Part II: Applications of Network Models

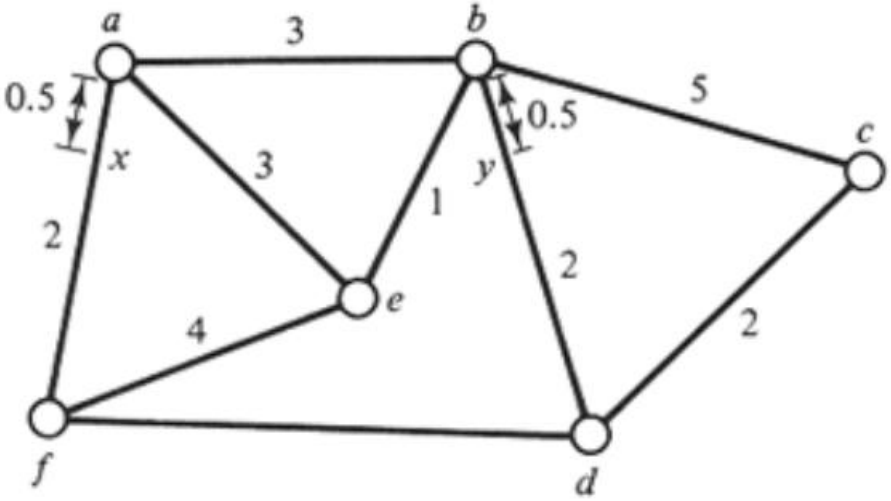
Lecture 2: Minimum Spanning Tree and Routing Problems

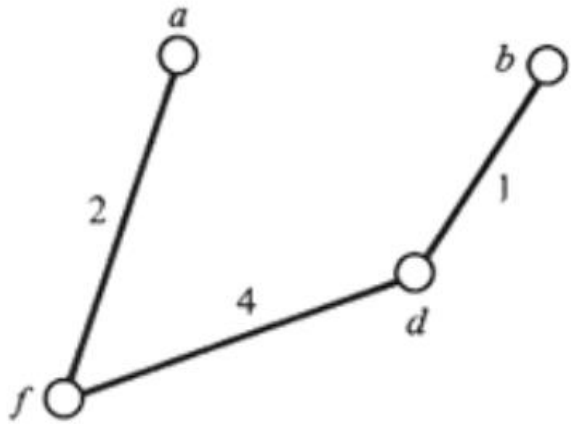
Junlong Zhang

zhangjunlong@tsinghua.edu.cn

Minimum-Spanning-Tree Problem

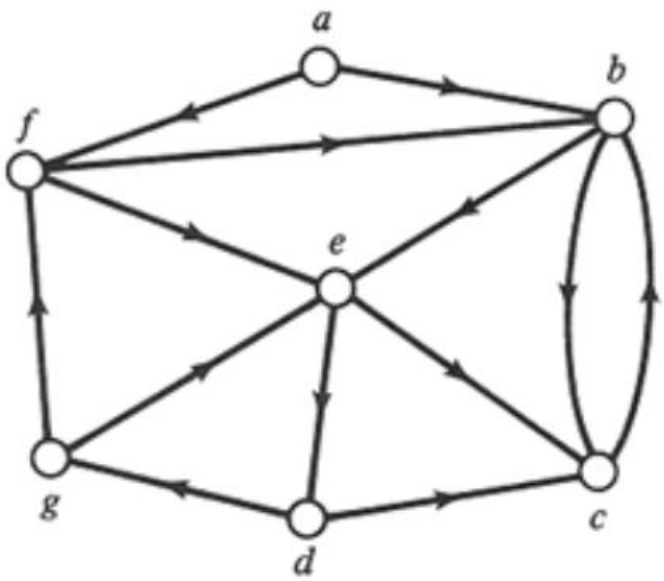
- For an undirected graph $G(N, A)$
  - A tree is a connected subgraph that has no cycles
  - A tree with t nodes has exactly $t - 1$ edges and there exists a single path between any two nodes on the tree
  - A spanning tree contains the complete set N of the nodes of G

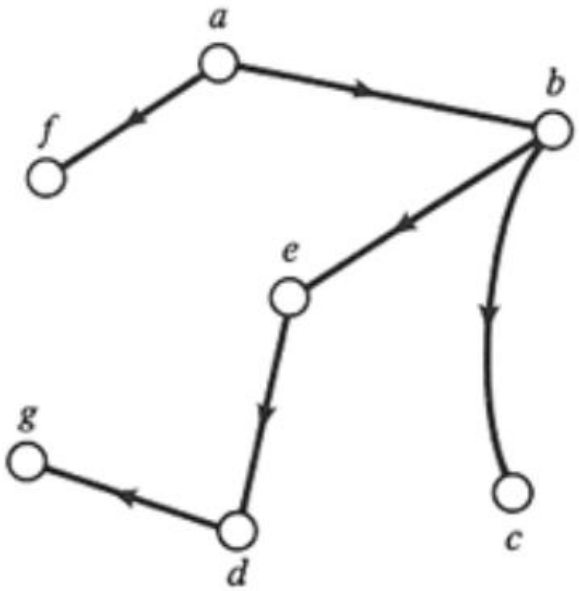




Minimum-Spanning-Tree Problem

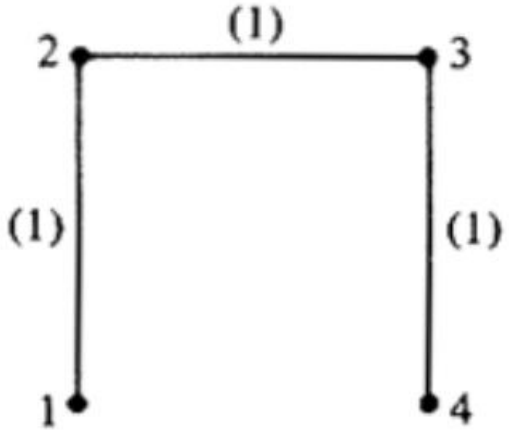
- For a directed graph $G(N, A)$
  - A directed tree has a root node and a unique path from that node to every other node in the tree
  - An directed spanning tree (arborescence) is a directed tree that contains the complete set N of the nodes of G

