

Kluwer Academic Publishers 2001
10.1007/1-4020-0611-X_1068

# Traveling salesman problem

Karla L. Hoffman<sup>1</sup>, Manfred Padberg<sup>2</sup> and Giovanni Rinaldi<sup>3</sup>

(1) George Mason University, Fairfax, Virginia, USA

(2) New York University, New York, USA

(3) CNR – IASI, Rome, Italy

---

Without Abstract

---

## INTRODUCTION

The *traveling salesman problem* (TSP) is one which has commanded much attention of mathematicians and computer scientists specifically because it is so easy to describe and so difficult to solve. The problem can simply be stated as: if a traveling salesman wishes to visit exactly once each of a list of  $m$  cities (where the cost of traveling from city  $i$  to city  $j$  is  $c_{ij}$ ) and then return to the home city, what is the least costly route the traveling salesman can take? A complete historical development of this and related problems can be found in *Hoffman and Wolfe* (1985), *Applegate et al.* (2006), and *Cook* (2011).

The importance of the TSP is that it is representative of a larger class of problems known as *combinatorial optimization problems*. The TSP problem belongs in the class of such problems known as NP-complete. Specifically, if one can find an efficient (i.e., polynomial-time) algorithm for the traveling salesman problem, then efficient algorithms could be found for all other problems in the NP-complete class. To date, however, no one has found a polynomial-time algorithm for the TSP. Does that mean that it is impossible to solve *any* large instances of such problems? To the contrary, nowadays many practical optimization problems of truly large scale are solved to optimality routinely. From 1992 to 2006 Concorde, a software created by D. Applegate, R.E. Bixby, V. Chvátal, and W.J. Cook (*Applegate et al.* 1995, 2006), solved (among many others) a traveling salesman problem which models the production of printed circuit boards having 7,397 holes (cities), a problem over the 13,509 largest cities in the U.S., one over the 24,978 cities of Sweden, and, finally, a 85,900 city problem arising from a VLSI application. So, although the question of what it is that makes a problem “difficult” may remain open, the computational record of specific instances of TSP problems coming from practical applications is optimistic.

How are such problems tackled today? Obviously, one cannot consider a brute force approach (for example, for a 16 city traveling salesman problem there are 653,837,184,000 distinct routes that would need to be evaluated). Rather than enumerating all possibilities, successful algorithms for solving the TSP problem have

been capable of eliminating most of the round-trips without ever explicitly considering them.

## FORMULATIONS

The first step to solving instances of large TSPs must be to find a good mathematical formulation of the problem. In the case of the traveling salesman problem, the mathematical structure is a graph where each city is denoted by a point (or node) and lines are drawn connecting every two nodes (called arcs or edges). Associated with every line is a distance (or cost). When the salesman can get from every city to every other city directly, then the graph is said to be *complete*. A round-trip of the cities corresponds to some subset of the lines, and is called a tour or a Hamiltonian cycle in graph theory. The length of a tour is the sum of the lengths of the lines in the round-trip.

Depending upon whether or not the direction in which an edge of the graph is traversed matters, one distinguishes the *asymmetric* from the *symmetric* traveling salesman problem. To formulate the *asymmetric* TSP on  $m$  cities, one introduces zero-one variables

$$x_{ij} = \begin{cases} 1 & \text{if the edge } i \rightarrow j \text{ is in the tour} \\ 0 & \text{otherwise} \end{cases}$$

and, given the fact that every node of the graph must have exactly one edge pointing towards it and one pointing away from it, one obtains the classic assignment problem. These constraints alone are not enough since this formulation would allow “subtours,” that is, it would allow disjoint loops to occur. For this reason, a proper formulation of the asymmetric traveling salesman problem must remove these subtours from consideration by the addition of “subtour elimination” constraints. The problem then becomes

$$\begin{aligned} \min \quad & \sum_{j=1}^m \sum_{i=1}^m c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^m x_{ij} = 1 \quad \text{for } i = 1, \dots, m \\ & \sum_{i=1}^m x_{ij} = 1 \quad \text{for } j = 1, \dots, m \\ & \sum_{i \in K} \sum_{j \in K} x_{ij} \leq |K| - 1 \quad \text{for all } K \subset \{1, \dots, m\} \end{aligned}$$

where  $K$  is any nonempty proper subset of the cities  $1, \dots, m$ . The cost  $c_{ij}$  is allowed to be different from the cost  $c_{ji}$ . Note that there are  $m(m - 1)$  0-1 variables in this formulation.

