The Shortest-Network Problem

What is the shortest network of line segments interconnecting an arbitrary set of, say, 100 points? The solution to this problem has eluded the fastest computers and the sharpest mathematical minds

by Marshall W. Bern and Ronald L. Graham

The imaginary Steiner Telephone Company figured that it would save millions of dollars if it could find the shortest possible network of telephone lines to interconnect its 100 customers. In search of a solution, Steiner hired the Cavalieri Computer Company, known for the world's fastest programmers and computers. After a week Cavalieri presented a program to solve Steiner's problem and showed that the program had indeed found a shortest network for 15 of the customers in just one hour. Steiner paid Cavalieri $1,000 for the program and promised to pay one cent per second for the time it would take a computer to generate the complete solution. By the time the computer had finished the calculation for all 100 customers, the telephone company owed trillions of dollars in computer expenses and its customers had all moved many kilometers away—either by choice or by continental drift!

Did Cavalieri sell Steiner a faulty program? This dilemma is one example of the Steiner problem, which asks for the shortest network of line segments that will interconnect a set of given points. The Steiner problem cannot be solved by simply drawing lines between the given points, but it can be solved by adding new ones, called Steiner points, that serve as junctions in a shortest network. To determine the location and number of Steiner points, mathematicians and computer scientists have developed algorithms, or precise procedures. Yet even the best of these algorithms running on the fastest computers cannot provide a solution for a large set of given points because the time it would take to solve such a problem is impractically long. Furthermore, the Steiner problem belongs to a class of problems for which many computer scientists now believe an efficient algorithm may never be found. For this reason the Cavalieri Computer Company should be exonerated.

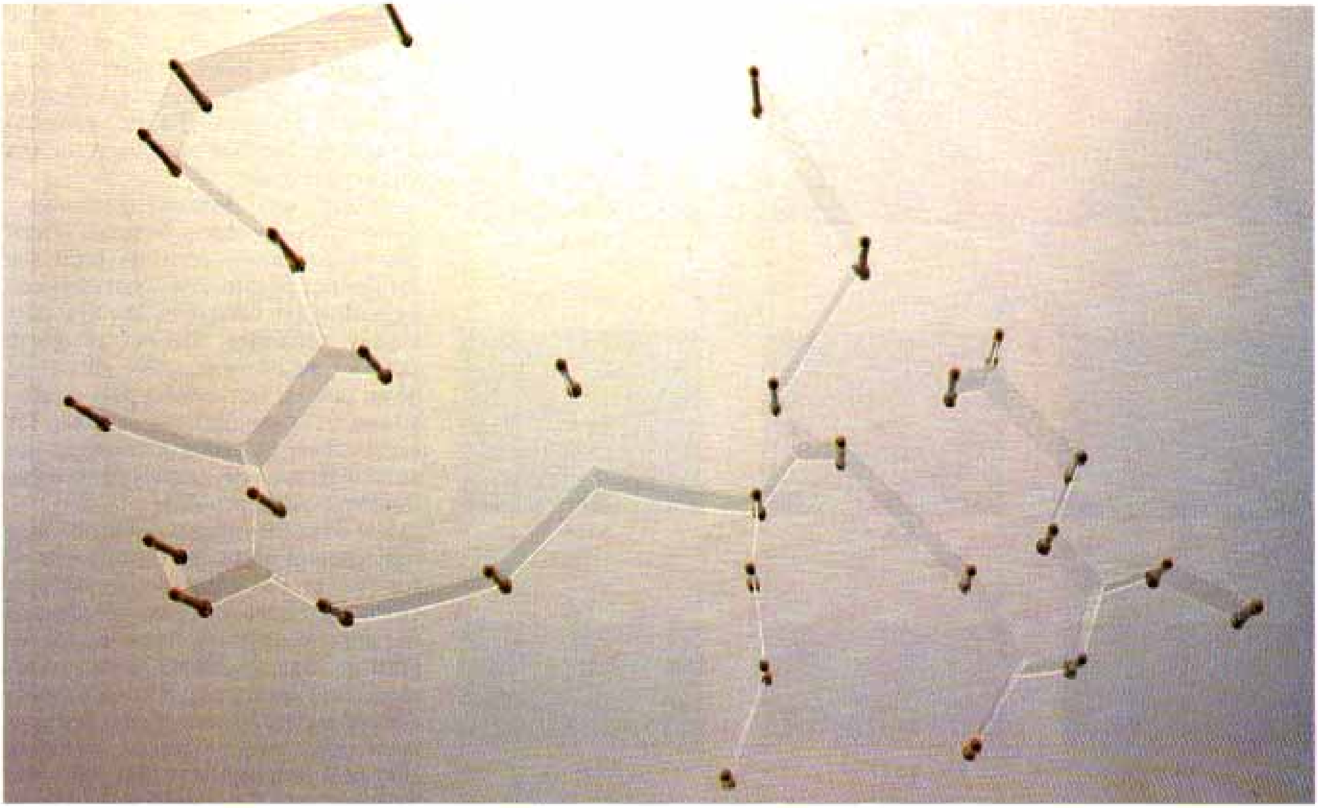
On the other hand, Cavalieri could have developed a practical program that would have yielded solutions somewhat longer than the shortest network. Approximate solutions to the shortest-network problem are computed routinely for numerous applications, among them designing integrated circuits, determining the evolutionary tree of a group of organisms and minimizing materials used for networks of telephone lines, pipelines and roadways.