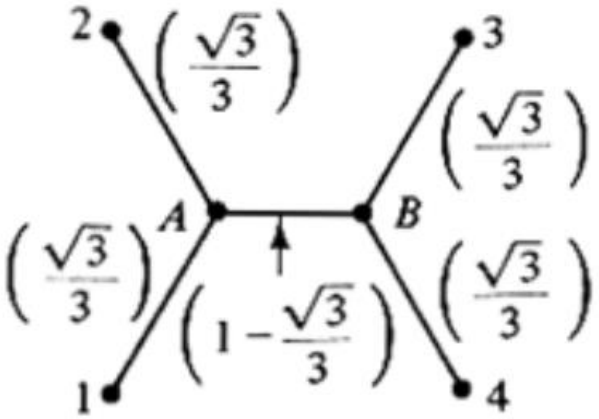




Minimum-Spanning-Tree Problem

- Minimum-spanning-tree (MST) problem: finding the spanning tree with minimum total link length.
- One example: given locations of n rural towns and the associated distances, build the minimum-length roads to connect, directly or indirectly, all pairs of towns
- Steiner's problem: has the same objective but allows the introduction of artificial nodes.





Minimum-Spanning-Tree Problem

Fundamental property of MST's. Suppose that a procedure for finding the MST of any graph G has been discovered and that in the course of the construction of the MST, by using this procedure, the set N of nodes of G has been partitioned into k distinct trees $T_1(N_1, A_1)$, $T_2(N_2\ A_2)$, ..., $T_k(N_k, A_k)$ with $N_1 \cup N_2 \cup \ldots \cup N_k = N$ and $N_i \cap N_j = \varnothing$ for i, j = 1, 2, ..., k ($i \neq j$). By the construction assumption T_1, T_2, ... , T_k will eventually become constituent parts of the MST. Note that a "tree" may also consist of a single node of G. Now let T_s be one of these trees and let (i^*, j^*) be the shortest link with the property that one of its roots, node i^*, is in the tree T. and the other root, node j^*, is not in the tree T_s. Let T_t be the tree to which j^* belongs. It is then true that the link (i^*, j^*) must be a link in the final MST and that, therefore, trees T_s and T_t can be connected next through link (i^*, j^*)

Minimum-Spanning-Tree Problem

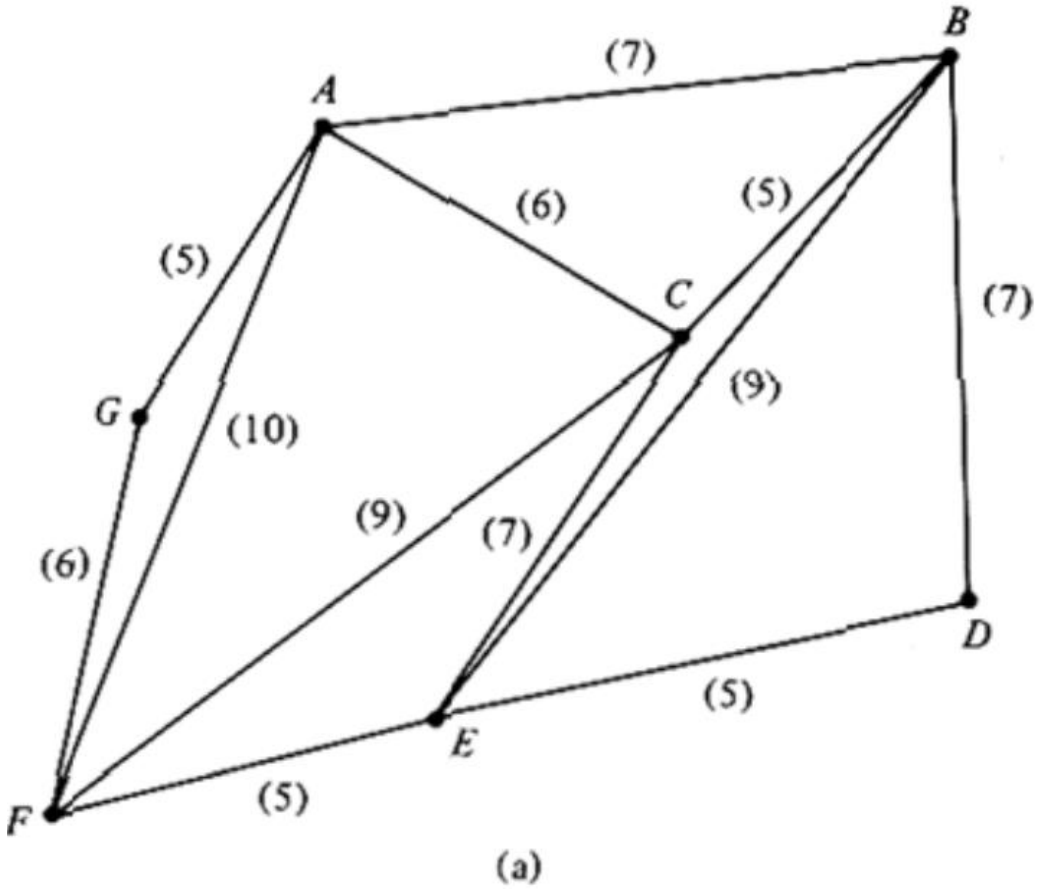
Proof: (By contradiction) Assume that (i^*, j^*) is not part of the final MST, which, however, must, by assumption, contain both the tree T_s and the tree T_t. By definition of the MST, there is exactly one path leading from T_s to T_t. Let then (i, j) be that link on the MST which has one of its roots, i, in T_s and the other, j, not in T_s and which is the first link on the path from T_s to T_t (It is possible that $i = i^*$ or that $j = j^*$.) By definition of (i^*, j^*), $l(i, j) > l(i^*, j^*)$. If we then replace (i, j) by (i^*, j^*), a new spanning tree will result which will have less total length than the MST--a contradiction.

