Journal of the Operational Research Society (2010) 61, 35–40

www.palgrave-journals.com/jors/

A concise guide to the Traveling Salesman Problem

G Laporte*

Canada Research Chair in Distribution Management, HEC Montréal, Canada

The *Traveling Salesman Problem* (*TSP*) is one of the most famous problems in combinatorial optimization. Hundreds of papers have been written on the TSP and several exact and heuristic algorithms are available for it. Their concise guide outlines the most important and best algorithms for the symmetric and asymmetric versions of the TSP. In several cases, references to publicly available software are provided.

Journal of the Operational Research Society (2010) **61,** 35–40. doi:10.1057/jors.2009.76

Keywords: Traveling Salesman Problem; exact algorithms; heuristics; software

1. Introduction

The *Traveling Salesman Problem* (*TSP*) is arguably the most famous problem in combinatorial optimization. Simply stated, it consists of determining a shortest tour passing exactly once through each of the n vertices of a graph (such a tour is called *Hamiltonian*). In its most common interpretation, the TSP is the problem of constructing a shortest salesman tour through n cities. The popularity of the TSP derives partly from the contrast between the simplicity of its statement and its computational complexity. What makes this problem so important is also the wealth of its applications in manufacturing (namely in drilling operations and in Very Large-Scale Integration (VLSI) placement problems), in distribution management and in scheduling. For example, in the classical *Vehicle Routing Problem* (see, eg, Toth and Vigo, 2002) each vehicle route is a Hamiltonian tour on a subset of vertices and can therefore be optimized separately by means of any of the available algorithms for the TSP. The study of the TSP has also led to the emergence of several of the popular optimization techniques routinely used in operational research. Common heuristics like 2-opt and 3-opt, and exact algorithms such as branch-and-bound, branch-and-cut and branch-and-cut-and-price all originate from the study of the TSP.

Hundreds of articles and several surveys and books have been written on the TSP. In particular, the three books by Lawler *et al* (1985), Gutin and Punnen (2002) and Applegate *et al* (2006) and the survey of Jünger *et al* (1995) provide a great wealth of information on this problem. So much theory has been developed for the TSP and so many algorithms are available that it is often difficult to find one's way through the scientific literature. Also, researchers often do not know where to look for the best algorithms. Given the abundant and ever growing literature on the TSP, it would be unrealistic to aim for an exhaustive treatment of the subject in a review paper. Rather, I have opted to write a concise guide focusing on some of the best and most popular algorithms, with a strong bias toward those that are simple to implement or are publicly available on the Web. This guide also provides pointers to some of the most important references on the TSP. I hope my selection will prove valuable to students, researchers and practitioners.

Some notation is necessary at this stage. The TSP is defined on a complete undirected graph $G=(V, E)$ if it is *symmetric* or on a directed graph $G=(V, A)$ if it is *asymmetric*. The *set* $V=\{1,\ldots,n\}$ is the vertex set, $E=\{(i, j): i, j \in V, i<j\}$ is an *edge set* and $A=\{(i, j): i, j \in V, i \neq j\}$ is an *arc set*. A cost matrix $C=(c_{ij})$ is defined on E or on A. The cost matrix satisfies the *triangle inequality* whenever $c_{ij} \leqslant c_{ik} + c_{kj}$, for all i, j, k. In particular, this is the case of planar problems for which the vertices are points $P_i=(X_i, Y_i)$ in the plane, and $c_{ij}=\sqrt{(X_i-X_j)^2+(Y_i-Y_j)^2}$ is the *Euclidean distance*. The triangle inequality is also satisfied if c_{ij} is the length of a shortest path from i to j on G.

The remainder of this article is organized as follows. The next two sections describe exact and heuristic algorithms for the symmetric TSP, while the asymmetric case is handled in a similar way in the following two sections. Conclusions follow.