The Steiner problem, in its general form, first appeared in a paper by Miloš Kössler and Vojtěch Jarník in 1934, but the problem did not become popular until 1941, when Richard Courant and Herbert E. Robbins included it in their book *What Is Mathematics?* Courant and Robbins linked the problem to the work of Jakob Steiner, a 19th-century mathematician at the University of Berlin. Steiner's work sought the single point whose connections to a set of given points had the shortest possible total length. In about 1640, however, a special case of both problems—the one Steiner worked on and the one that bears his name—was first posed: Find the point P that minimizes the sum of the distances from P to each of three given points. Evangelista Torricelli and Francesco Cavalieri solved the problem independently. Torricelli and Cavalieri deduced that if the angles at point P are all 120 degrees or more, then the total distance is minimized.

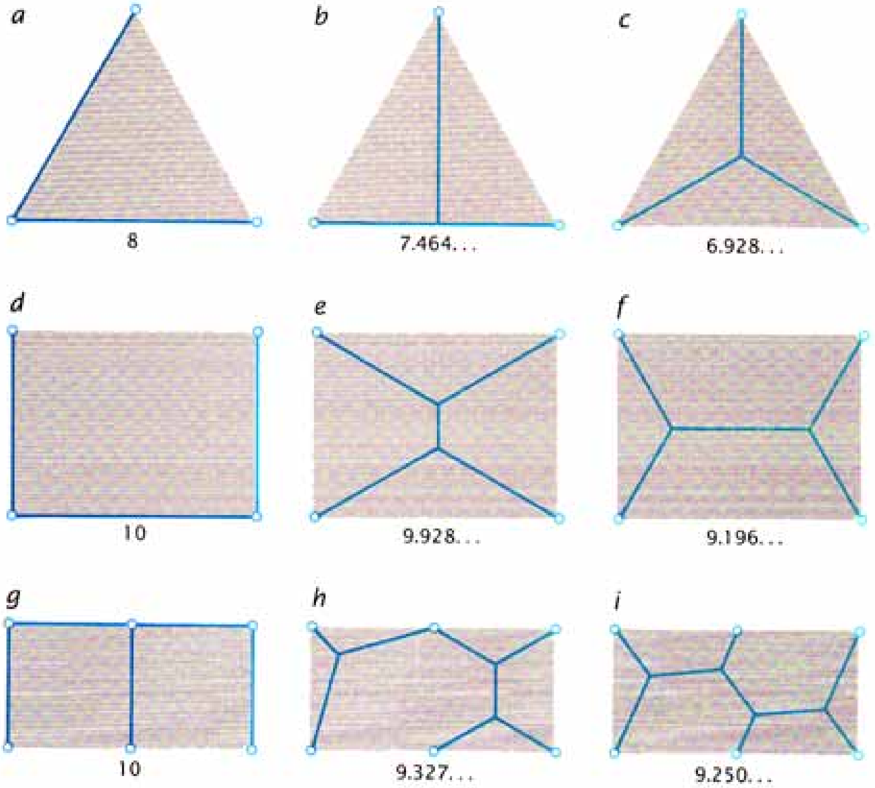
Knowing that the angles at P measure at least 120 degrees, Torricelli and Cavalieri developed a geometric construction for finding P [*see top illustration on page 87*]. Line segments are drawn connecting the given points (call them A, B and C, with B at the vertex of the largest angle). If angle B is greater than or equal to 120 degrees, then point P coincides with point B. In other words, the shortest network is simply the line segments between A and B and between B and C. If angle B is less than 120 degrees, then point P must be somewhere inside the triangle. To find it, an equilateral triangle is drawn along the longest side of the triangle, namely the side between points A and C. The third vertex of the equilateral triangle, labeled X, is opposite point B. The equilateral triangle is circumscribed, and a line segment is drawn from point B to X. Point P lies where the line intersects the circle. Joining points A, B and C to P creates three angles of exactly 120 degrees and yields the shortest network. Furthermore, the length of the line from B to X turns out to be equal to the length of the shortest network. For the purpose of our article we shall call X the replacement point, since replacing points A and C with X leaves the length of the network unchanged.