Corollary: In an undirected network, G. the link of shortest length out of any node is part of the MST.

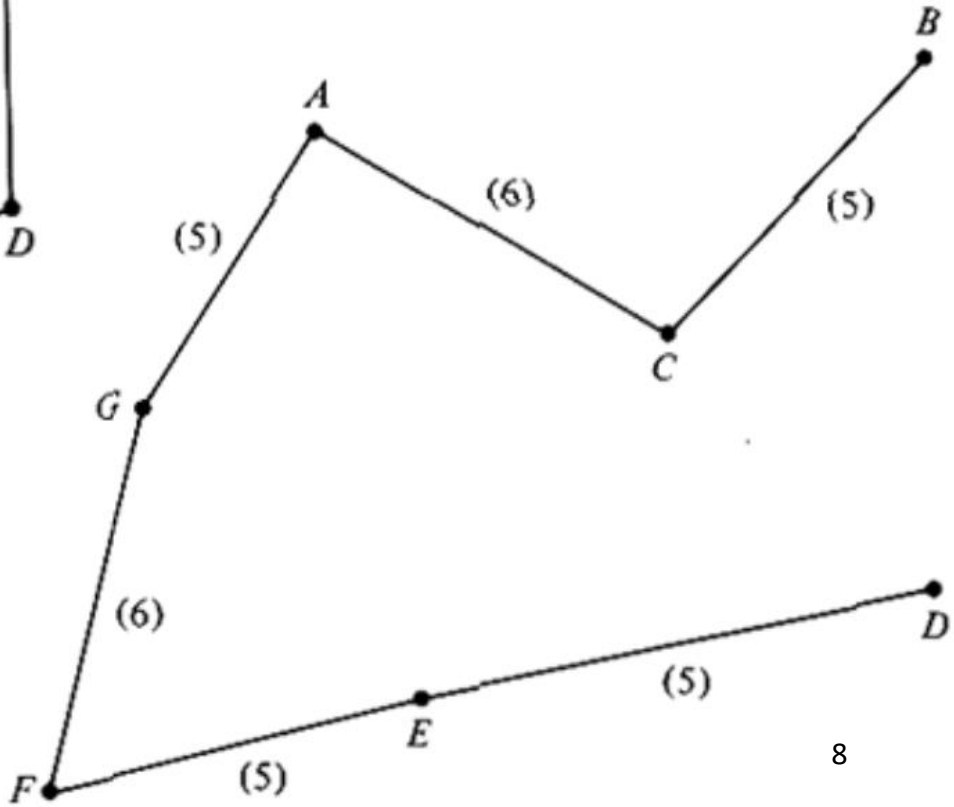
MST Algorithm

- Step 1. Begin at some arbitrary node i. Find the node closest to i, say j, and connect it to i. Break ties, if any, arbitrarily.
- Step 2. If all nodes have been connected, stop.
- Step 3. Find the one among the still isolated nodes which is closest to the already connected nodes and connect it with the already connected nodes. Break ties, if any, arbitrarily. Go to Step 2.

MST Example



(a)



Routing Problems

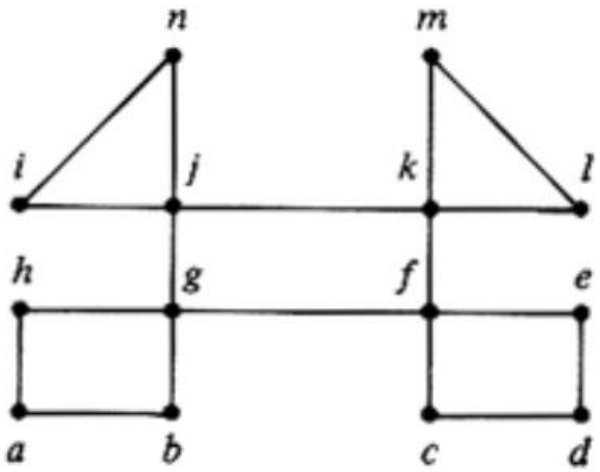
- Design of routes for vehicles or people
  - cleaning and sweeping of streets
  - plowing of snow after a snowstorm
  - delivery of mail to residences
  - collection of refuse from houses

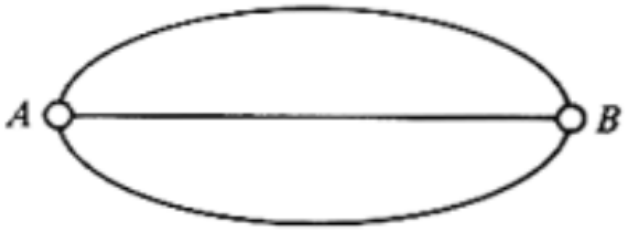
  - daily routing of school buses
  - distribution of newspapers to newsstands
  - delivery of mail packages to addressees

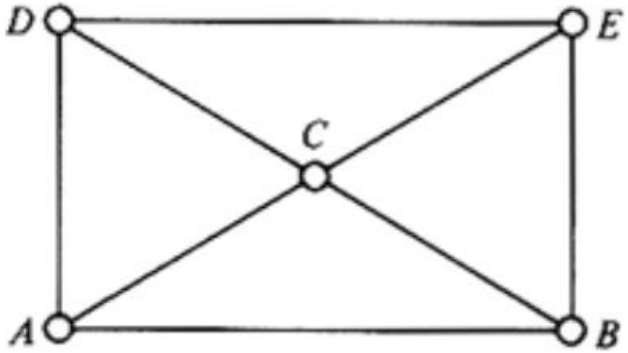
Edge Covering: Chinese Postman's Problem (CPP)

