \cdot 0 + 1 \cdot 0 + 1 \cdot 0 + 1 \cdot 0 + 1 \cdot 0 + 0 \cdot 0 + 1 \cdot 1 + 1 \cdot 1] = 8$  while similarly  $f(\mathbf{X}_2) = 0$ . Further, if  $\mathcal{M}$  is the set of all matrices obtainable from a given matrix  $\mathbf{M}$  by a mixture of row and column permutations, then we want to find a matrix  $\mathbf{M}^*$  in  $\mathcal{M}$  such that  $f(\mathbf{M}^*) \geq f(\mathbf{M}')$  for every  $\mathbf{M}' \in \mathcal{M}$ . Note: There may be more than one  $\mathbf{M}^*$  that will suffice here. For example,  $f(\mathbf{X}_3) = 8 = f(\mathbf{X}_1)$  for the following matrix  $\mathbf{X}_3$ .

$$\mathbf{X}_3 = \begin{matrix} & \begin{matrix} a & b & c & d \end{matrix} \\ \begin{matrix} S \\ A \\ J \\ M \end{matrix} & \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

It is possible to show that the result of any mixture of row and column permutations of a matrix can be produced by doing a single row permutation followed by a single column permutation. Further, permuting the rows does not affect which column a given entry is in. Thus, for example, if

$$\mathbf{X} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \text{and} \quad \mathbf{X}' = \begin{pmatrix} c & d \\ a & b \end{pmatrix},$$

the interchange of rows 1 and 2 in going from  $\mathbf{X}$  to  $\mathbf{X}'$  leaves  $a$  and  $c$  in column 1 and  $b$  and  $d$  in column 2. Likewise, permuting the columns does not affect which row a given entry is in. Thus, if we can maximize each of

$$S_1(\mathbf{A}) = \sum_{i=1}^{m-1} \sum_{j=1}^n a_{ij} a_{i+1,j}$$

and

$$S_2(\mathbf{A}) = \sum_{j=1}^{n-1} \sum_{i=1}^m a_{ij} a_{i,j+1},$$

separately, we will maximize  $f(\mathbf{A})$ .

Now we come to the traveling salesman problem; we will use it as a tool, and we will use it twice. First, to maximize  $S_1$ , it suffices to minimize

$$-S_1 = \sum_{i=1}^{m-1} \sum_{j=1}^n -a_{ij} a_{i+1,j}.$$

(This is needed because the TSP asks for a minimum.) Now, given matrix  $\mathbf{A}$ , for each row  $i$  we introduce a vertex  $i$ . For any two rows  $k$  and  $l$ , we join them by an undirected edge with weight

$$c_{kl} = \sum_{j=1}^n -a_{kj} a_{lj}.$$

In the resulting undirected graph  $G'_3$ , each Hamilton path  $h$  describes a permutation of the rows of  $\mathbf{A}$ . Further, if  $\mathbf{A}' = [a'_{ij}]$  is formed from  $\mathbf{A}$  by carrying out this permutation of the rows for a minimum weight Hamilton path, then

$$\sum_{i=1}^{m-1} \sum_{j=1}^n -a'_{ij} a'_{i+1,j}$$

is precisely the sum of the weights along  $h$ . Since  $h$  is a minimum weight Hamilton path in  $G'_3$ , this means that

$$\sum_{i=1}^{m-1} \sum_{j=1}^n a'_{ij} a'_{i+1,j}$$

is largest among all possible orderings of the rows of  $\mathbf{A}$ .

Thus the maximum value of this half of  $f(\mathbf{A})$  is found by finding a minimum weight Hamilton path in  $G'_3$ . To convert this method to the TSP (for possible future solutions of the TSP as discussed before), add one more vertex 0 to  $G'_3$  and join 0 to each other vertex by an edge of weight 0, thus forming graph  $G_3$ . A solution of the TSP in  $G_3$  corresponds to a permutation of the rows of  $\mathbf{A}$  that maximizes  $S_1$ . A graph  $G_3$  for the matrix  $\mathbf{X}_2$  is shown in Figure 3, with a Hamilton circuit corresponding to the row ordering of matrix  $\mathbf{X}_1$  shown by the bold edges. This circuit is read in the order  $J, M, S, A$ .

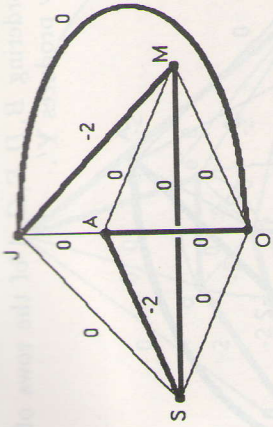


Figure 3. Graph  $G_3$  for the rows of matrix  $\mathbf{X}_2$ .

We can do a similar thing with  $S_2$ , using a graph  $G_4$  and starting with the columns of  $\mathbf{A}$  for vertices (see Exercise 6). A graph  $G_4$  for the matrix  $\mathbf{X}_2$  is shown in Figure 4, with a Hamilton circuit corresponding to the column ordering of matrix  $\mathbf{X}_1$  shown by the bold edges. This circuit is read in the order  $a, b, c, d$  to obtain  $\mathbf{X}_1$ .

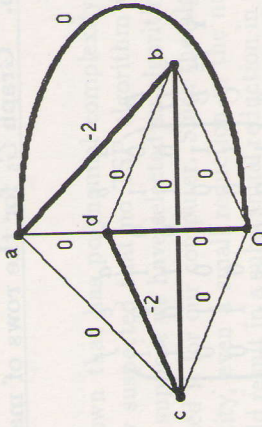


Figure 4. Graph  $G_4$  for the columns of matrix  $\mathbf{X}_2$ .

Thus the data array rearrangement problem becomes a pair of TSPs.