To formulate the *symmetric* traveling salesman problem, one notes that the direction traversed is immaterial, so that  $c_{ij} = c_{ji}$ . Since direction does not now matter, one can

consider the graph where there is only one arc (undirected) between every two nodes. Thus, we let  $x_j \in \{0,1\}$  be the decision variable where  $j$  runs through all edges  $E$  of the undirected graph and  $c_j$  is the cost of traveling that edge. To find a tour in this graph, one must select a subset of edges such that every node is contained in exactly two of the edges selected. Thus, the problem can be formulated as a 2-matching problem in a graph having  $m(m-1)/2$  0-1 variables, that is, half of the number of the previous formulation. As in the asymmetric case, subtours must be eliminated through subtour elimination constraints. The problem can therefore be formulated as

$$\begin{aligned} \min \quad & (1/2) \sum_{j=1}^m \sum_{k \in J(j)} c_k x_k \\ \text{s.t.} \quad & \sum_{k \in J(j)} x_k = 2 \quad \text{for all } j = 1, \dots, m \\ & \sum_{j \in E(K)} x_j \leq |K| - 1 \quad \text{for all } K \subset \{1, \dots, m\} \\ & x_j = 0 \text{ or } 1 \quad \text{for all } j \in E, \end{aligned}$$

where  $J(j)$  is the set of all undirected edges connected to node  $j$  and  $E(K)$  is the subset of all undirected edges connecting the cities in any proper, nonempty subset  $K$  of all cities. Of course, the symmetric problem is a special case of the asymmetric one, but practical experience has shown that algorithms for the asymmetric problem perform, in general, badly on symmetric problems. Thus, the latter need a special formulation and solution treatment. In addition, as an ATSP instance can be easily turned into a symmetric one with twice the number of nodes, any algorithm for STSP can be used to solve an ATSP.

---

## ALGORITHMS

Exact approaches to solving such problems require algorithms that generate both a *lower bound* and an *upper bound* on the true minimum value of the problem instance. Any round-trip tour that goes through every city exactly once is a feasible solution with a given cost that cannot be smaller than the minimum cost tour. Algorithms that construct feasible solutions, and thus upper bounds for the optimum value, are called *heuristics*. These solution strategies produce answers but often without any quality guarantee as to how far off they may be from the optimal answer. Heuristic algorithms that find a feasible solution in a single attempt are called *constructive heuristics* while algorithms that iteratively modify and try to improve some given starting solution are called *improvement heuristics*. When the solution one obtains is dependent on the initial starting point of the algorithm, the same algorithm can be used multiple times from various (random) starting points. Often, if one needs a solution quickly, one may settle for a well-designed heuristic algorithm that has been shown empirically to find “near-optimal” tours to many TSP problems. Research by *Golden and Stewart* (1985), *Jünger, Reinelt and Rinaldi* (1994), *Johnson and McGeoch* (2002), and *Applegate et al.* (2006) describes algorithms that find solutions to extremely large TSPs (problems with hundreds of thousands, or even millions of