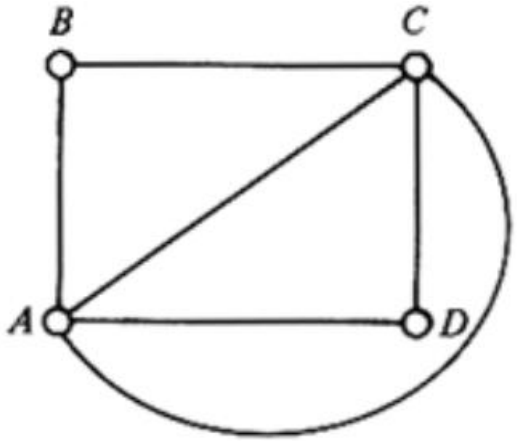
- CPP on an undirected graph: find a minimum-cost circuit that will traverse every edge of G *at least* once.
- An Euler tour is a circuit which traverses every edge on a graph *exactly once* (beginning and terminating at the same node).
- An Euler path is a path which traverses every edge on a graph *exactly once*.
- Euler's Theorem: A connected graph G possesses an Euler tour (Euler path) if and only if G contains exactly zero (exactly two) nodes of odd degree.

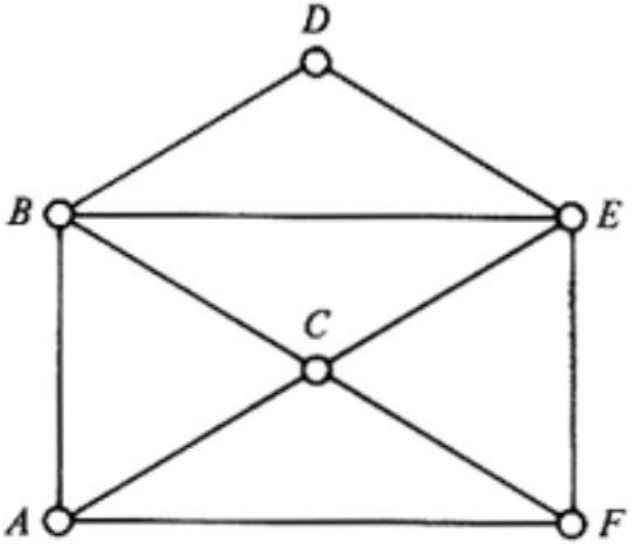
The Chinese Postman's Problem

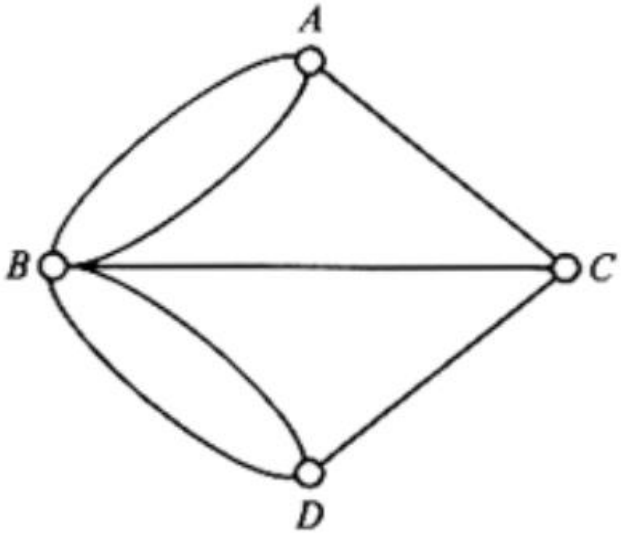












Obtaining an Euler Tour

- Euler-Tour Algorithm
  - Begin at n_0 and traverse successively the edges of G. Deleting from G each edge as it is traversed.
  - Do not traverse an edge that is an *isthmus*, whose deletion would divide the yet undeleted part of G into two separate nonempty components.
  - Continue until all edges of G have been deleted.

Solving the Chinese Postman's Problem

- Lemma 1: The number of nodes of odd degree in an undirected graph G is always even.

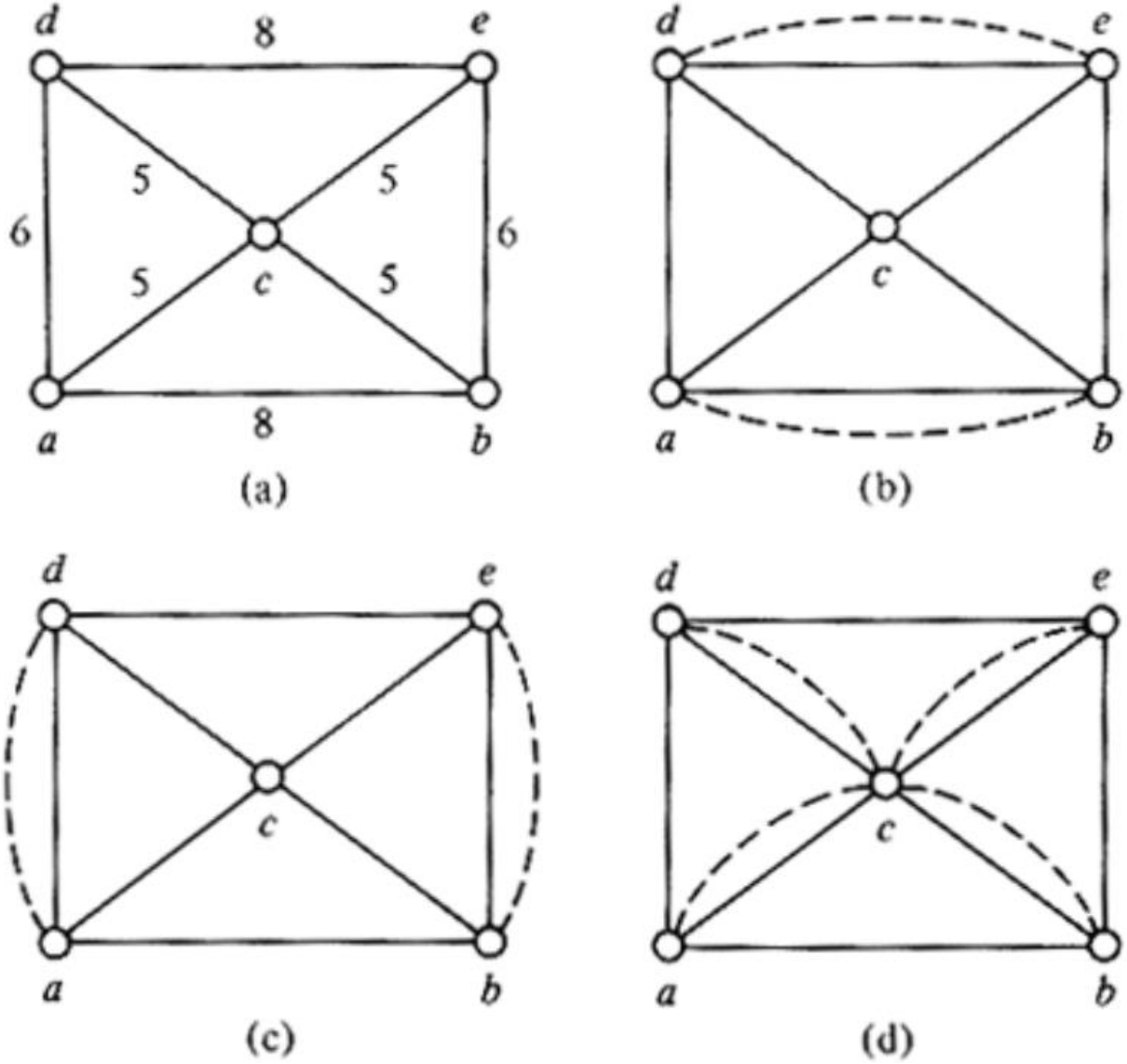
Hint: The sum of the degrees of all the nodes in G is an even number since each edge is incident on two nodes.