**Example 5** Suppose in polling companies, we obtained the  $5 \times 6$  relationship matrix  $\mathbf{Y}$  shown here.

$$\mathbf{Y} = \begin{matrix} & a & b & c & d & e & f \\ \begin{matrix} A \\ B \\ C \\ D \\ E \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \end{matrix}$$

Find clusters for the five companies.

**Solution:** Following the procedure just described, we form the graph  $G_5$  with vertices 0, A, B, C, D, and E, as shown in Figure 5. Solving the TSP there, we find the shortest Hamilton circuit shown by the bold edges in Figure 5. This

corresponds to the ordering  $B, D, E, C, A$  of the rows of  $Y$ . Reordering the rows of  $Y$  in this way produces  $Y'$ .

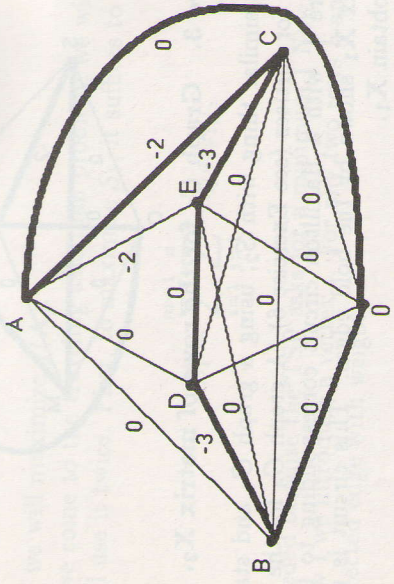


Figure 5. Graph  $G_5$  for the rows of matrix  $Y$ .

$$Y' = \begin{matrix} & a & b & c & d & e & f \\ B & 0 & 1 & 0 & 1 & 0 & 1 \\ D & 0 & 1 & 0 & 1 & 0 & 1 \\ E & 1 & 0 & 1 & 0 & 1 & 0 \\ C & 1 & 0 & 1 & 0 & 1 & 0 \\ A & 1 & 0 & 0 & 0 & 1 & 0 \end{matrix}$$

Next, we form the graph  $G_6$  with vertices  $0, a, b, c, d, e, f$ , as shown in Figure 6. Solving the TSP here, we find the shortest Hamilton circuit shown by the bold edges, corresponding to the ordering  $f, b, d, e, a, c$  of the columns of  $Y'$  as well as of  $Y$ . Carrying out this reordering, we get  $Y''$ .

$$Y'' = \begin{matrix} & f & b & d & e & a & c \\ B & 1 & 1 & 1 & 0 & 0 & 0 \\ D & 1 & 1 & 1 & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 1 & 1 & 1 \\ C & 0 & 0 & 0 & 1 & 1 & 1 \\ A & 0 & 0 & 0 & 1 & 1 & 0 \end{matrix}$$

Thus we see that companies  $B$  and  $D$  belong together in an industry carrying out activities  $b, d$ , and  $f$ , while companies  $A, C$ , and  $E$  belong in another industry which specializes in activities  $a, c$ , and  $e$  (although company  $A$  does not do activity  $c$ ). Note that  $f(Y) = 0$  while  $f(Y'') = 17$ .  $\square$

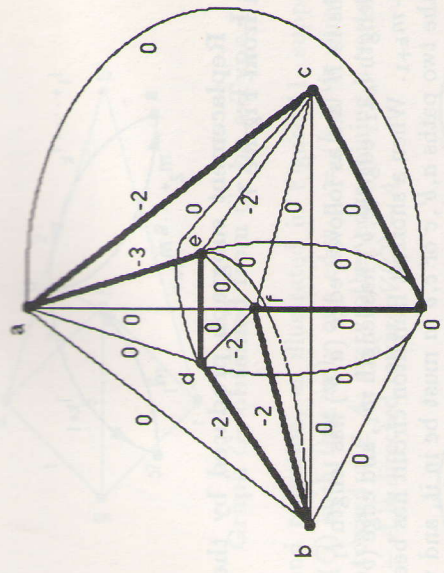


Figure 6. Graph  $G_6$  for the columns of matrix  $Y$ .

### Reductions

No algorithm is known that has polynomial complexity and solves the TSP. Worse, it is strongly suspected that no TSP algorithm with polynomial complexity exists. Now, suppose we have selected an algorithm for solving the TSP and have programmed it for our computer. Since the algorithm does not have polynomial complexity, even a small reduction in the number of vertices in a graph we give it could result in a substantial reduction in the time our computer requires to solve the TSP.

One reduction which is easy to make, although it is not commonly possible, occurs when the graph  $G$  contains a subgraph  $H$  as shown in Figure 7, where vertices  $a$  and  $c$  may be incident with more edges than those shown, but the vertices  $b_i$  with  $i \in \{1, 2, \dots, k\}$  meet only the edges shown in the figure.

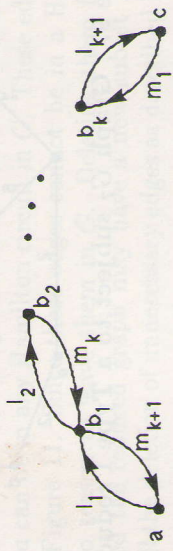


Figure 7. A subgraph subject to a Type I reduction.

Then any Hamilton circuit in  $G$  contains exactly one of the two paths  $a, b_1, b_2, \dots, b_{k-1}, b_k, c$  or  $c, b_k, b_{k-1}, \dots, b_2, b_1, a$ . Therefore, we can reduce the graph  $G$  to a graph  $G'$  by replacing the subgraph  $H$  with the subgraph  $H'$  shown in Figure 8, where the vertices  $a$  and  $c$  are as before, but the other vertices and the edges of  $H$  have been replaced by a single vertex  $b'$  and the four edges shown.