2. Exact algorithms for the symmetric TSP

Many formulations are available for the TSP (see, for example, the recent surveys of Orman and Williams (2006) and Öncan *et al* (2009)). Among these, the Dantzig *et al* (1954) formulation is one of the best and most important. It lies at the basis of the most efficient algorithms in use today. Incidentally, an early description of Concorde, which is recognized as the most performing exact algorithm currently available, was published under the title 'Implementing the Dantzig–Fulkerson–Johnson algorithm for large traveling

**Correspondence: G Laporte, Canada Research Chair in Distribution Management, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada.*
E-mail: gilbert@crt.umontreal.ca

salesman problems' (Applegate *et al*, 2003). This formulation associates a binary variable x_{ij} with each edge (i, j), equal to 1 if and only if the edge appears in the optimal tour. The formulation is as follows.

$$\text{(STSP) minimize} \sum_{i<j} c_{ij}x_{ij} \tag{1}$$

subject to

$$\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2 \qquad (k \in V) \tag{2}$$

$$\sum_{i,\, j \in S} x_{ij} \leqslant |S| - 1 \qquad (S \subset V, 3 \leqslant |S| \leqslant n-3) \tag{3}$$

$$x_{ij} = 0 \text{ or } 1 \qquad (i, j) \in E. \tag{4}$$

In this formulation, constraints (2), (3) and (4) are referred to as degree constraints, subtour elimination constraints and integrality constraints, respectively. In the presence of (2), constraints (3) are algebraically equivalent to the connectivity constraints

$$\sum_{i \in S,\, j \in V \setminus S \text{ or } i \in V \setminus S,\, j \in S} x_{ij} \geqslant 2 \quad (S \subset V, 3 \leqslant |S| \leqslant n-3). \tag{5}$$

When this formulation was first introduced, the simplex method was in its infancy and no algorithms were available to solve integer linear programs. Even in the absence of integrality constraints, the exponential number of subtour elimination constraints precludes a direct application of this formulation for all but trivial instances. The authors therefore used a strategy consisting of initially relaxing constraints (3) and the integrality requirements, which were gradually reintroduced after visually examining the solution to the relaxed problem. They were thus capable of optimally solving a 42-vertex instance containing cities in the continental United States.

Martin (1966) used a similar approach. Initially he did not impose upper bounds on the x_{ij} variables and imposed subtour elimination constraints on all sets $S = \{i, j\}$ for which j is the closest neighbour of i. Integrality was reached by applying the 'Accelerated Euclidean algorithm', an extension of the 'Method of integer forms' (Gomory, 1963). However, his method was not fully automated and required visual inspection of the solution obtained at each iteration in order to determine which constraints to introduce. Using this approach the 42-vertex problem was solved after three iterations. To my knowledge, Miliotis (1976, 1978) was the first to devise a fully automated algorithm based on constraint relaxation and using either branch-and-bound or Gomory cuts to reach integrality. Land (1979) later puts forward a cut-and-price algorithm combining subtour elimination constraints, Gomory cuts and column generation, but no branching. This algorithm was capable of solving nine Euclidean 100-vertex instances out of 10.

Table 1 Computation times for Concorde

n	*Type*	*Sample size*	*Mean CPU seconds*
100	random	10 000	0.7
500	random	10 000	50.2
1000	random	1000	601.6
2000	random	1000	14065.6
2500	random	1000	53737.9

It has long been recognized that the linear relaxation of STSP can be strengthened through the introduction of valid inequalities. Thus, Edmonds (1965) introduced the 2-*matching inequalities*, which were then generalized to *comb inequalities* (Chvátal, 1973). Some generalizations of comb inequalities, such as *clique tree inequalities* (Grötschel and Pulleyblank, 1986) and path inequalities (Cornuéjols *et al*, 1985) turn out to be quite effective. Several other less powerful valid inequalities are described in Naddef (2002). In the 1980s a number of researchers have integrated these cuts within relaxation mechanisms and have devised algorithms for their separation. This work, which has fostered the growth of polyhedral theory and of branch-and-cut, was mainly conducted by Padberg and Hong (1980), Crowder and Padberg (1980), Grötschel and Padberg (1985), Padberg and Grötschel (1985), Padberg and Rinaldi (1987, 1991), and Grötschel and Holland (1991). The largest instance solved by the latter authors was a drilling problem of size $n = 2392$.