The three-point and the multipoint Steiner problem share many properties. The form of the solutions, known as trees, is such that removing any line segment from the shortest network detaches one of the given points. In other words, one cannot follow the network from a given point back to the same point without retracing lines.

MARSHALL W. BERN and RONALD L. GRAHAM have studied shortest-network problems for many years. Bern is a researcher at the Xerox Palo Alto Research Center. After earning his M.A. at the University of Texas at Austin in 1980, he worked in the signal-processing division of Ensco, Inc. He received his Ph.D. in computer science from the University of California, Berkeley, in 1987. In his spare time Bern grows cacti and dabbles in painting and printmaking. Graham is Adjunct Director for Research, Information Sciences Division, at the AT&T Bell Laboratories and university professor at Rutgers University. He got his Ph.D. in mathematics from Berkeley in 1962. He enjoys pointing out that he is a former president of the International Jugglers Association. This is Graham's second article for SCIENTIFIC AMERICAN.



SOAP-BUBBLE COMPUTER (*top*) challenges an electronic computer (*bottom*) to find the shortest network of line segments interconnecting 29 cities. The soap-bubble computer, in which the placement of pegs mimics the geography, minimizes the length of soap films in a local area. It provides a short network, but not necessarily the shortest one. The electronic computer implements an algorithm, authored by Ernest J. Cockayne and Denton E. Hewgill of the University of Victoria, that guarantees the true shortest network. The 29-point problem is close to the current limit of computing capabilities.



NETWORK PROBLEM for points arranged at the vertexes of an equilateral triangle, a rectangle and a "ladder" has various solutions. In *a, d* and *g* the points are connected without adding new points, in what is known as the minimum spanning solution, or tree. Steiner trees, which are made by adding additional junction points, are shown in *c, e, f, h* and *i*. Only *c, f* and *i* are shortest Steiner trees, or shortest networks. The number under each solution gives the approximate total length of the line segments.

Solutions to the three-point and multipoint problem are therefore called Steiner trees, the segments are known as edges, and the points—analogous to *P*—that must be added to construct the tree are called Steiner points.

The three-point Steiner problem also provides information about the geometry of shortest Steiner trees. First, every angle measures at least 120 degrees, which implies that every given point is connected to the tree by no more than three edges. Second, at every Steiner point exactly three edges meet, at 120-degree angles. Third, the number of edges in a tree is always one less than the number of given points added to the Steiner points. And last, since exactly three edges meet at every Steiner point and at least one edge must touch every given point, the maximum number of Steiner points in any problem is two fewer than the number of given points.

For the same number and arrangement of given points, many different Steiner trees can be constructed that have those properties. Some of the trees, known as locally minimal solutions, cannot be shortened by a small perturbation, such as moving an edge slightly or splitting a Steiner point. But not every locally minimal Steiner tree is a shortest solution possible. Large-scale rearrangements of the Steiner points may be necessary to transform a network into a shortest possible tree, called a globally minimal Steiner tree.

Let us demonstrate with a set of given points that define the four corners of a rectangle measuring three meters by four. The solutions have two Steiner points, which can be arranged in two different ways. Each arrangement forms a Steiner tree that has three edges connected to each Steiner point at 120 degrees. If the Steiner points are arranged parallel to the width, the locally minimal Steiner tree that results is about 9.9 meters long. If the Steiner points are arranged parallel to the length, a globally minimal Steiner tree results, measuring about 9.2 meters.