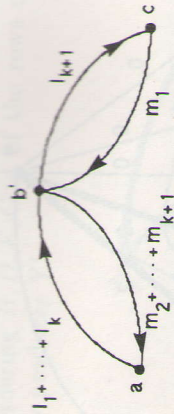


Figure 8. Replacement subgraph produced by the Type I reduction from Figure 7.

The edge lengths in  $H'$  are as follows: edge  $(a, b')$  has length  $l_1 + l_2 + \dots + l_k$ , edge  $(b', c)$  has length  $l_{k+1}$ , edge  $(c, b')$  has length  $m_1$ , and edge  $(b', a)$  has length  $m_2 + m_3 + \dots + m_{k+1}$ . When a shortest Hamiltonian circuit has been found in  $G'$ , exactly one of the two paths  $a, b', c$  or  $c, b', a$  must be in it, and we can replace that path by  $a, b_1, b_2, \dots, b_{k-1}, b_k, c$  or  $c, b_k, b_{k-1}, \dots, b_2, b_1, a$ , respectively, to obtain a shortest Hamiltonian circuit in  $G$ . Hereafter we will call the reduction of replacing  $H$  with  $H'$  a **Type I reduction**.

Type I reductions are also available in the undirected case, as illustrated in the following example.

**Example 6** Solve the TSP in the 8-vertex graph  $G_7$  shown in Figure 9.

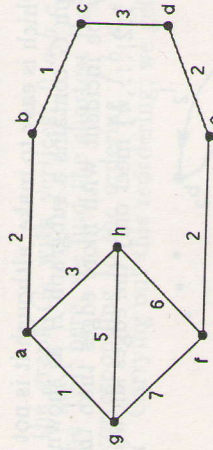


Figure 9. Graph  $G_7$  subject to a Type I reduction.

**Solution:** We can use a Type I reduction to replace the path  $a, b, c, d, e, f$  with the path  $a, b', f$  in which the edge  $\{a, b'\}$  is given weight 2 and the edge  $\{b', f\}$  is given weight  $1 + 3 + 2 + 2 = 8$ . The result  $G_8$  is shown in Figure 10. In  $G_8$ , it is easy to see that there are only two Hamiltonian circuits, namely  $a, b', f, g, h, a$  and  $a, b', f, h, g, a$ . Since the first of these has length 25 while the second one has length 22, the second is clearly the solution to the TSP in  $G_8$ . Returning to Figure 9,  $b'$  is replaced by  $b, c, d, e$  to give the shortest Hamiltonian circuit  $a, b, c, d, e, f, h, g, a$ , having length 22 in  $G_7$ .  $\square$

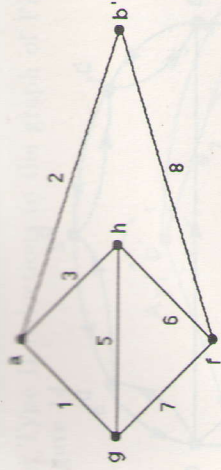


Figure 10. Graph  $G_8$  obtained from  $G_7$  by a Type I reduction.

For a second reduction, as illustrated in Figure 11, suppose there is a subset  $A$  of  $V(G)$  such that only one edge  $e = (b, a)$  is directed from  $b \in V(G) - A$  and toward  $a \in A$ ; every other edge joining a vertex in  $A$  with a vertex in  $V(G) - A$  is directed toward  $V(G) - A$ .

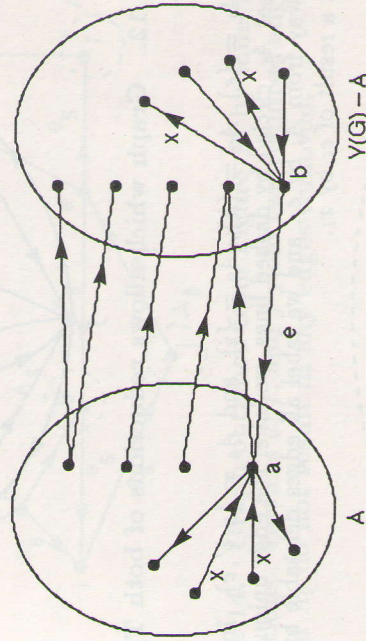


Figure 11. Edge  $e$  must be in any Hamiltonian circuit of  $G$ .

Then any Hamiltonian circuit in  $G$  must include edge  $e$  since the Hamiltonian circuit must cross from  $V(G) - A$  to  $A$  and only  $e$  is available. Hence no edge incident with  $b$  and directed away from  $b$  and no edge incident with  $a$  and directed toward  $a$  can be in any Hamiltonian circuit in  $G$ . These edges are marked with an "x" in Figure 11. Since these edges cannot be in a Hamiltonian circuit in  $G$ , there is no reason to leave them in  $G$ . Deleting them from the graph reduces it; indeed the reduced graph may have a new subset  $A'$  of the same type. We will call the deletion of unnecessary edges as described here a **Type II reduction**.