- A pairwise matching of a subset $N' \subset N$ of nodes of a graph G is a pairing of all the nodes in N' (assuming that the number of nodes in N' is even).
- A minimum-length pairwise matching of the nodes in N' is then a matching such that the total length of the shortest paths between the paired nodes is minimum.

Chinese Postman Algorithm--Undirected Graph

- Step 1. Identify all nodes of odd degree.
- Step 2. Find a minimum-length pairwise matching of the m odd-degree nodes (m is even by Lemma 1) and identify the $m/2$ shortest paths between the two nodes of each pair.
- Step 3. For each pair found in Step 2, add to the graph G the edges of the shortest path between the two nodes in the pair. The resulting graph is G^1 with no nodes of odd degree.
- Step 4. Find an Euler tour on G^1. This Euler tour is an optimal solution to the CPP on G. The length of the optimal tour is equal to the total length of the edges in G plus the total length of the edges in the minimum-length matching.

Chinese Postman Algorithm--Undirected Graph



Chinese Postman Algorithm--Undirected Graph

- In step 2, with m odd degree nodes in G, there are

$$\prod_{i=1}^{m/2}(2i-1)$$

possible distinct pairwise matching combinations

- For an exact algorithm for minimum length matchings on undirected graphs with complexity $O(n^3)$, see

Edmonds, J. , & Johnson, E. L. (1973). Matching, Euler tours and the Chinese postman. *Mathematical Programming*, 5(1), 88-124.

Node Covering: The Traveling Salesman Problem

- Traveling salesman problem (TSP): Find the minimum distance route that begins at a given node, visits all the members of a specified set of nodes exactly (or at least) once, and returns eventually to the initial node.

- TSP1
  - completely connected network

    (there exists a link between any two nodes)
  - triangular inequality is satisfied
  - symmetric distance matrix
  - each node is visited exactly once

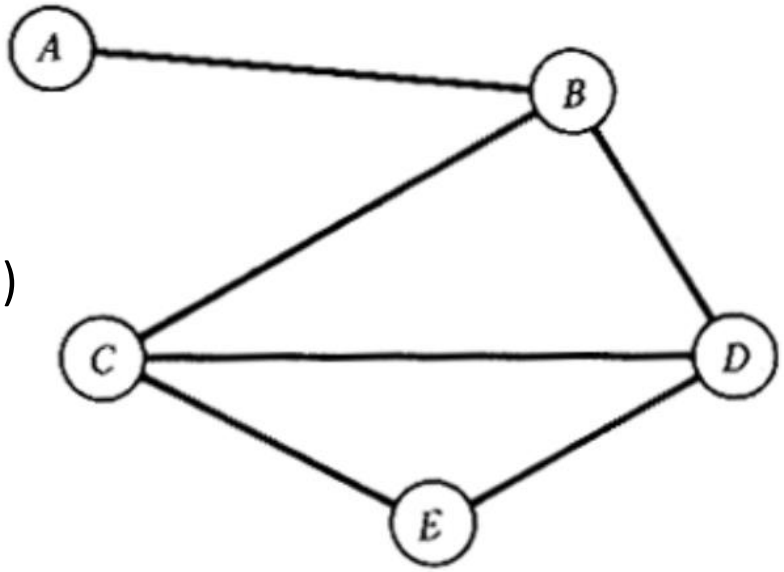


FIGURE 6.19 Network on which a traveling salesman tour must visit at least one node (node *B*) twice.

Heuristic Algorithm for TSP1

- Step 1. Find the minimum spanning tree T of G.
- Step 2. Let n_0 of the n nodes of T be odd-degree nodes. Find a minimum-length pairwise matching of these n_0 nodes on G. Let M be the graph consisting of the links in the optimal matching. Let $H = M \cup T$.
- Step 3. Find an Euler tour on H starting and ending at the depot. This Euler tour is a (approximate) solution to the TSP.
- Step 4. Check for nodes of H visited more than once in the Euler tour and improve the TSP solution by taking advantage of the triangular inequality.

TSP1 Example

FIGURE 6.21 Depot and nine points to be visited.

TABLE 6-1 Distance matrix for refuse-collection example.