The culmination of this line of research is the development of Concorde by Applegate *et al* (2003, 2006), which is today the best available solver for the symmetric TSP. It is freely available at www.tsp.gatech.edu. This computer program is based on branch-and-cut-and-price, meaning that both some constraints and variables are initially relaxed and dynamically generated during the solution process. The algorithm uses 2-matching constraints, comb inequalities and certain path inequalities. It makes use of sophisticated separation algorithms to identify violated inequalities. A detailed description of Concorde can be found in the book by Applegate *et al* (2006), which was awarded the 2007 INFORMS Frederick W. Lanchester Prize. Table 1 summarizes some of the results reported by Applegate *et al* (2006) for randomly generated instances in the plane. All tests were run on a cluster of compute nodes, each equipped with a 2.66 GHz IntelXeon processor and 2 Gbyte of memory. The linear programming solver used was CPLEX 6.5.

It can be seen that Concorde is quite reliable for this type of instances. All small TSPLIB instances ($n \leqslant 1000$) were solved within 1 min on a 2.4 GHz ADM Opteron processor. On 21 medium-size TSPLIB instances ($1000 \leqslant n \leqslant 2392$), the algorithm converged 19 times to the optimum within a computing time varying between 5.7 and 3345.3 s. The two exceptions required 13999.9 s and 18226404.4 s. The largest instance now solved optimally by Concorde arises from a VLSI application and contains 85 900 vertices (Applegate *et al*, 2009).

3. Heuristics for the symmetric TSP

A great variety of heuristics are available for the symmetric TSP. Thorough comparative analyses are provided in Golden and Stewart (1985), and in Johnson and McGeoch (1997, 2002). *Constructive heuristics* attempt to produce a good starting tour whereas *improvement heuristics* usually apply exchange operations to a tour in order to shorten it. Given that for small and medium instances the starting solution does not very much influence the final solution produced by an improvement heuristic, it is not worth spending much time constructing a first tour. Thus a *nearest neighbour* heuristic is very quick and adequate for most purposes and does not assume any structure in the cost matrix. For planar instances with uniformly distributed vertices, a *strip heuristic* (Daganzo, 1984) is simple to apply. The idea is to divide the plane into vertical strips of equal width w. Vertices are then visited in non-decreasing order of ordinates in odd-index strips and in non-increasing order of ordinates in even-index strips. This algorithm is very fast and convenient for drilling problems that must be solved in real time. Daganzo (1984) shows that if δ is the vertex density, then the optimal width strip is $w^* = \sqrt{3}\delta^{-1/2}$ for the Manhattan metric, and $w^* = \sqrt{2.95}\delta^{-1/2}$ for the Euclidean metric. Gaboune *et al* (1994) have shown that $w^* = 1.7389\delta^{1/2}$ for the Chebychev metric, which is frequently used in drilling operations. These authors have also derived results for three-dimensional metrics encountered in manufacturing.

Improvement heuristics are mostly based on edge exchange or chain exchange mechanisms. For example, the r-opt heuristic iteratively removes r edges from the current tour, considers all feasible reconnections and implements any improving move to yield the new current tour. The 2-opt heuristic and the 3-opt heuristics were first proposed by Flood (1956) and by Croes (1958), respectively. The more generalized concept of r-opt was put forward and analyzed by Lin (1965). Checking that a solution is r-opt optimal requires $O(n^r)$ operations, which explains why only 2-opt or 3-opt are normally used. However, using sophisticated data structures, Johnson and McGeoch (2002) have produced a highly efficient implementation of 3-opt that does not take much more time than 2-opt and yields significantly better results. Some heuristics such as Or-opt (Or, 1976) remove and relocate chains of vertices in an iterative fashion and can be viewed as restricted versions of 3-opt. Babin *et al* (2007) have shown that some hybrid versions of 2-opt and Or-opt outperform either of these two heuristics, both in solution quality and computing time.