**Example 7** Use reductions of Types I and II to find a Hamiltonian circuit in the graph of Figure 12.

**Solution:** In Figure 12, we find that we can replace vertices  $b_1, b_2$ , and  $b_3$  with the single vertex  $b'$  by a Type I reduction, obtaining the graph shown in Figure 13. In Figure 13, we notice four occurrences of a set  $A$  suited to a Type II

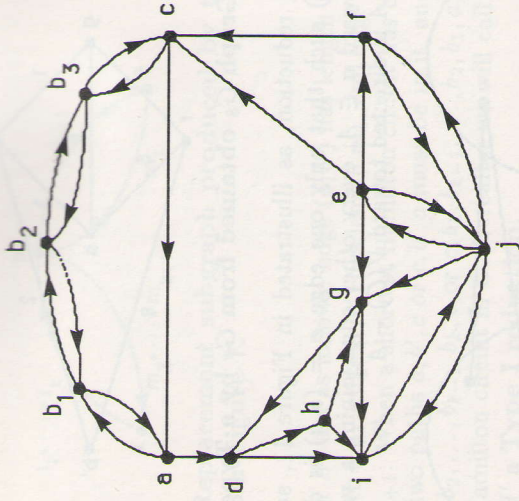


Figure 12. Graph which allows reductions of both Types I and II.

reduction:  $A_1 = \{e\}$ ,  $A_2 = \{h\}$ ,  $A_3 = \{i\}$ , and  $A_4 = \{a, b', c\}$ . In Figures 13 and 14, for set  $A_i$ , circled by dashed lines, we label the edge uniquely directed toward or away from  $A_i$  by  $e_i$ , and we label all edges deletable by a Type II reduction as a result of  $e_i$  by  $x_i$ .

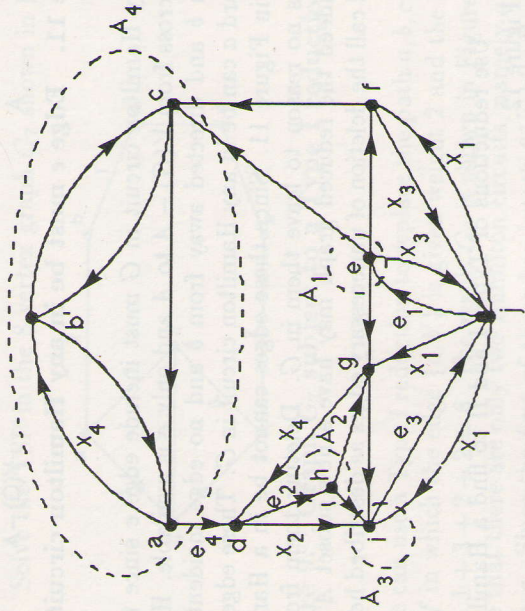


Figure 13. Graph obtained from Figure 12 by one Type I reduction.

After applying Type II reductions to the graph of Figure 13, we get the graph shown in Figure 14.

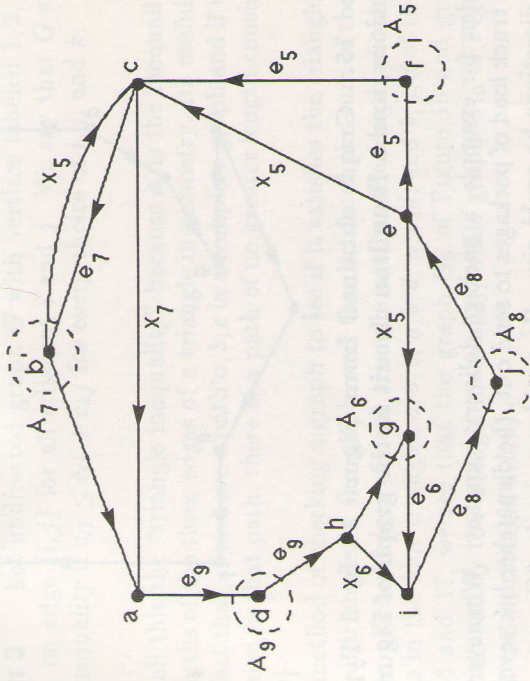


Figure 14. Graph obtained from Figure 13 by four Type II reductions. Vertices  $d$ ,  $f$ , and  $j$  yield two required edges, but  $A_8$  and  $A_9$  do not result in any edges removed.

There we see three new sets,  $A_5 = \{f\}$ ,  $A_6 = \{g\}$ , and  $A_7 = \{b'\}$ . Each of these leads to the deletion of one or more further edges, resulting in the unique Hamilton circuit of Figure 15.

It follows that the graph of Figure 12 has the unique Hamilton circuit

$$a, d, h, g, i, j, e, f, c, b_3, b_2, b_1, a,$$

which is thus the solution of the TSP for this graph no matter what weights are placed on the edges.  $\square$

### Approximation Algorithms

Reductions do not always exist in a graph, and even when they do, they may not reduce the problem enough to make it solvable in a reasonable amount of time. When a solution must be found in spite of this problem, we may decide that a good approximation to the minimum solution will suffice.



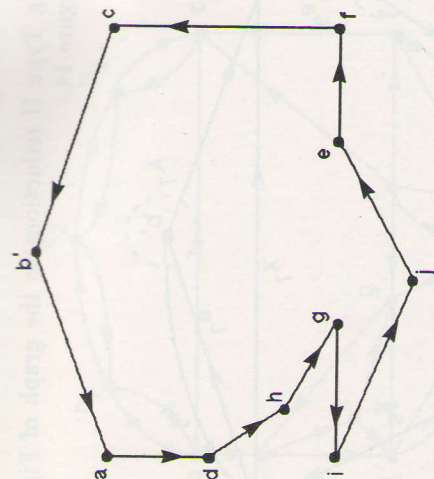


Figure 15. Graph obtained from Figure 14 by 3 Type II reductions, and a Hamilton circuit in the graph of Figure 13.

Consider, for example, a package delivery company. Whenever the company has a truck load of packages to send out, the dispatcher has several choices of routing directions he could give. He could say simply, "Go out and deliver these packages," leaving it to the driver of the truck to find the best route to follow. But the driver might not know the city well, and so he might waste much time and money going out of the way to make deliveries that could have been made more efficiently. The dispatcher might instead give the driver a prepared route to follow. But who will prepare the route, and how? The dispatcher, at a time when he is directing the movements of 50 other trucks as well? No, the company provides the dispatcher with a computer programmed to solve the TSP for each truck that is sent out.