From \ To	1	2	3	4	5	6	7	8	9	10
1	0	25	43	57	43	61	29	41	48	71
2	25	0	29	34	43	68	49	66	72	91
3	43	29	0	52	72	96	72	81	89	114
4	57	34	52	0	45	71	71	95	99	108
5	43	43	72	45	0	27	36	65	65	65
6	61	68	96	71	27	0	40	66	62	46
7	29	49	72	71	36	40	0	31	31	43
8	41	66	81	95	65	66	31	0	11	46
9	48	72	89	99	65	62	31	11	0	36
10	71	91	114	108	65	46	43	46	36	0

TSP1 Example

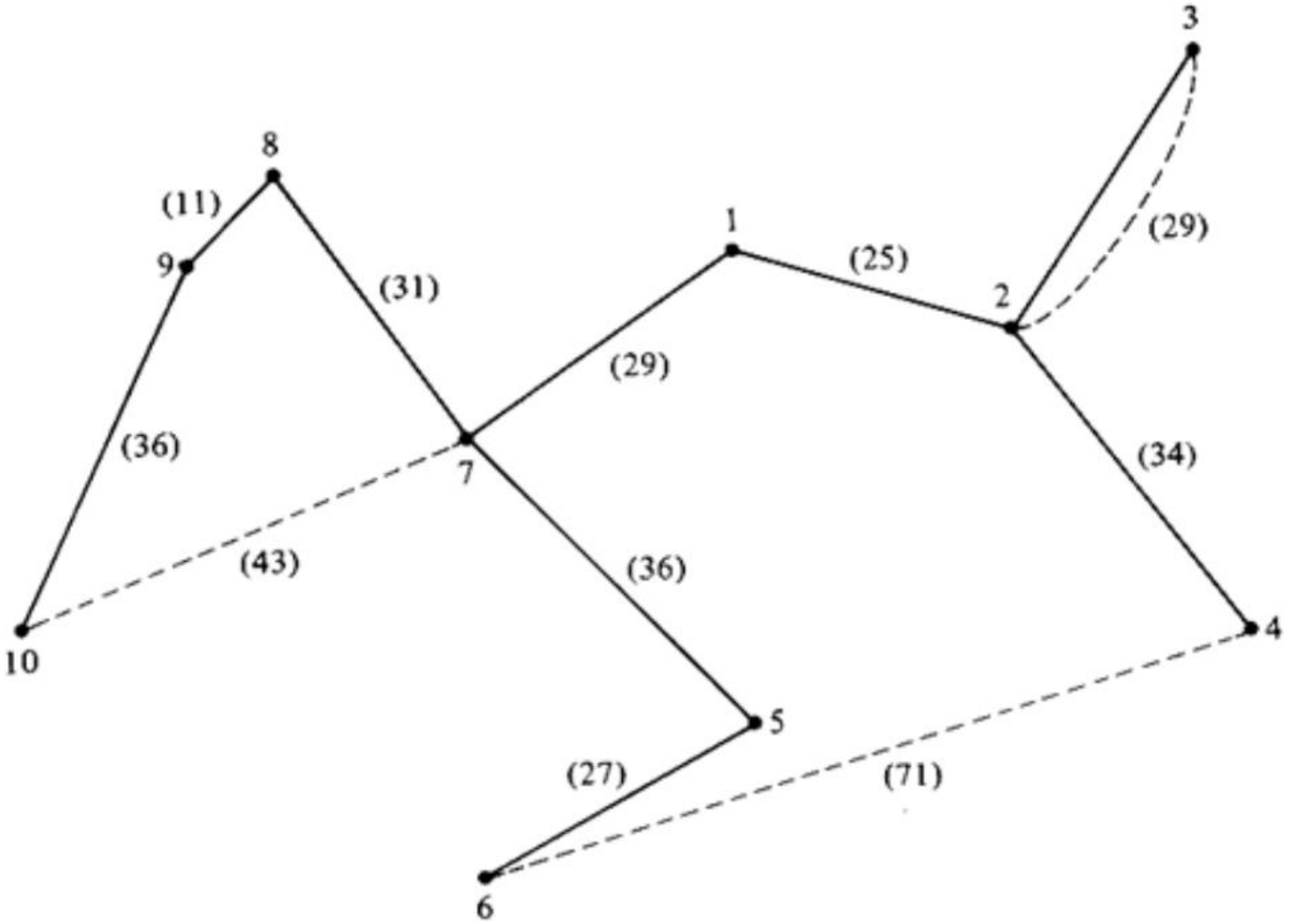


FIGURE 6.23 Ten-Node problem after the matching of odd-degree nodes.