The Lin and Kernighan (1973) heuristic has been shown to be one of the best available improvement methods for the symmetric TSP. It applies sets of 2-opt exchanges which, taken globally, produce a cost reduction. However, an efficient implementation of this algorithm requires a fair amount of sophistication regarding data structures and programming techniques, which makes it difficult to reproduce. The best-known implementation of the Lin and Kernighan heuristic is due to Helsgaun (2000). It consistently produces optimal solutions on small and medium size instances, in less than a second for $n = 100$ and in less than a minute for $n = 1000$ (on a 300 MHz G3 Power Macintosh). The largest instance solved optimally with this heuristic contains 85 900 vertices (Applegate *et al*, 2009).

On large instances that cannot be solved optimally, the Helsgaun heuristic typically produces quasi-optimal solution. Optimality gaps are measured with respect to the Held and Karp (1971) shortest spanning 1-tree lower bound. Thus the 1 904 711-city 'World TSP' was solved within 0.112% of the Held and Karp lower bound within 256.1 days of computing time, using parallel processors (Applegate *et al*, 2003). The Helsgaun heuristic is programmed in C and is available at http://www.akira.ruc.dk/~keld/research/LKH/.

4. Exact algorithms for the asymmetric TSP

The Dantzig *et al* (1954) formulation extends easily to the asymmetric case. Here x_{ij} is a binary variable, associated with arc (i, j) and equal to 1 if and only if the arc appears in the optimal tour. The formulation is as follows.

$$\text{(ATSP)} \quad \text{minimize} \sum_{i \neq j} c_{ij} x_{ij} \tag{6}$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad (i \in V,\ i \neq j) \tag{7}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad (j \in V,\ j \neq i) \tag{8}$$

$$\sum_{i,j \in S} x_{ij} \leqslant |S| - 1 \qquad (S \subset V, 2 \leqslant |S| \leqslant n-2) \tag{9}$$

$$x_{ij} = 0 \text{ or } 1 \qquad (i, j) \in A. \tag{10}$$

As in STSP, constraints (7) and (8), (9) and (10) are degree constraints, subtour elimination constraints and integrality constraints, respectively.

An interesting feature of ATSP is that relaxing the subtour elimination constraints yields a *Modified Assignment Problem* (MAP), which is an *Assignment Problem* (AP) in which assignments along the main diagonal are forbidden. The linear relaxation of this problem always has an integer solution and is easy to solve by means of a specialized AP algorithm, (see for example Carpaneto and Toth (1987), Dell'Amico and Toth (2000) and Burkard *et al* (2009)).

Many algorithms based on the AP relaxation have been devised. They differ in their lower bounding and branching rules. Some of the best known are those of Eastman (1958), Little *et al* (1963), Carpaneto and Toth (1980), Carpaneto *et al* (1995) and Fischetti and Toth (1992). Surveys of these algorithms and others have been presented in Balas and Toth

(1985), Laporte (1992) and Fischetti *et al* (2002). It is interesting to note that Eastman (1958) described what is probably the first ever branch-and-bound algorithm, 2 years before this method was suggested as a generic solution methodology for integer linear programming (Land and Doig, 1960), and 5 years before the term 'branch-and-bound' was coined by Little *et al* (1963).

The Carpaneto *et al* (1995) algorithm has the dual advantage of being fast and simple, and its Fortran implementation is publicly available (http://www.acm.org/calgo/contents/). The problem solved at a generic node of the branch-and-bound tree is a MAP and subproblems are created by branching on the arcs of subtours. Branching is performed on the subtour having the least number of arcs not already imposed in earlier branches. Before solving a MAP, a lower bound on its optimal solution is computed through a cost reduction procedure, which helps eliminating quickly some active nodes from consideration. At the root of the search tree, an upper bound on the optimal solution value is computed through the solution of a patching algorithm (see the next section). Various additional techniques are also employed, such as removing some arcs that cannot belong to an optimal solution, solving successive MAPs parametrically to avoid unnecessarily repeating some operations, and merging subtours if this yields an equivalent solution. The Fischetti and Toth (1992) algorithm improves slightly on that of Carpaneto *et al* (1995) by computing better lower bounds at the nodes of the search tree. These bounds combine three different relaxations of the TSP (the AP relaxation and two relaxations based on shortest spanning arborescences) within an additive bounding procedure.

The Carpanteo, Dell'Amico and Toth algorithm works rather well on randomly generated instances but it often fails on some rather small structured instances with as few as 100 vertices (Fischetti *et al*, 2002).