A brute-force approach to discovering a shortest network is to search through all possible locally minimal Steiner trees, calculate their lengths and choose the shortest one. Because Steiner points can be placed anywhere, however, it is not clear that all possible locally minimal Steiner trees can be computed in a finite amount of time. Z. A. Melzak of the University of British Columbia overcame the difficulty and developed the first algorithm for the Steiner problem.

Melzak's algorithm considers many possible connections between given points and many possible locations for Steiner points. The algorithm can be outlined in two parts. The first part simply separates the set of given points into every possible subset of given points. The second part creates a number of possible Steiner trees for each subset by using a construction similar to the one employed to solve the three-point problem.

Just as in the three-point problem, a replacement point can be substituted for two of the given points without changing the length of the solution. In the general problem, however, the algorithm must guess which pair to replace, and eventually it tries all possible guesses. Moreover, the replacement point may be placed on either side of a line segment that joins the pair, because the equilateral triangle used in the construction can be oriented in two directions. After one pair of points in the subset is replaced by one of the two possible replacement points, each subsequent step of the algorithm replaces either two given points, a given point and a replacement point or two replacement points with another replacement point until the subset is reduced to three points.

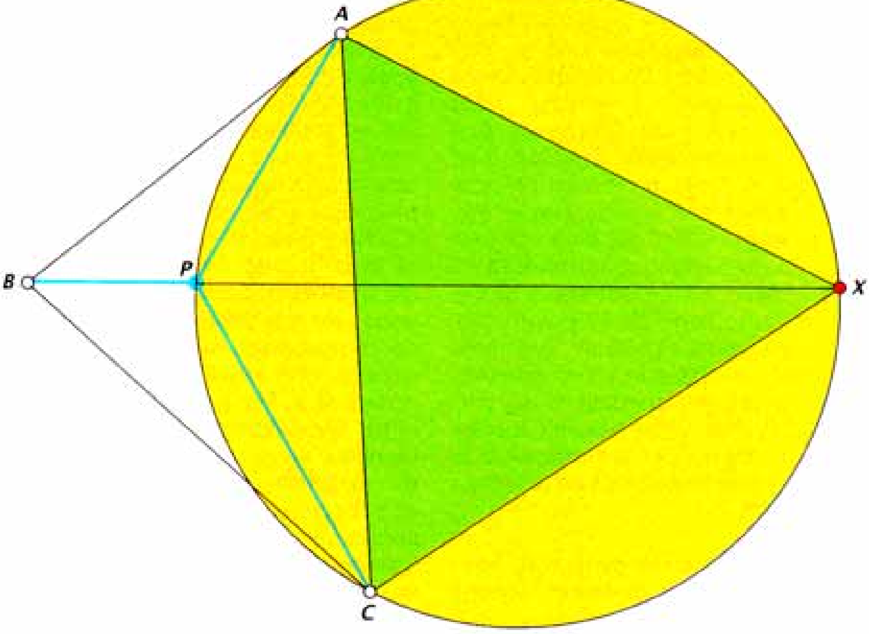
Once the Steiner point for those three points is found, the algorithm works backward, attempting to determine the Steiner point corresponding to each replacement point [*see bottom illustration on opposite page*]. An attempt can fail because of contradictory constraints on the placement of Steiner points. A successful attempt, however, creates a Steiner tree connecting each given point in the subset with one edge. After considering all replacement sequences, the algorithm chooses the shortest of these Steiner trees for the subset. Combining shortest Steiner trees for subsets in all possible ways to span the original set of given points yields all possible locally minimal Steiner trees, and the geometry of the overall shortest network can be determined.

Melzak's algorithm can take an enormous amount of time even for small problems because it considers so many possibilities. A 10-point problem, for instance, can be separated into 512 subsets of given points. Although two-point subsets do not require much work, each of the 45 subsets of eight points has two million replacement sequences. Furthermore, there are more than 18,000 ways to recombine the subsets into trees.