variables) to within 1 or 2% of optimality in very reasonable times. The heuristic algorithm of Lin and Kernighan appears so far to be the most effective in terms of solution quality, in particular with the variant proposed by *Helsgaun* (2000), which was able to find, for the first time, the optimal solution (although without a quality guarantee) of several instances of *TSPLIB*, a well known library of TSP problems described in *Reinelt* (1991). For genetic algorithmic approaches to the TSP, see *Potvin* (1996); for simulated annealing approaches see *Aarts, Korst and Laarhoven* (1988); for neural net approaches, see *Potvin* (1993); for tabu search approaches, see *Fiechter* (1990); and for a very effective evolutionary algorithm, see *Nagata* (2006). Probabilistic analysis of heuristics are discussed in *Karp and Steele* (1985); performance guarantees for heuristics are given in *Johnson and Papadimitriou* (1985) and *Arora* (2002), where an amazing result concerning the polynomial-time approximability is described for *Euclidean TSP* instances (where the nodes are points in the plane and the traveling costs are the Euclidean distances between the points). For an analysis of the heuristics for the ATSP, see *Johnson et al.* (2002).

In order to know about the closeness of the upper bound to the optimum value, one must also know a lower bound on the optimum value. If the upper and lower bounds coincide, a proof of optimality is achieved. If not, a conservative estimate of the true relative error of the upper bound is provided by the difference of the upper and the lower bound divided by the lower bound. Thus, one needs both upper and lower bounding techniques to find provably optimal solutions to hard combinatorial problems or even to obtain solutions meeting a quality guarantee.

So how does one obtain and improve the lower bound? A *relaxation* of an optimization problem is another optimization problem whose set of feasible solutions properly contains all feasible solution of the original problem and whose objective function value is less than or equal to the true objective function value for points feasible to the original problem. Thus we replace the “true” problem by one with a larger feasible region but that is more easily solvable. This relaxation is continually refined so as to tighten the feasible region so that it more closely represents the true problem. The standard technique for obtaining lower bounds on the TSP problem is to use a relaxation that is easier to solve than the original problem. These relaxations can have either discrete or continuous feasible sets. Several relaxations have been considered for the TSP. Among them are the *n-path relaxation*, the *assignment relaxation*, the *2-matching relaxation*, the *1-tree relaxation*, and the *linear programming relaxation*. For randomly generated asymmetric TSPs, problems having up to 7,500 cities have been solved, in the early '90s, using an assignment relaxation which adds subtours within a branch and bound framework and which uses an upper bounding heuristic based on subtour patching, (*Miller and Pekny*, 1991). For the symmetric TSP, the 1-tree relaxation and the 2-matching relaxations have been most successful. These relaxations have been embedded into a branch-and-bound framework.

The process of finding constraints that are violated by a given relaxation, is called a *cutting plane* technique and all successes for large TSP problems have used cutting planes to continuously tighten the formulation of the problem. To obtain a tight relaxation the inequalities utilized as cutting planes in many computational approaches to the TSP are often *facet-defining* inequalities.

One of the simplest classes of cuts that have been shown to define *facets* of the underlying TSP polytope is the subtour elimination cut. Besides these constraints, *comb* inequalities, *clique tree* inequalities, *path*, *wheelbarrow* and *bicycle* inequalities, *ladder*, *crown*, *domino* and many other inequalities have also been shown to define facets of this polytope. The underlying theory of facet generation for the symmetric traveling salesman problem is provided in *Grötschel and Padberg (1985)*, *Jünger, Reinelt and Rinaldi (1994)* and *Naddef (2002)*; analogous results for the ATSP polytope are provided in *Balas and Fischetti (2002)*. The algorithmic descriptions of how these inequalities are used in cutting plane approaches are discussed in *Padberg and Rinaldi (1991)*, in *Jünger, Reinelt and Rinaldi (1994)*, and in *Applegate et al. (2006)* where it is also shown how the polynomial-time equivalence between optimization and separation can be turned into a powerful algorithmic tool to generate inequalities not necessarily belonging to one of the known types.