5. Heuristics for the asymmetric TSP

Johnson *et al* (2002), who have conducted a thorough computational study of the available heuristics for the ATSP, report that this problem appears to be a more difficult problem than its symmetric counterpart, both with respect to optimization and approximation. Again, the Helsgaun algorithm stands out in terms of time and solution quality. Since this heuristic was originally designed for the symmetric case, it is necessary to first transform the asymmetric TSP into an equivalent symmetric TSP to be able to apply this algorithm to asymmetric instances. Each vertex i is first duplicated into two vertices i^+ and i^- and the costs are redefined as $c_{i^+j^-} = c_{ij}, c_{j^+i^-} = c_{ij}, c_{i^-i^+} = -M$ and $c_{ij} = M'$ otherwise, where M and M' are large numbers such that all arcs with cost M are included in the tour and none of those with cost M' is included. Johnson *et al* (2002) have applied the Helsgaun heuristic to four asymmetric TSP instances containing 100, 316, 1000 and 3162 vertices, originating from two practical contexts and with different cost structures. Over these eight instances, the gap with respect to the Held and Karp lower bound ranged between 0.68% and 5.72%, and computing times ranged between 1.40 s (for $n = 100$) and 6244 s (for $n = 3162$). These are normalized times on a Compaq ES40 with 500-MHz Alpha processors and 2 Gbyte of main memory.

If one is interested in a very fast and easy-to-implement heuristic, the patching algorithm (Karp, 1979; Karp and Steele, 1985) is a very good contender. It first solves the assignment relaxation and subtours are gradually merged until only one tour exists. There are several ways of implementing this heuristic. Johnson *et al* (2002) recommend the following PATCH implementation: repeatedly select the two subtours containing the largest number of vertices and combine them in the best possible way into a single subtour by removing an arc from each and adding two arcs to make the reconnection. Johnson *et al* (2002) have applied PATCH to 32 instances containing the same 100, 316, 1000 and 3162 vertices as above, and in which the distance matrix was defined according to eight different rules. On the 100-vertex instances, the gaps with respect to the Held and Karp lower bound ranged between 0% and 24.41%, with a strong dependence on the metric used and almost no dependence on instance size. Average running times varied between 3.25 s (for $n = 100$) and 66.75 s (for $n = 3, 162$).

6. Conclusions

There exists an abundant and rich literature on the TSP. In this paper I have attempted to pinpoint the main bibliographic sources and some of the best exact and heuristic algorithms for the symmetric and asymmetric versions of the problem. All recommended algorithms are among the best in their class. They are all publicly available on the Web or are very easy to implement. I hope this concise guide will prove to be a valuable source of information for students, researchers and practitioners.

Acknowledgements— This research was partially supported by the Canadian Natural Sciences and Engineering Research Council under grant 39682-05. This support is gratefully acknowledged. Thanks are due to a referee who provided valuable comments.

References

Applegate DL, Bixby RE, Chvátal V and Cook WJ (2003). Implementing the Dantzig–Fulkerson–Johnson algorithm for large scale traveling salesman problems. *Math Program Ser B* **97**: 91–153.

Applegate DL, Bixby RE, Chvátal V and Cook WJ (2006). *The Traveling Salesman Problem: A Computational Study*. Pinceton Series in Applied Mathematics: Princeton.

Applegate DL, Bixby RE, Chvátal V, Cook WJ, Espinoza DG, Goycoolea M and Helsgaun K (2009). Certification of an optimal TSP tour through 85900 cities. *Opns Res Lett* **37**: 11–15.

Babin G, Deneault S and Laporte G (2007). Improvements to the Or-opt heuristic for the symmetric travelling salesman problem. *J Opl Res Soc* **58**: 402–407.

Balas E and Toth P (1985). Branch and bound methods. In: Lawler EL, Lenstra JK, Rinnooy Kan AHG and Shmoys DB (eds). *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley: Chichester, pp 361–401.

Burkard R, Dell'Amico M and Martello S (2009). *Assignment Problems*. SIAM: Philadelphia.