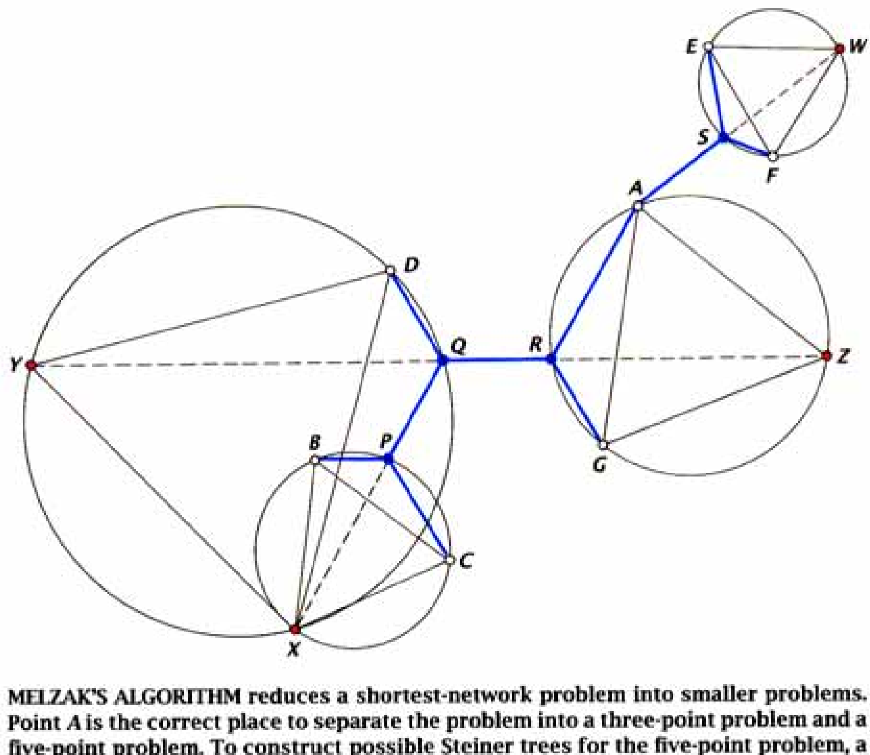
To be sure, investigators have found better ways to organize the computation and increase the speed of the algorithm. Instead of considering the problem's geometry, they focus on possible patterns of connections in the network—what is known as the network's topology. A topology specifies which points are connected to one another, but not the actual locations of Steiner points. Assuming a certain topology, one can find an appropriate replacement sequence relatively quickly. This organization greatly increases the speed of computing shortest Steiner trees for the subsets. For an eight-point subset, for example, the algorithm needs to consider only about 10,000 different topologies rather than two million different replacement sequences.

Because the number of topologies grows rapidly with the size of the subset, Steiner problems might become more manageable if only very small subsets of the set of given points needed to be considered. Experiments with Melzak's algorithm suggest that the shortest network for more than six random points can usually be separated into shortest networks for smaller sets of points. By considering special arrangements of points called ladders, however, Fan R. K. Chung of Bell Communications Research and one of us (Graham) demonstrated that there are arbitrarily large sets of points for which the shortest Steiner tree cannot be separated. A ladder is an arrangement in which all the given points are equally spaced along two parallel lines. A general solution was discovered for this quite special Steiner problem. It showed that the number of Steiner points in a shortest Steiner tree for a ladder with an odd number of "rungs" is the maximum: the number of given points minus two. Such a Steiner tree cannot be separated because the placement of every Steiner point requires that every given point be considered simultaneously. Thus one cannot simply declare a cutoff on the size of subsets considered in Melzak's algorithm.

A number of investigators improved on Melzak's algorithm by finding subtler ways to reduce the amount of work [*see illustration on next page*]. These methods prune, or eliminate, parts of the computation that would only yield long networks. New pruning



SHORTEST NETWORK for three points *A*, *B* and *C* can be constructed. An equilateral triangle *ACX* (*green*) is erected along the longest side of triangle *ABC* and then circumscribed with a circle (*yellow*). The intersection of the circle and a line segment from *B* to *X*, the equilateral triangle's third vertex, marks point *P*, known as the Steiner point. Joining points *A*, *B* and *C* to *P* forms three angles of 120 degrees and yields the shortest network. The length of line segment *BX* equals the network's length.