But even now there is a problem. One truck might well have packages to deliver to more than 100 different addresses. Thus we seem to be asking the computer to solve the TSP 50 or more times per day on graphs with 100 or more vertices each. As said before, in general we do not know how to solve this problem in less than many years per graph. An acceptable compromise would be to program the computer to spend just a few seconds to find a good, though not necessarily the best, route. But what does the term "good" mean in this context? One answer is that, if a shortest route has length  $k$ , a route is good if its length is at most  $2k$ , or some other small constant times  $k$ . Formally, we say we have a performance guarantee in this case.

**Definition 2** A performance guarantee for an algorithm  $A$  is a theorem stating that there is a constant  $c$  such that, given a graph  $G$  with weighted edges and having a minimum Hamilton circuit of length  $k$ , the algorithm  $A$  will find a Hamilton circuit of length at most  $ck$ .  $\square$

We have such an approximation algorithm in the case of graphs that satisfy the triangle inequality.

**Definition 3** Let undirected graph  $G$  with vertices labeled  $1, 2, \dots, n$  have weight  $c_{ij}$  on edge  $\{i, j\}$  for all adjacent  $i$  and  $j$ . We say that  $G$  satisfies the triangle inequality if  $c_{ij} \leq c_{ik} + c_{kj}$  for every choice of  $i, j$ , and  $k$ .  $\square$

We call this the "triangle inequality" because it is the inequality satisfied by the lengths of the three edges of a triangle in geometry. Its usefulness comes from the fact that, if we have a path  $a, b, c$  in a complete graph and if the vertex  $b$  is not needed in that path, there is a path of no greater length consisting of the edge  $\{a, c\}$ .

One method of checking a graph to see if it satisfies the triangle inequality is to find all triangles in the graph, and then to check the sum of the lengths of each of the pairs of edges of each triangle against the length of the third side. Doing this in the four triangles  $a, b, c$ ;  $a, b, d$ ;  $a, c, d$ ; and  $b, c, d$  of the graphs in Figures 16 and 17, we see that the graph  $G_9$  of Figure 16 and graph  $G_{11}$  of Figure 17 do satisfy the triangle inequality, while graph  $G_{10}$  of Figure 17 does not ( $1 + 2 < 6$  in triangle  $a, b, c$ ).

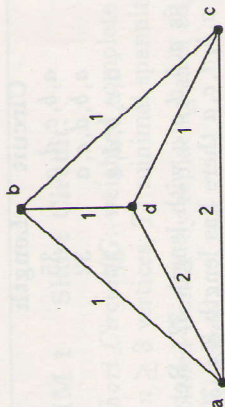
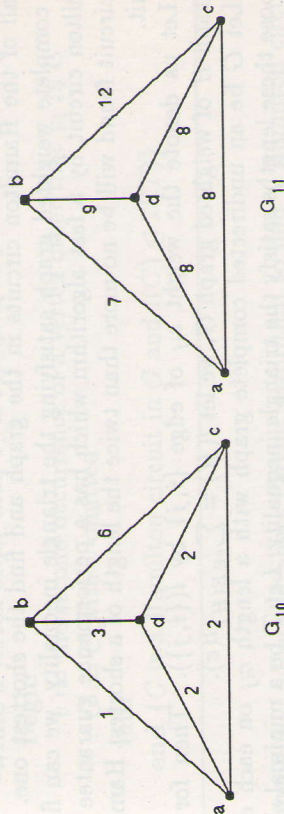
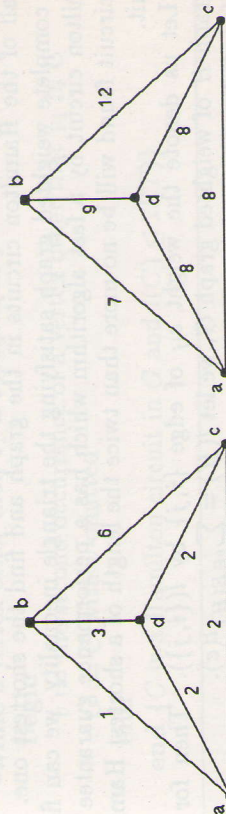


Figure 16. Graph  $G_9$  satisfies the triangle inequality.



$G_{10}$

Figure 17. Graph  $G_{10}$  does not satisfy the triangle inequality. Graph  $G_{11}$ , obtained from  $G_{10}$  by adding 6 to the weight of each edge, does satisfy the triangle inequality.



$G_{11}$

If  $G$  is a graph with finite nonnegative lengths  $c_{ij}$  on each edge  $\{i, j\}$ , then we can form a related graph  $G'$  which satisfies the triangle inequality. To do this, first let  $M$  be a constant such that  $M \geq c_{ij}$  for all  $i$  and  $j$ . Then add  $M$  to every edge length in  $G$  to form  $G'$ . The triangle inequality is satisfied in  $G'$  since

$$c'_{ij} + M \leq M + M \leq c'_{ik} + M + c'_{kj} + M$$

for every choice of  $i, j, k$ , and  $M$ . Further, not only does  $G'$  satisfy the triangle inequality, but we do not have to find many triangles and verify the inequality on each to know that  $G'$  satisfies it.

But the process described in the previous paragraph increases the length of every Hamilton circuit of  $G$  by exactly  $nM$ , where  $n = |V(G)|$ , so a shortest Hamilton circuit in  $G'$  has the same edges in the same order as a shortest Hamilton circuit in  $G$ . Hence solving the TSP in  $G'$  solves it in  $G$ .