TSP1 Example

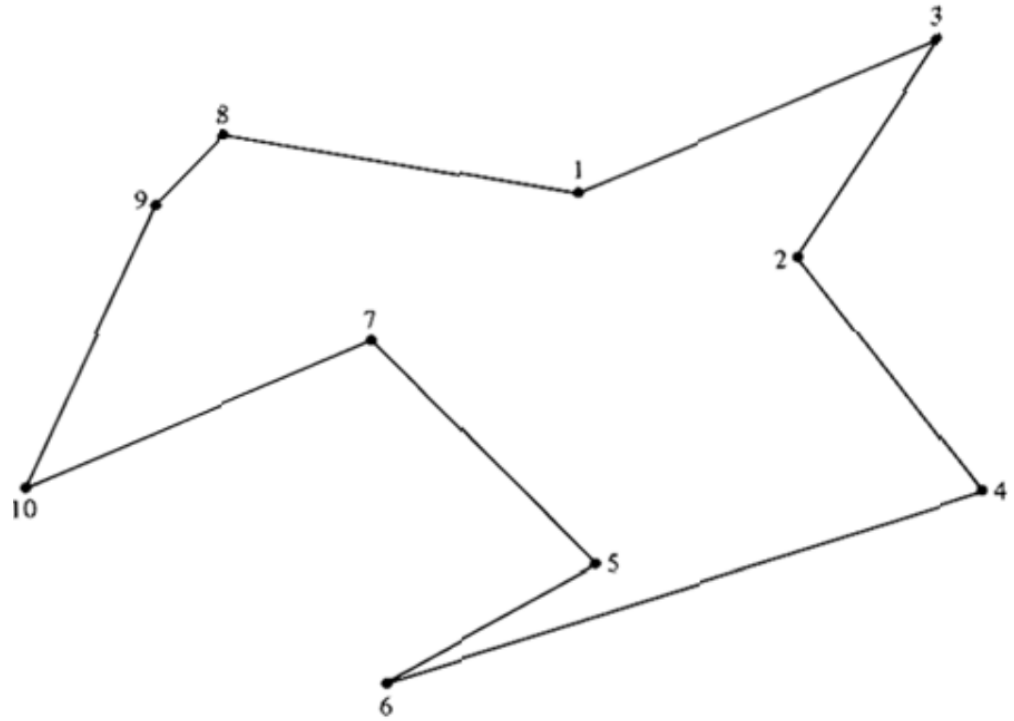


FIGURE 6.24 Solution that results from the heuristic algorithm.

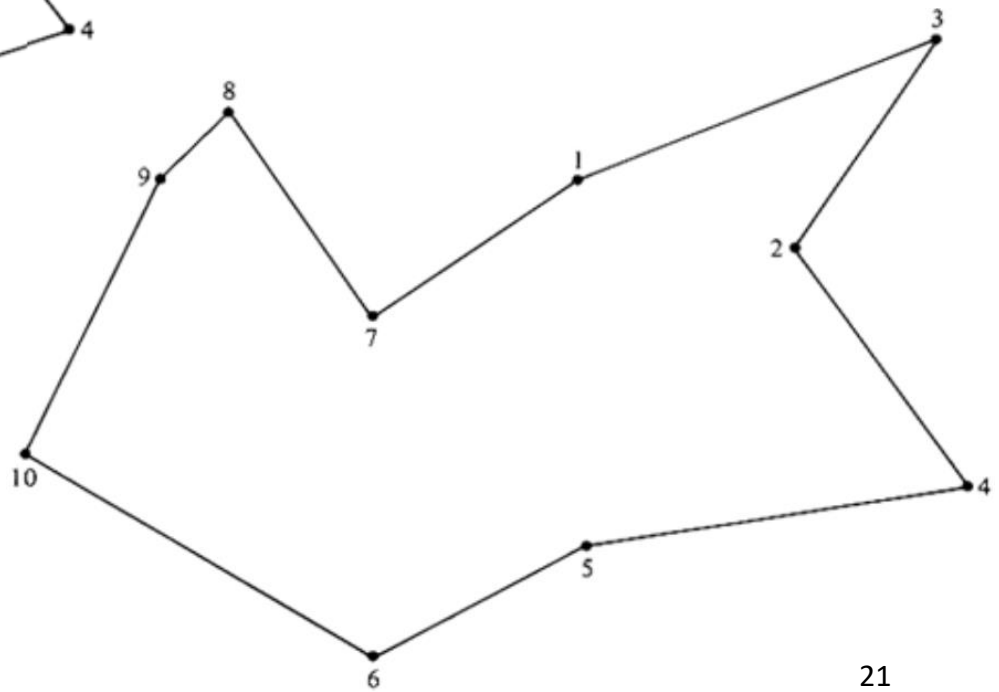


FIGURE 6.25 True optimum solution.

Heuristic Algorithm for TSP1

- Denote by $L(T)$, $L(M)$, $L(H)$, and $L(TST)$ the length of T, M, H, and the optimum traveling salesman tour.
- Theorem: $L(H) < \frac{3}{2} * L(TST)$.
- Proof
  - Since TST covers all nodes and visits each exactly once, a spanning tree will be obtained if any one of the links on the TST is removed. It follows that $L(T) < L(TST)$.
  - For the n_0 points optimally matched in Step 2, rematch them optimally by using only links on TST. Let M' be the set of links in this matching. Then $L(M') < L(TST)/2$, otherwise M' cannot be minimum length matching on TST.
  - $L(H) = L(M) + L(T) \leq L(M') + L(T) < 3/2 * L(TST)$

Euclidean TSP

- Euclidean TSP: TSP1 with Euclidean distance.
- Property 1: The optimum traveling salesman tour does not intersect itself.