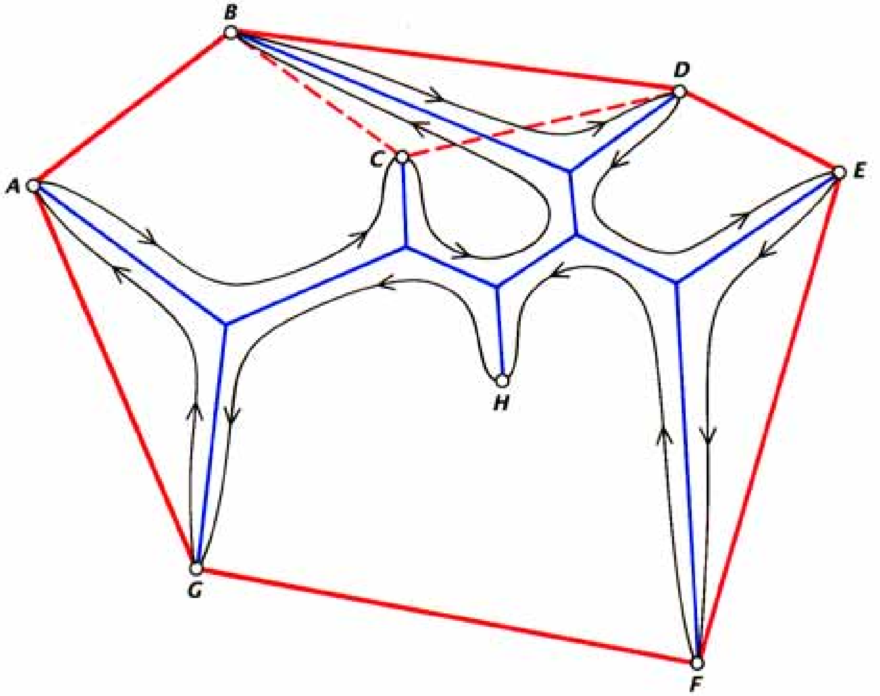


MELZAK'S ALGORITHM reduces a shortest-network problem into smaller problems. Point *A* is the correct place to separate the problem into a three-point problem and a five-point problem. To construct possible Steiner trees for the five-point problem, a pair of points (*B* and *C*, for instance) can be replaced with a single point (*X* in this case) by constructing an equilateral triangle on one side of *B* and *C*. The problem is thus reduced to four points: *X*, *D*, *G* and *A*. A pair of these points can then be replaced—in this case, first *D* and *X* with *Y* and then *G* and *A* with *Z*. Each of the equilateral triangles that results (*XDY*, *AGZ* and *BCX*) is circumscribed with a circle. The points at which a line from *Y* to *Z* intersects two of the circles give the Steiner points *Q* and *R*, and the intersection of a line from *X* to *Q* with the remaining circle determines the Steiner point *P*. Since the best partitioning and pairing cannot be determined in advance, all possibilities must be considered to find the shortest tree.

techniques have reduced computation times substantially. Programs based on Melzak's algorithm, such as one written in 1969 by Ernest J. Cockayne of the University of Victoria, could solve all nine-point problems and some 12-point problems in about half an hour. A program written recently by Cockayne and a colleague at Victoria, Denton E. Hewgill, uses a powerful pruning technique introduced by Pawel Winter of the University of Copenhagen to solve all 17-point problems and most randomly generated 30-point problems in a few minutes. Winter's pruning method is so successful at eliminating possible topologies that the bulk of the computation is now the recombination of solutions for subsets.

For any of these programs, however, running times can depend quite sensitively on the geometry as well as on the number of points. Moreover, the computation time of even the most sophisticated algorithm grows exponentially with the number of points, and Steiner problems of 100 points are still well out of reach. Will an efficient algorithm ever be found to compute solutions for large Steiner problems?

Advances in theoretical computer science have convinced most investigators that the existing algorithms for Steiner problems cannot be substantially improved. This theory assigns a size to each instance, or example, of a problem. For Steiner problems there is a natural measure of size: the number of given points. One then considers the number of basic computer operations—such as additions, subtractions or comparisons—an algorithm may need in order to solve an instance of a certain size. Since different instances of the same size may require different numbers of operations, one looks at the maximum number of operations as a function of size. If the number of operations increases by the size of the instance (n) to some power, as in the expressions n^2, $5n$ or $6n + n^{10}$, the procedure is called a polynomial-time algorithm. These algorithms are considered efficient, at least in a theoretical sense. If the number of operations increases exponentially with size, as in cases such as 2^n, 5^n or $3n^2 \times 4^n$, the procedure is known as an exponential-time algorithm.