Cutting plane procedures can then be embedded into a tree search in an algorithmic framework referred to as *branch and cut* and proposed in *Padberg and Rinaldi (1991)* where it is shown how such approach made it possible to solve some still unsolved instances of sizes up to 2,392 nodes. Some of the largest TSP problems solved have used parallel processing to assist in the search for optimality. This is the case of the software *Concorde* where all the known algorithmic ideas for the TSP (and many new ones) have been carefully implemented. With this code *Applegate et al. (2006)* managed to solve all problems of the *TSPLIB* to optimality; for the largest one, of 85,900 nodes, they used 96 workstations for a total of 139 years of CPU time.

As our understanding of the underlying mathematical structure of the TSP problem improves, and with the continuing advancement in computer technology, it is likely that many difficult and important combinatorial optimization problems will be solved using a combination of cutting plane generation procedures, heuristics, variable fixing through logical implications and reduced costs, and tree search.

---

## APPLICATIONS

One might ask, however, whether the TSP problem is important enough to have received all of the attention it has. Much of the attention that the problem has received is because it is a relatively simple problem to describe and yet a difficult (from a complexity viewpoint) optimization problem to solve. However, there are important cases of practical problems that can be formulated as TSP problems and many other problems are generalizations of this problem. Besides the drilling of printed circuits boards described above, problems having the TSP structure occur in the analysis of the structure of crystals (*Bland and Shallcross, 1987*), in the overhauling of gas turbine engines (*Pante, Lowe and Chandrasekaran, 1987*), in material handling in a warehouse (*Ratliff and Rosenthal, 1981*), in cutting stock problems (*Garfinkel, 1977*), in the clustering of data arrays (*Lenstra and Rinooy Kan, 1975*), in the sequencing of jobs on a single machine (*Gilmore and Gomory, 1964*), in the assignment of routes for planes of a specified fleet (*Boland, Jones, and Nemhauser, 1994*) and in genome sequencing (*Ben-Dor and Chor, 1997*; *Ben-Dor, Chor, and Pelleg, 2000*). Related variations on the traveling salesman problem

include the resource constrained traveling salesman problem which has applications in scheduling with an aggregate deadline (*Pekny and Miller, 1991*). This paper also shows how the prize collecting traveling salesman problem (*Balas, 2002*) and the orienteering problem (*Golden, Levy and Vohra, 1987; Fischetti, Salazar and Toth, 2002*) are special cases of the resource constrained TSP. Most importantly, the traveling salesman problem often comes up as a subproblem in more complex combinatorial problems, the best known and important one of which is the *vehicle routing problem*. This is the problem of determining for a fleet of vehicles which customers should be served by each vehicle and in what order each vehicle should visit the customers assigned to it. For relevant surveys, see *Christofides (1985)*, *Fisher (1987)*, and the book *The Vehicle Routing Problem*, edited by *Toth and Vigo (2001)*.

For the interested reader seeking more information on this important problem, we suggest the seminal paper on the problem by *Dantzig, Fulkerson and Johnson (1954)*, the books by *Lawler, Lenstra, Rinnooy Kan and Shmoys (1985)*, by *Reinelt (1994)* and by *Gutin and Punnen (2002)*, and the survey and the annotated bibliography by *Jünger, Reinelt and Rinaldi (1994, 1997)*, which summarize most of the research up through 2002 and provide extensive references. For a deep understanding of how algorithms for TSP work, we recommend the book by *Applegate et al (2006)*. Besides providing a wide overview on TSP history and on its applications, the volume gives a detailed description of how all components of the *Concorde* software are built: a most valuable source for algorithm designers. Finally, for a more general audience we recommend the book by *Cook (2011)* which requires almost no mathematical background to read, but very nicely and completely describes the TSP from several interesting view points. The computer program *Concorde*, the *TSPLIB* and many other sources of information on the TSP are available electronically at a web site that can be easily located through web search.