**Example 8** For the graph  $G_{10}$  of Figure 17 we can let  $M = 6$ . Adding 6 to the length of each edge of  $G_{10}$  yields the graph  $G_{11}$ , thus converting a graph which does not satisfy the triangle inequality into one that does. The three Hamilton circuits of  $G_{11}$  and their lengths are

Circuit	Length
$a, b, c, d, a$	35
$a, b, d, c, a$	32
$a, c, b, d, a$	37

The shortest of these is  $a, b, d, c, a$  with length 32. Returning to  $G_{10}$ , we see that Hamilton circuit  $a, b, d, c, a$  there has length  $8 = 32 - 4(6)$  and it is easy to check that this is shortest.  $\square$

Of course, we have no need for the triangle inequality if we are going to list all of the Hamilton circuits in the graph and find the shortest one. But in a complete weighted graph satisfying the triangle inequality, we can find a Hamilton circuit by a fast algorithm which has a performance guarantee that the circuit found will be no more than twice the length of a shortest Hamilton circuit.

Let us denote the weight  $c_{ij}$  of edge  $\{i, j\}$  by  $l(\{i, j\})$ . Then for any subgraph  $H$  of weighted graph  $G$ , we let  $l(H) = \sum_{e \in E(H)} l(e)$ .

Let  $G$  be an undirected complete graph with a length  $c_{ij}$  on each edge. Suppose these lengths satisfy the triangle inequality. Let  $T$  be a minimal spanning tree of  $G$  (found by using Prim's algorithm, for example, as described in Section 8.6 of *Discrete Mathematics and Its Applications*, Second Edition, by Rosen). We will use  $T$  and the triangle inequality to find a Hamilton circuit  $C$  in  $G$  such that  $l(C) \leq 2l(T)$ .

In the algorithm *Short Circuit*, presented next, we begin with a circuit which includes all of the vertices and which has total length  $2l(T)$  because it includes each edge of  $T$  exactly twice. Listing this circuit as a sequence of vertices, one at a time we delete second occurrences  $v$  of vertices, replacing each with the edge from the vertex immediately before  $v$  to the vertex immediately after  $v$  in the sequence. Each time, the circuit length stays the same or is reduced because the graph satisfies the triangle inequality. For example, for the graph  $G_{12}$  of Figure 18, we start with the vertex sequence  $a, b, c, b, a$ . Noting the presence of a second occurrence of  $b$ , we replace it with the edge  $\{c, a\}$ , thus obtaining the vertex sequence  $a, b, c, a$  which describes a Hamilton circuit.

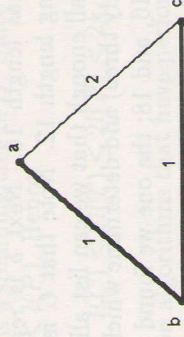


Figure 18. Graph  $G_{12}$ .  $T$  is indicated by bold edges.

#### ALGORITHM 1 Short Circuit.

```

procedure Short Circuit( $G$ : weighted complete undirected
graph with  $n \geq 3$  vertices;  $T$ : minimal spanning tree in  $G$ )
 $T'$  := graph formed from  $T$  by replacing each edge of  $T$  with
two parallel edges
 $v_1$  := vertex of degree 2 in  $T'$ 
 $C$  := the vertex sequence of an Euler circuit in  $T'$  beginning
at  $v_1$ 
while a vertex other than  $v_1$  is repeated in  $C$ 
begin
 $v$  := the second occurrence of a vertex other than  $v_1$  in  $C$ 
 $C$  :=  $C$  with  $v$  omitted
end
end { $C$  is a Hamilton circuit in  $G$  and  $l(C) \leq 2l(T)$ }

```

**Example 9** Use the Short Circuit algorithm to find a Hamilton circuit in the graph  $G_{13}$  of Figure 19.

**Solution:** Note that  $G_{13}$  satisfies the triangle inequality. The edges of a minimal spanning tree are drawn bold in Figure 19. The circuit  $C$  described

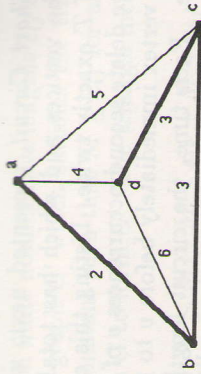


Figure 19. Graph  $G_{13}$ .  $T$  is indicated by bold edges.

in the algorithm is  $a, b, c, d, c, b, a$ , and  $l(C) = 2 + 3 + 3 + 3 + 2 = 16$ . In this circuit,  $c$  is the first vertex that is repeated, so the circuit  $C$  becomes  $a, b, c, d, b, a$ , and then has length 16. Next  $b$  is repeated, so the circuit  $C$  becomes  $a, b, c, d, a$ , having length 12. Note that  $C$  is now a Hamilton circuit in  $G_{13}$ . This graph is small enough that we can list all of the Hamilton circuits in the graph (there are only three) and determine which is shortest. These three circuits have lengths 12, 16, and 18; the one we found is shortest.  $\square$

**Example 10** Use the Short Circuit algorithm to find a Hamilton circuit in the graph  $G_1$  of Figure 1.

**Solution:** Since  $1 + 2 < 4$  in triangle  $a, b, c$ , this graph does not satisfy the triangle inequality. So we add  $M = 7$ , the largest edge length, to the length of each edge, obtaining the graph  $G_{14}$  of Figure 20. A minimal tree  $T$  is shown in Figure 20 with bold edges. Starting at  $d$ , we obtain  $C = d, b, c, b, a, e, a, b, d$ . Now  $b$  is the first repeated vertex, so we change  $C$  into  $d, b, c, a, e, a, b, d$ . Next  $a$  is the first repeated vertex, so  $C$  becomes  $d, b, c, a, e, b, d$ . Finally,  $b$  is repeated again, causing us to change  $C$  to  $d, b, c, a, e, d$ . This sequence corresponds to the same Hamilton circuit as the sequence  $a, c, b, d, e, a$  of Example 1, having length 18. While this is not the shortest Hamilton circuit in  $G_1$ , it is not bad and it was quickly found.  $\square$

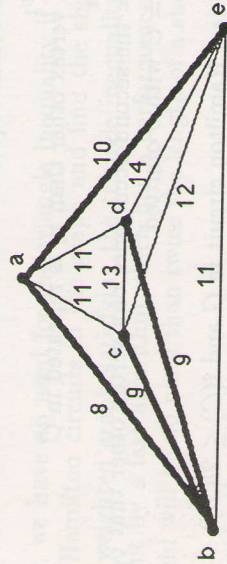


Figure 20. Graph  $G_{14}$  obtained from graph  $G_1$  of Figure 1 by adding 7 to the weight of each edge.

