Exercises

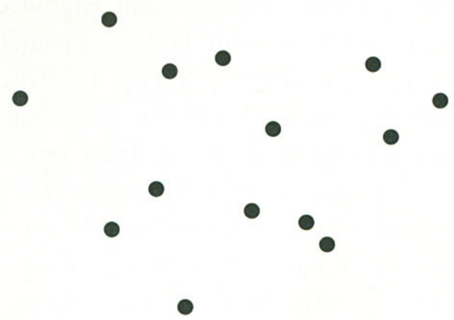
9.1 A simple on-line approximation algorithm for solving the bin packing problem is to put an object into the ith bin as long as it is available and into the $(i + 1)$-th bin if otherwise. This algorithm is called the next-fit (NF) algorithm. Show that the number of bins resulting from this FF algorithm is no more than twice the number of bins needed in an optimal solution.

9.2 Show that if the sequence of items to be considered is $1/2$, $1/2n$, $1/2$, $1/2n$, … , $1/2$ (totally $4n - 1$ items), then the NF algorithm will indeed use nearly twice the number of bins that are really required.

9.3 Show that there does not exist any polynomial time approximation algorithm for the traveling salesperson problem such that the error caused by the approximation algorithm is bounded within $\varepsilon \cdot TSP$ where ε is any constant and TSP denotes an optimal solution.

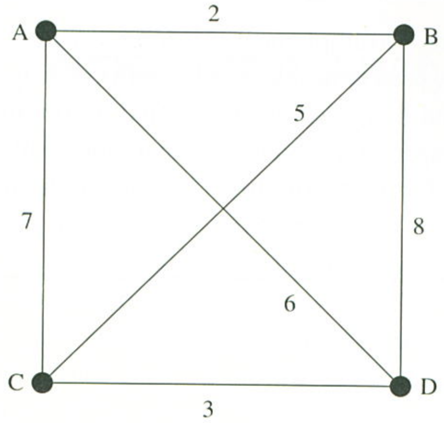
(**Hint:** Show that the Hamiltonian cycle problem can be reduced to this problem.)

9.4 Apply an approximation convex hull algorithm to find an approximate convex hull of the following set of points.

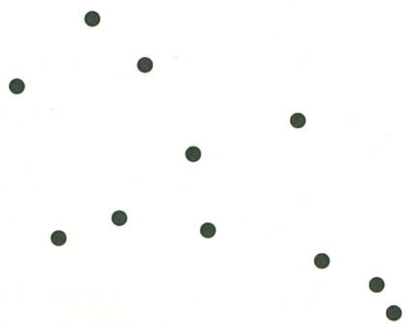


9.5 Let there be a set of points densely distributed on a circle. Apply the approximation Euclidean traveling salesperson algorithm to find an approximate tour for this set of points. Is this result also an optimal one?

9.6 Consider the four points on a square as shown below. Solve the bottleneck traveling salesperson problem approximately by the algorithm introduced in this chapter. Is the result obtained also optimal?



9.7 Use the approximation algorithm for the rectilinear m-center problem to find an approximate solution of the rectilinear 2-center problem for the following set of points.



9.8 Consider the following bottleneck optimization problem. We are given a set of points in the plane and we are asked to find k points such that among these k points, the shortest distance is maximized. This problem is shown to be NP-complete by Wang and Kuo (1988). Try to develop an approximation algorithm to solve this problem.

9.9 Read Section 12.3 of Horowitz and Sahni (1978) about approximation algorithms for scheduling independent tasks. Apply the longest processing time (LPT) rule to the following scheduling problem: There are three processors and seven tasks, where task times are $(t_1, t_2, t_3, t_4, t_5, t_6, t_7) = (14, 12, 11, 9, 8, 7, 1)$.

9.10 Write a program to implement the approximation algorithm for the traveling salesperson problem. Also write a program to implement the branch-and-bound algorithm to find an optimal solution of the traveling salesperson problem. Compare the results. Draw your own conclusion. Is it worthwhile to use such an approximation algorithm?