See **Assignment problem; Branch and bound; Branch and cut; Chinese postman; Combinatorics; Combinatorial and integer optimization; Computational complexity; Graph theory; Heuristics; Linear programming; Networks.**

---

## References

- [1]. Aarts, E.H.L., Korst, J.H.M., and Laarhoven, P.J.M. (1988). "A Quantitative Analysis of the Simulated Annealing Algorithm: A Case Study for the Traveling Salesman Problem," *Jl. Stats. Phys.* 50, 189–206
- [2]. Applegate, D., Bixby, R.E., Chvátal, V., and Cook, W. (1995). "Finding cuts in the TSP(A preliminary report) DIMACS Technical Report 95-05, Rutgers University, New Brunswick, USA.
- [3]. Applegate, D., Bixby, R.E., Chvátal, V., and Cook, W. (2006). *The Traveling Salesman Problem: A Computational Study*, Princeton University Press,

Princeton, USA.

- [4] Arora, S. (2002). "Approximation Algorithms for Geometric TSP," in Gutin, G. and A.P. Punnen, eds., *The Traveling Salesman Problem and its Variations*, Kluwer, The Netherlands, 207-222.
- [5]. Balas, E. (2002). "The Prize Collecting Traveling Salesman Problem and its Applications," in Gutin, G. and A.P. Punnen, eds., *The Traveling Salesman Problem and its Variations*, Kluwer, The Netherlands, 663-696.
- [6] Balas, E. and M. Fischetti (2002). "Polyhedral Theory for the Asymmetric Traveling Salesman Problem," in Gutin, G. and A.P. Punnen, eds., *The Traveling Salesman Problem and its Variations*, Kluwer, The Netherlands, 117-168.
- [7] Ben-Dor, A. and B. Chor (1997). "On Constructing Radiation Hybrid Maps," *Journal of Computational Biology* 4, 517-533.
- [8] Ben-Dor, A., B. Chor and D. Peleg (2000). "RHO-Radiation Hybrid Ordering," *Genome Research* 10, 365-378.
- [9]. Bland, R.E. and Shallcross, D.F. (1987). "Large Traveling Salesman Problem Arising from Experiments in X-ray Crystallography: a Preliminary Report on Computation," Technical Report No. 730, School of OR/ IE, Cornell University, Ithaca, New York.
- [10]. Burkard, R.E., Deineko, V.G., van Dal, R., van der Veen, J.A.A., and Woeginger, G.J. (1998). "Well-Solvable Cases of the Traveling Salesman Problem: A Survey," *SIAM Review* 40, 496–546.
- [11]. Cook, W. (2011). *In Pursuit of the Salesman: Mathematics at the Limits of Computation*, Princeton University Press, Princeton, USA.
- [12]. Dantzig, G.B., Fulkerson, D.R., and Johnson, S.M. (1954). "Solution of a Large-scale Traveling Salesman Problem," *Operations Research* 2, 393–410.
- [13]. Fiechter, C.N. (1990). "A Parallel Tabu Search Algorithm for Large Scale Traveling Salesman Problems," Working Paper 90/1, Department of Mathematics, Ecole Polytechnique Federale de Lausanne, Switzerland.
- [14]. Fisher, M.L. (1988). "Lagrangian Optimization Algorithms for Vehicle Routing Problems," *Operational Research '87*, G.K. Rand, ed., 635–649.
- [15]. Golden, B.L., Levy, L., and Vohra, R. (1987). "The Orienteering Problem," *Naval Research Logistics* 34, 307–318.