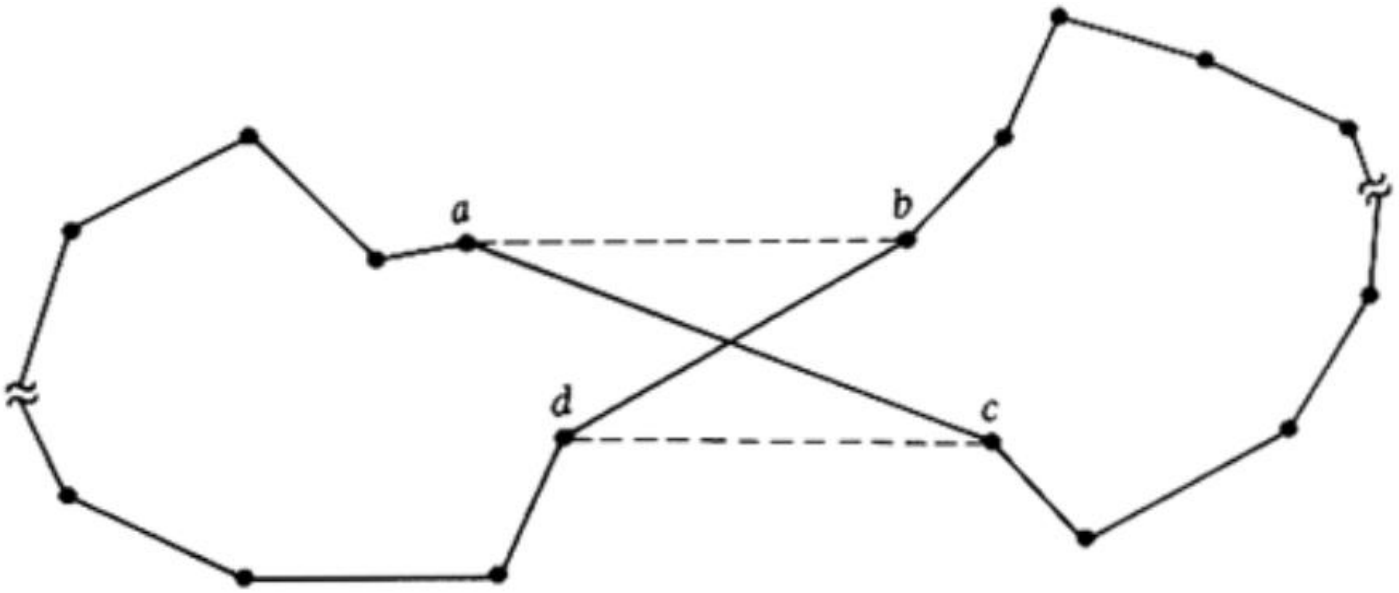


FIGURE 6.26 The traveling salesman tour can be improved by substituting the intersecting links (a, c) and (b, d) by (a, b) and (c, d).

Euclidean TSP

- Property 2: Let m of the n points in the Euclidean TSP define the convex hull of the points. Then the order in which these m points appear in the optimal TSP tour is the same as the order in which these same points appear on the convex hull.

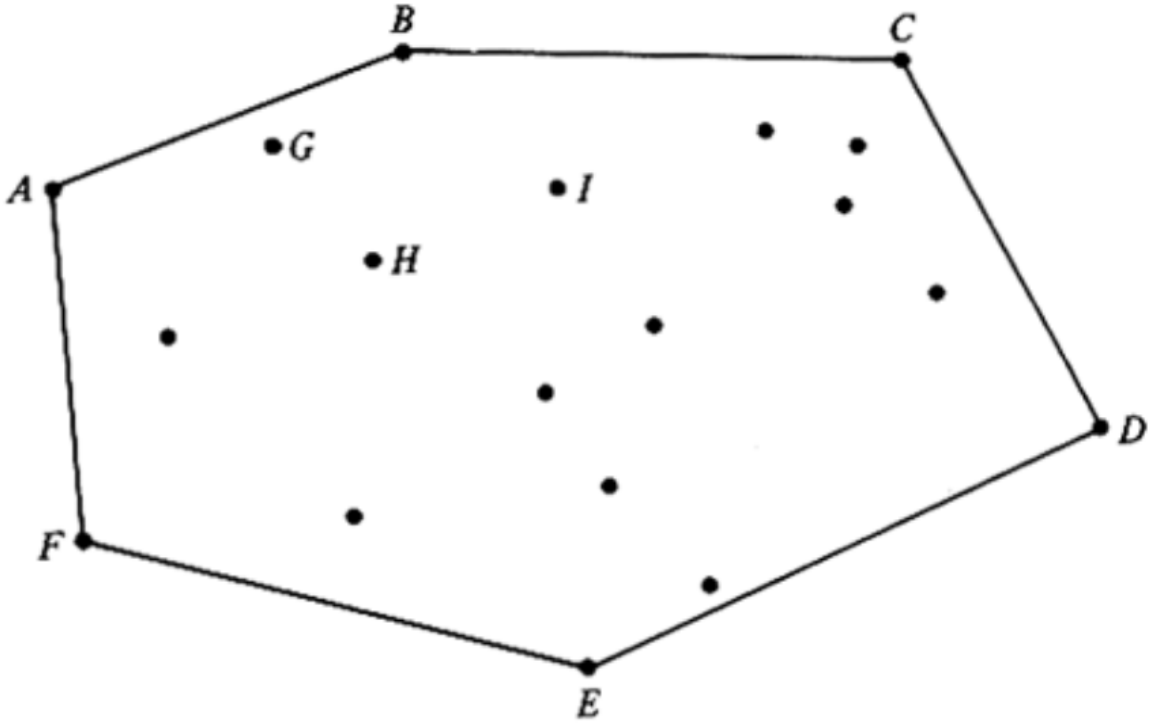


FIGURE 6.27 The convex hull of the points (*A, B, C, D, E, F, G, H, I, . . .*) is the polygon *ABCDEF*.

Probabilistic View of the Euclidean TSP

- Observation: a given TSP tour varies in proportion to the square root of the area in which it is contained.
- Assume n points are randomly and independently dispersed over an area A with the location determined by a uniform distribution. Let $L(n, A)$ be the length of the optimal TSP tour.
- Theorem

$$\lim_{n\to\infty} E[L(n, A)]/\sqrt{n} = K\sqrt{A}$$

$$L(n, A)/\sqrt{n} \xrightarrow[n\to\infty]{} K\sqrt{A} \qquad \text{(with probability 1)}$$

- K is a constant and approximately equal to 0.765.
- Application of the approximation formula $0.765\sqrt{nA}$
  - Assessing a heuristic solution to a TSP
  - Preliminary planning of urban collection and delivery systems

Homework

- Reading: Sections 6.1-6.4
- Homework (6 problems)
  - **P1**: Prove Euler's theorem by using your own language.
  - **P2**: Prove Property 1 and Property 2 on Slide 24.
  - **Problem 6.1**: https://web.mit.edu/urban_or_book/www/book/chapter6/problems6/6.1.html
  - **Problem 6.3**: https://web.mit.edu/urban_or_book/www/book/chapter6/problems6/6.3.html
  - **Problem 6.4**: https://web.mit.edu/urban_or_book/www/book/chapter6/problems6/6.4.html
  - **Problem 6.6**: https://web.mit.edu/urban_or_book/www/book/chapter6/problems6/6.6.html

  - ***Due on May 12 (Wed), before 15:00 pm***
  - ***Late submission will NOT be accepted!***
  - ***Copy and paste will be graded ZERO!***