Notice in this last example that we did not use the new lengths shown in Figure 20; even  $T$  would use the same edges in  $G_1$  as it does in  $G_{14}$ . In fact, for the purposes of algorithm *Short Circuit*, it is enough to know that the new lengths can be found, without bothering to actually find them. The real reason

for the requirement that the lengths must satisfy the triangle inequality is to allow the proof of the next theorem, which is the performance guarantee for the algorithm *Short Circuit*.

**Theorem 1** Let  $G$  be a weighted complete undirected graph in which the triangle inequality holds and in which all weights are nonnegative. Let  $C$  be a circuit obtained by the algorithm *Short Circuit*. If the minimum length Hamilton circuit in  $G$  has length  $k$ , then  $l(C) \leq 2k$ .

**Proof:** Let  $C'$  be a shortest Hamilton circuit in  $G$ , and let  $e$  be an edge of  $C'$ . Let  $P = C' - e$ . Then  $P$  is a spanning tree of  $G$ . Since  $T$  is a minimum spanning tree of  $G$ , we have  $l(T) \leq l(P)$ . Hence

$$l(C) \leq 2l(T) \leq 2l(P) \leq 2k. \quad \blacksquare$$

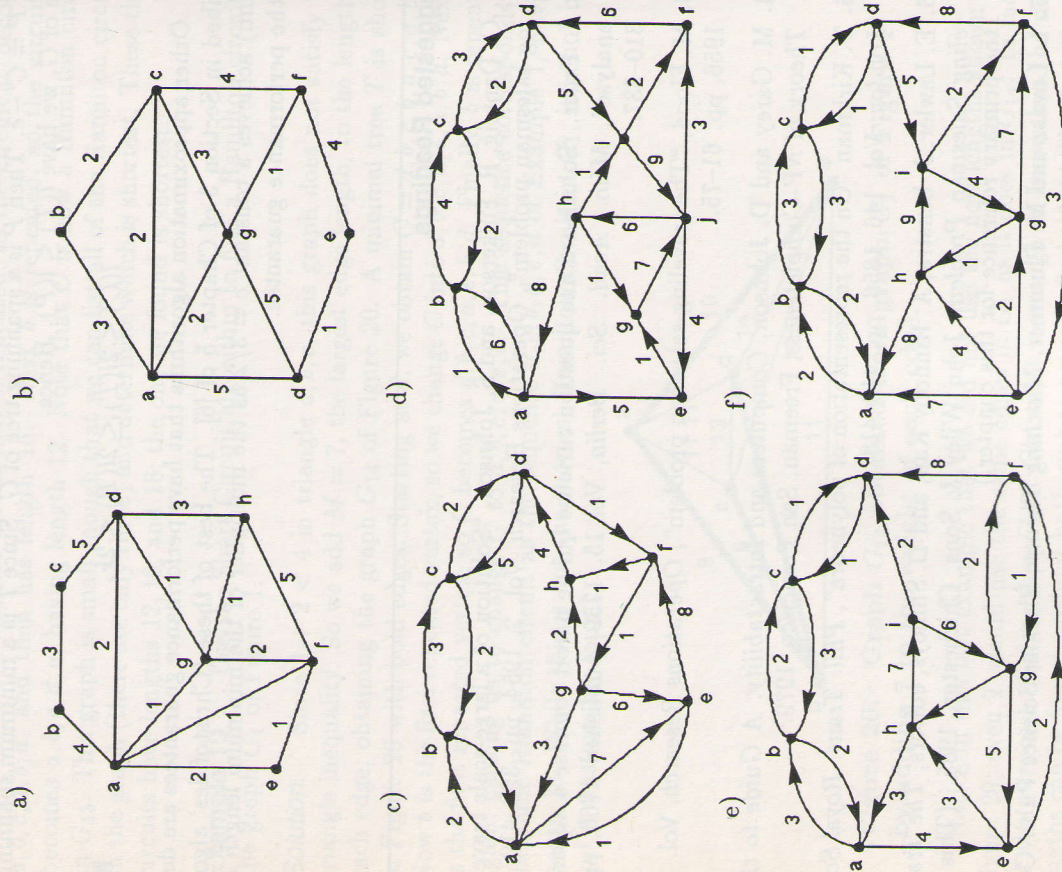
Other approximation algorithms that have performance guarantees are described in Section 3 of Chapter 5 of [6]. The best of these (Christofides' algorithm) achieves a constant  $c = 3/2$  as the multiplier of the minimum length  $k$  in the performance guarantee.

