# DIMACS Researchers Win a Gold Medal in TSP Competition

*by Fred Rispoli*

The traveling salesperson problem (or TSP for short) is to start from a given city, visit all the cities on a particular list, and return to the initial city by traveling the shortest possible total distance. For over a decade the TSP has received an enormous amount of attention in the Discrete Mathematics and Operations Research literature. Recently many computer codes for solving the TSP have been tested on a standard set of problems or "benchmarks" known as the TSPLIB. This practice is reminiscent of a chess tournament in which computer algorithms compete against a select set of expert players, in order to rank different algorithms. Some of the TSPLIB problems were obtained from recreational sources, such as the problem of finding knight tours on a chess board. Others were obtained by considering sets of cities together with the actual distances. Still others were obtained from applications such as drilling holes on a circuit board and from setting x-ray crystallography equipment.

In the spring of 1992, a team consisting of DIMACS members David Applegate (AT&T Bell Labs), Vasek Chvatal (Rutgers) and Bill Cook (Bellcore) along with Bob Bixby (of the Center for Research on Parallel Computation, Rice University) claimed a gold medal by solving fifteen previously unsolved problems from the TSPLIB with sizes ranging from 417 to 3,038 cities. The largest previously solved TSPLIB problem had 2,392 cities. In June 1993, the team broke its own record by solving a TSPLIB problem with 4,461 cities using a parallel implementation. The computations took 27 days on a network consisting of 75 processors working in parallel (the estimated running time on a single workstation comes to about eighteen months). The main ideas used were inspired by a landmark 1954 paper of Dantzig, Fulkerson and Johnson [1] who used "cutting planes" to convert fractional solutions from the associated linear programming problem to integer solutions, and were the first to solve a non-trivial TSP with 48 cities (see below for an explanation of cutting planes). To make this approach work on very large -scale problems the DIMACS team developed a fast technique to identify the cutting planes, and a "shrinking strategy" for dividing the problem into smaller subproblems.

Somewhat surprisingly, real-life applications of the TSP have not been numerous. So why all the attention? Perhaps the best explanation is that the TSP is a problem that is natural and easy to state, but so hard to solve that it has given new meaning to the phrase "hard problem". Just ten years ago researchers used to only dream about solving some of the larger problems in the TSPLIB that can now be solved in hours, making the problem an irresistible challenge.

References


1. Dantzig, G., Fulkerson, R., Johnson, S., "Solution of a large-scale traveling salesman problem," *Operations Research*, Vol. 2 (1954), 393-410.

2. Kolata, Gina, *Math Problem, Long Baffling, Slowly Yields,* New York Times, March 21, 1991.

3. Michaels, J., and Rosen, K., eds., "The Traveling Salesman Problem" in *Applications of Discrete Mathematics,* McGraw-Hill, 1991.


## Cutting Plane Method for Solving the TSP

A full description of cutting-plane methods is quite difficult, however the intuition behind the method is just "successive approximation." The first step is to convert the TSP into a vector problem. Assume that there are n vertices (cities), and for each edge {i, j} between vertices i and j there is a cost $c_{ij}$ (distance). A traveling salesperson tour, or TS tour, is a cycle that visits each vertex exactly once (a Hamiltonian cycle). The *best* TS tour is the one with the minimum total cost (sum of costs of the edges). Now, think of {i,j}and {j,i}as different edges; for each edge {i, j} introduce a variable $x_{ij}$. Let $x_{ij} = 1$ mean that edge {i, j} is used in the traveling salesperson tour and $x_{ij} = 0$ mean that {i, j} is not used. Create a vector of 0's and 1's ordered lexicographically where, for example, $x_{12}$ precedes $x_{14}$ and $x_{24}$ precedes $x_{31}$. Then, each tour can be represented as a 0-1 vector of length n(n-1).

A small example showing how to represent a tour as a vector is shown on page 6. Assume that n = 4 for the moment.

Certainly, not all 0-1 vectors represent tours. To represent the problem correctly, we must add a number of geometric constraints. For example, $x_{12} + x_{13} + x_{14} = 1$, since exactly one edge must leave vertex 1. Similarly, $x_{12} + x_{32} + x_{42} = 1$, and so on. Geometrically, each of these equations is a "hyperplane": all the vectors representing tours are restricted to the intersection of these planes.

Finding the minimum cost TS tour in this case (n=4) is equivalent to finding a 0-1 vector which (1) satistifies all the relevant geometric constraints; and (2) minimizes the sum $c_{12}x_{12} + c_{13}x_{13} + \ldots + c_{43}x_{43}$.

*(Continued on page 10)*