PRUNING METHODS increase the efficiency of algorithms for finding short networks. One way to prune, or rule out, possible networks (devised by Cockayne) is to consider the order in which a rubber band (*red*) stretched around the set of given points touches them. The rubber band touches all the points except *C* and *H*, but *C* can be included in the sequence because the angle formed by point *C* and two consecutive points in contact with the rubber band measures at least 120 degrees. The order of points is then *ABCDEFG*. An unbroken path (*purple*) traced around a possible network (*blue*) touches the points in the order *ACBDEFHG*. Since *B* and *C* are reversed with respect to the order established by the rubber band, this network can be pruned.

Although for very small problems both polynomial- and exponential-time algorithms are practical, for large problems the solution times of exponential-time algorithms are so slow that these algorithms are hopelessly impractical [see "The Efficiency of Algorithms," by Harry R. Lewis and Christos H. Papadimitriou; SCIENTIFIC AMERICAN, January, 1978]. For sufficiently large problems a polynomial-time algorithm executed on even the slowest machine will yield an answer sooner than an exponential-time algorithm running on a supercomputer.

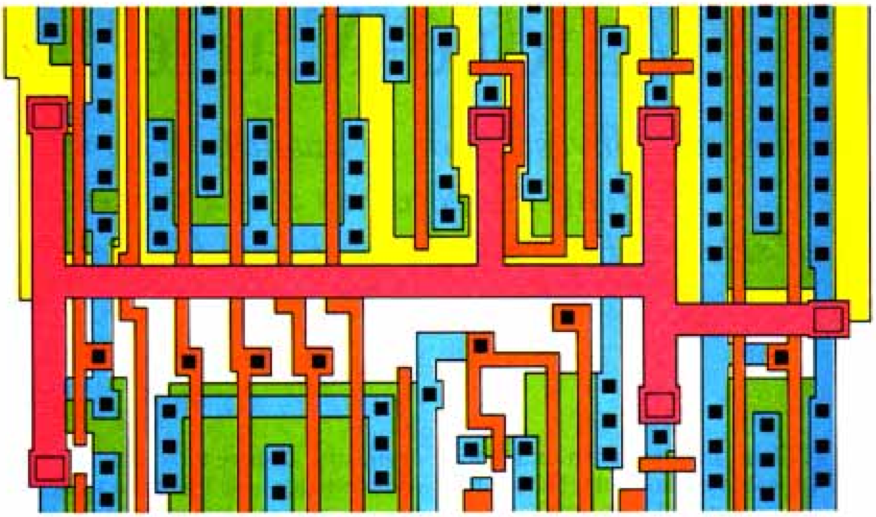
Even though exponential-time algorithms have been found for the Steiner problem (Melzak's algorithm, for example), no polynomial-time algorithms have yet been found. The prospects for an efficient algorithm are not good. In 1971 Stephen A. Cook of the University of Toronto proved that if a polynomial-time algorithm could be found for any single problem in a group now known as NP-hard problems, that algorithm could be used to solve all other problems efficiently in a large class of hard problems including NP-hard problems. Later one of the authors (Graham), working with Michael R. Garey and David S. Johnson of the AT&T Bell Laboratories, proved that the Steiner problem is an NP-hard problem. Since all NP-hard problems have to date foiled the efforts of thousands of workers, it is considered unlikely that any NP-hard problem, including the Steiner problem, can be solved by a polynomial-time algorithm. Proving that NP-hard problems cannot be solved efficiently, however, is the preeminent problem in theoretical computer science.

Although it does not appear likely that an efficient, polynomial-time algorithm will be found for computing shortest networks, there are practical algorithms producing slightly longer networks. One example is the algorithm for solving the minimum-spanning-tree problem, which searches for the shortest network of line segments that will interconnect a set of given points without adding any new ones. To solve it one connects the two given points that are closest together, and in each subsequent step one connects the closest pair of points that can be joined without forming a closed path. After all, an edge can be removed from a closed path and leave given points still connected by the remaining edges.

Edgar N. Gilbert and Henry O. Pollak of Bell Laboratories have conjectured that the ratio of a shortest Steiner tree to a minimum spanning tree is at least

$\sqrt{3}/2$, that is, the Steiner tree's length is at most about 13.4 percent shorter than the minimum spanning tree's length. The ratio of $\sqrt{3}/2$ occurs in a simple example: three given points forming an equilateral triangle. Although the conjecture remains unproved, Chung and one of the authors (Graham) have proved that the Steiner tree is at most 17.6 percent shorter than the spanning tree.