### Suggested Readings

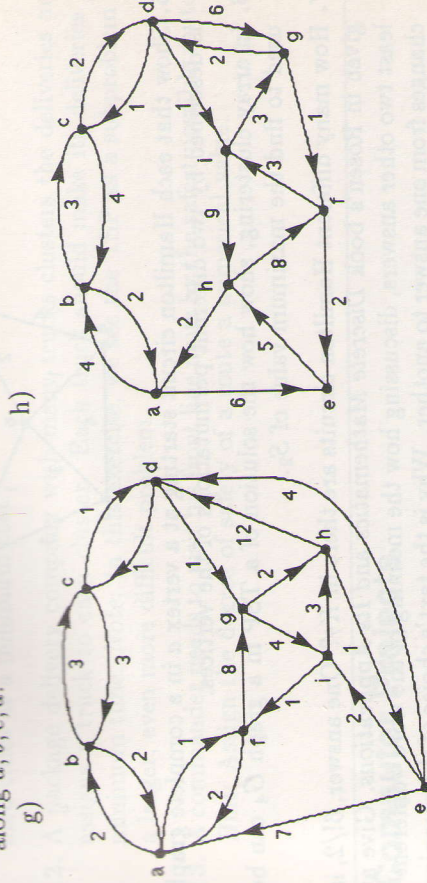
1. G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a large-scale traveling-salesman problem", *Operations Research*, Vol. 2, 1954, pp. 393–410.
2. L. Euler, "Solution d'une question curieuse qui ne paroit soumise à aucune analyse", *Mém. Acad. Sci. Berlin*, Vol 15, 1759, published 1766, pp. 310–337.
3. M. Flood, "The traveling-salesman problem", *Operations Research*, Vol. 4, 1956, pp. 61–75.
4. M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.
5. T. Kirkman "On the representation of polyhedra", *Phil. Trans. Royal Soc. London*, Vol. 146, 1856, pp. 413–418.
6. E. Lawler, J. Lenstra, A. Rinnooy Kan, and D. Shmoys, editors, *The Traveling Salesman Problem*, John Wiley & Sons, Chichester, 1985. (This is the primary reference for this chapter.)
7. L. Lovász and M. Plummer, *Matching Theory*, Elsevier Science Publ. Co., New York, 1986.

**Exercises**

1. Solve the traveling salesman problem by listing all of the possible Hamiltonian circuits and then choosing the shortest one. Notice sequences of vertices such as those forced by vertices of degree two; they can be used to reduce the number of possibilities. Also, use reductions when possible.



Hint for g) and h): Try working from the two different possible directions along a, b, c, d.

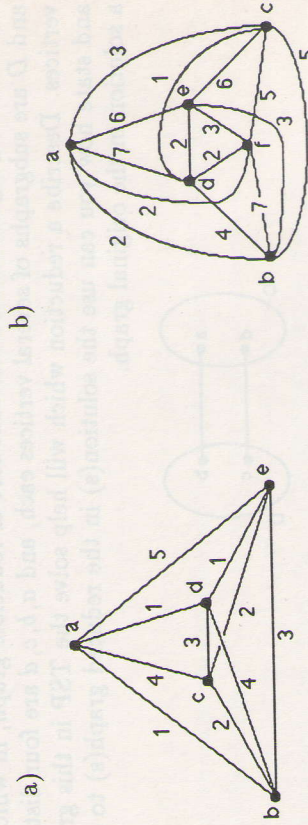


2. Use the method shown in this chapter to permute the rows and columns of the given matrix and thus show the industrial groupings of the companies and their activities.

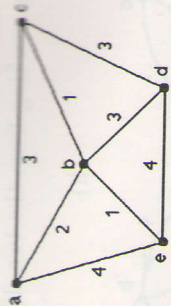
	a	b	c	d	e
A	1	1	0	0	0
B	0	0	1	0	1
C	0	1	0	1	0
D	0	0	1	0	1
E	0	1	0	1	0

a)      b)

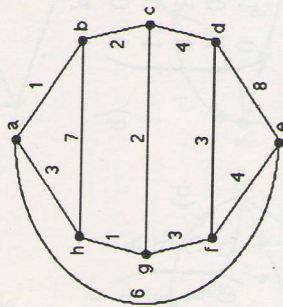
3. Use procedure Short Circuit on the following graphs.



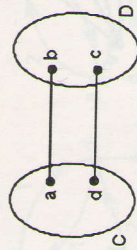
\*4. Find a minimum cost Hamiltonian circuit in the following graph. What strategy did you follow? Discuss the weaknesses of your strategy when applied to large examples.



5. Show that each Hamilton circuit starting at a vertex  $a$  in a complete graph is described by two different permutations of the vertices.
6. In array clustering, show how the solution of a TSP in a graph  $G_4$  can be used to find the maximum value of  $S_2$ .
7. How many different Hamilton circuits are there in  $K_7$ ? One answer,  $6!/2$ , is given in Rosen's book *Discrete Mathematics and Its Applications*. Give at least two other answers, discussing how the meaning of the word "different" changes from one answer to another. Why is the text's choice of the meaning of "different" best, or is it?
8. Apply procedure Short Circuit to the following graph. Does anything go wrong? Why?



- \*9. The following figure shows a different sort of reducible graph, in which  $C$  and  $D$  are subgraphs of several vertices each, and  $a, b, c, d$  are four distinct vertices. Describe a reduction which will help solve the TSP in this graph and state how you can use the solution(s) in the reduced graph(s) to find a solution in the original graph.



- In Exercises 10–13, describe the vertices, edges, and edge weights that will convert the problem into a TSP.
10. An armored car must visit its list of banks as quickly as possible using routes thought to be safe.

11. A rail line from New York to San Francisco must pass through many specified cities at a minimum cost.
12. A package delivery company with many trucks clusters the deliveries and assigns a truck to each cluster. Each truck should make its deliveries in minimum time. *Note:* In this exercise, we see the TSP as a subproblem of a larger, even more difficult problem.
13. A commander needs to visit his front line units by the safest possible routes. *Hint:* Assign the degree of safety of a route a numerical value.

## Computer Projects

1. Write a computer program to find all Type I reductions in a directed graph.
2. Write a computer program to find all Type II reductions in a directed graph.
3. Write a computer program calling on the procedures of Computer Projects 1 and 2 to exactly solve the TSP in a directed graph of at most 50 vertices.