- [16]. Golden, B.L. and Stewart, W.R. (1985). "Empirical Analysis of Heuristics," in *The Traveling Salesman Problem*, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, eds., John Wiley, 207–250.
- [17]. Grötschel, M. and Holland, O. (1991). "Solution of Large Scale Symmetric Traveling Salesman Problems," *Mathematical Programming* 51, 141–202.
- [18]. Grötschel, M. and Padberg, M.W. (1985). "Polyhedral Theory," in *The Traveling Salesman Problem*, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, eds., John Wiley, 251–306.
- [19]. Gutin, G. and A.P. Punnen, eds. (2002). *The Traveling Salesman Problem and its Variations*, Kluwer, The Netherlands.
- [20] Helsgun, K. (2000). "An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic," *European Journal of Operational Research* 126, 106-130.
- [21]. Hoffman, A.J. and Wolfe, P. (1985). "History" in *The Traveling Salesman Problem*, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, eds., John Wiley, 1–16.
- [22]. Johnson, D.S. and Papadimitriou, C.H. (1985). "Performance Guarantees for Heuristics," in *The Traveling Salesman Problem*, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, eds., John Wiley, 145–180.
- [23]. Johnson, D.S., Gutin, G., McGeoch, L.A., Yeo, A., Zhang, W. and Zverovitch, A. (2002). "Experimental Analysis of Heuristics for the ATSP", in Gutin, G. and A.P. Punnen, eds., *The Traveling Salesman Problem and its Variations*, Kluwer, The Netherlands, 485-488.
- [24]. Johnson, D.S. and L.A. McGeoch (2002). "Experimental Analysis of Heuristics for the STSP", in Gutin, G. and A.P. Punnen, eds., *The Traveling Salesman Problem and its Variations*, Kluwer, The Netherlands, 369-444.
- [25]. Jünger, M., Reinelt, G., and Rinaldi, G. (1994). "The Traveling Salesman Problem," in M. Ball, T. Magnanti, C. Monma and G. Nemhauser, eds., *Handbook on Operations Research and the Management Sciences*, North Holland, 225–330.
- [26]. Jünger, M., Reinelt, G., and Rinaldi, G. (1997). "The Traveling Salesman Problem," in M. Dell'Amico, F. Maffioli, and S. Martello, eds., *Annotated Bibliographies in Combinatorial Optimization*, Wiley, New York, 199-221.
- [27]. Karp, R. and Steele, J.M. (1985). "Probabilistic Analysis of Heuristics," in *The Traveling Salesman Problem*, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, eds., John Wiley, 181–205.



- [28]. Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., and Shmoys, D.B., eds. (1985). *The Traveling Salesman Problem*, John Wiley, Chichester, UK.
- [29]. Miller, D. and Pekny, J. (1991). "Exact Solution of Large Asymmetric Traveling Salesman Problems," *Science* 251, 754–761.
- [30]. Nagata, Y. (2006). "New EAX Crossover for Large TSP Instances," *LNCS* 4193, 372-381.
- [31] Naddef, D. (2002). "Polyhedral Theory and Branch-and-Cut Algorithm for the Symmetric TSP", in Gutin, G. and A.P. Punnen, eds., *The Traveling Salesman Problem and its Variations*, Kluwer, The Netherlands, 21-116.
- [32]. Padberg, M.W. and Grötschel, M. (1985). "Polyhedral Computations," in *The Traveling Salesman Problem*, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, eds., John Wiley, 307–360.
- [33]. Padberg, M.W. and Rinaldi, G. (1991). "A Branch and Cut Algorithm for the Resolution of Large-scale Symmetric Traveling Salesmen Problems," *SIAM Review* 33, 60–100.
- [34]. Potvin, J.V. (1993). "The Traveling Salesman Problem: A Neural Network Perspective," *INFORMS J. Computing* 5, 328–348.
- [35]. Potvin, J.V. (1996). "Genetic Algorithms for the Traveling Salesman Problem," *Annals of Operations Research* 63, 339–370.
- [36]. Ratliff, H.D. and Rosenthal, A.S. (1981). "Order-Picking in a Rectangular Warehouse: A Solvable Case for the Traveling Salesman Problem," *PDRC Report Series No. 81-10*. Georgia Institute of Technology, Atlanta, Georgia.
- [37]. Reinelt, G. (1991). "TSPLIB—A traveling salesman library," *ORSA J. Computing* 3, 376–384.
- [38]. Reinelt, G. (1994). *The Traveling Salesman: Computational Solutions for TSP Applications*, Springer-Verlag, Berlin.
- [39]. Toth, P. and Vigo, D. (2001). *The Vehicle Routing Problem*, SIAM, Philadelphia, USA.