Minimum spanning trees can often be shortened 3 to 4 percent by carefully adding Steiner points and adjusting the tree. One of the authors (Bern) has shown that this kind of inexact algorithm has some theoretical justification, since the average length of an adjusted tree will be a little less than the average length of a minimum spanning tree.



VARIATIONS of the shortest-network problem have been applied to the design of electronic circuits in order to increase operating speeds. The shortest network of horizontal and vertical wires that interconnect a set of terminals is shown in magenta. The background shows other wires and terminals arranged in deeper layers.

The minimum-spanning-tree and shortest-network problems have been applied to constructing telephone, pipeline and roadway networks. The solutions, whether approximate or exact, can provide guidelines for the layout of the networks and the necessary amounts of materials. More complicated versions of the Steiner problem can accommodate the need to avoid certain geographic features or to find the shortest connections along preexisting networks.

Perhaps the most practical application of the Steiner problem is in the design of electronic circuits. A short network of wires on an integrated circuit requires less time to charge and discharge than a long network and so increases the circuit's speed of operation. The shortest-network problem on circuits, however, involves a different kind of geometry, since wires on a circuit generally run in only two directions, vertical and horizontal.

The problem, known as the rectilinear Steiner problem, was first investigated in 1965 by Maurice Hanan of the IBM Corporation's Thomas J. Watson Research Center in Yorktown Heights, N.Y. As in the original Steiner problem, the solution to the rectilinear version is also a tree containing Steiner points and given points, but edges meet at 90 or 180 degrees. Although Steiner points could conceivably lie anywhere in the rectilinear problem, Hanan showed that it is possible to restrict the locations of Steiner points in a shortest rectilinear network. A vertical and a horizontal line are drawn through each given point, and each intersection of two lines defines a possible Steiner point. An algorithm can try all subsets of possible Steiner points in order to compute a shortest network. As the number of given points increases, however, the solution time of such a brute-force algorithm grows exponentially. More sophisticated but still exponential-time algorithms can solve rectilinear Steiner problems that have about 40 points.

A rectilinear version of the minimum-spanning-tree problem, which can be solved efficiently by the algorithm that chooses the shortest connection at each step, unless that connection forms a closed path. Frank K. Hwang of Bell Laboratories has proved that a rectilinear Steiner tree is never shorter than a rectilinear minimum spanning tree by more than one-third.

The most surprising application of the Steiner problem is in the area of phylogeny. David Sankoff of the University of Montreal and other investigators defined a version of the Steiner problem in order to compute plausible phylogenetic trees. The workers first isolate a particular protein that is common to the organisms they want to classify. For each organism they then determine the sequence of the amino acids that make up the protein and define a point at a position determined by the number of differences between the corresponding organism's protein and the protein of other organisms. Organisms with similar sequences are thus defined as being close together and organisms with dissimilar sequences are defined as being far apart. In a shortest network for this abstract arrangement of given points, the Steiner points correspond to the most plausible ancestors, and edges correspond to a relation between organism and ancestor that assumes the fewest mutations. Since the phylogenetic Steiner problem is no easier than other Steiner problems, however, the problem—except as it is applied to small numbers of organisms—has served more as a thought experiment than as a practical research tool.

Although knowledge about algorithms has progressed greatly in recent years, the shortest-network problem remains tantalizingly difficult. The problem can be stated in simple terms, and yet solutions defy analysis. A tiny variation in the geometry of a problem may appear to be insignificant, and yet it can radically alter the shortest network for the problem. This sensitivity renders even peripheral questions about shortest networks quite challenging. The shortest-network problem will continue to frustrate and fascinate us for years to come.

FURTHER READING

STEINER MINIMAL TREES. E. N. Gilbert and H. O. Pollak in *SIAM Journal on Applied Mathematics*, Vol. 16, No. 1, pages 1–29; January, 1968.

COMPANION TO CONCRETE MATHEMATICS. Z. A. Melzak. John Wiley & Sons, Inc., 1973.

AN ALGORITHM FOR THE STEINER PROBLEM IN THE EUCLIDEAN PLANE. Pawel Winter in *Networks*, Vol. 15, No. 3, pages 323–345; Fall, 1985.

STEINER PROBLEM IN NETWORKS: A SURVEY. Pawel Winter in *Networks*, Vol. 17, No. 2, pages 129–167; Summer, 1987.