Carpaneto G and Toth P (1980). Some new branching and bounding criteria for the asymmetric traveling salesman problem. *Mngt Sci* **26**: 736–743.

Carpaneto G and Toth P (1987). Primal-dual algorithms for the assignment problem. *Discrete Appl Math* **18**: 137–153.

Carpaneto G, Dell'Amico M and Toth P (1995). Exact solution of large-scale, asymmetric travelling salesman problems. *ACM Trans Math Software* **21**: 394–409.

Chvátal V (1973). Edmonds polytopes and weakly Hamiltonian graphs. *Math Program* **5**: 29–40.

Cornuéjols G, Fonlupt J and Naddef D (1985). The travelling salesman problem on a graph and some related integer polyhedra. *Math Program* **33**: 1–27.

Croes GA (1958). A method for solving large scale symmetric traveling salesman problems. *Opns Res* **6**: 791–812.

Crowder H and Padberg MW (1980). Solving large-scale symmetric traveling salesman problems to optimality. *Mngt Sci* **26**: 495–509.

Daganzo CF (1984). The length of tours in zones of different shapes. *Transport Res B* **18**: 135–145.

Dantzig GB, Fulkerson DR and Johnson SM (1954). Solution of a large-scale traveling-salesman problem. *Opns Res* **2**: 393–410.

Dell'Amico M and Toth P (2000). Algorithms and codes for dense assignment problems: The state of the art. *Discrete Appl Math* **100**: 17–48.

Eastman WL (1958). *Linear programming with pattern constraints*. PhD thesis, Department of Economics, Harvard University, Cambridge, MA.

Edmonds J (1965). Maximum matching and a polyhedron with 0, 1-vertices. *J Res Natl Bur Stan* **69B**: 125–130.

Fischetti M and Toth P (1992). An additive bounding procedure for the asymmetric traveling salesman problem. *Math Program Ser A* **53**: 173–197.

Fischetti M, Lodi A and Toth P (2002). Exact methods for the asymmetric traveling salesman problem. In: Gutin G and Punnen AP (eds). *The Traveling Salesman Problem and Its Variations*. Kluwer: Boston. pp 169–205.

Flood MM (1956). The traveling-salesman problem. *Opns Res* **4**: 61–75.

Gaboune B, Laporte G and Soumis F (1994). Optimal strip sequencing strategies for flexible manufacturing operations in two and three dimensions. *Int J Flex Manuf Sys* **6**: 123–135.

Golden BL and Stewart WR (1985). Empirical analysis of heuristics. In: Lawler EL, Lenstra JK, Rinnooy Kan AHG and Shmoys DB (eds). *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley: Chichester, pp 207–249.

Gomory RE (1963). An algorithm for integer solutions to linear programs. In: Graves RL and Wolfe P (eds). *Recent Advances in Mathematical Programming*. McGraw-Hill: New York, pp 269–302.

Grötschel M and Holland O (1991). Solution of large-scale symmetric traveling salesman problems. *Math Program* **51**: 141–202.

Grötschel M and Padberg MW (1985). Polyhedral theory. In: Lawler EL, Lenstra JK, Rinnooy Kan AHG and Shmoys DB (eds). *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley: Chichester, pp 251–305.

Grötschel M and Pulleyblank WR (1986). Clique tree inequalities and the symmetric Travelling Salesman Problem. *Math Opns Res* **11**: 537–569.

Gutin G and Punnen AP (eds) (2002). *The Traveling Salesman Problem and Its Variations*. Kluwer: Boston.

Held M and Karp RM (1971). The traveling-salesman problem and minimum spanning trees: Part II. *Math Program* **1**: 6–25.

Helsgaun K (2000). An effective implementation of the Lin–Kernighan traveling salesman heuristic. *Eur J Opl Res* **126**: 106–130.

Johnson DS, Gutin G, McGeoch LA, Yeo A, Zhang W and Zverovitch A (2002). Experimental analysis of heuristics for the ATSP. In: Gutin G and Punnen AP (eds). *The Traveling Salesman Problem and Its Variations*. Kluwer: Boston, pp 445–487.

Johnson DS and McGeoch LA (1997). The traveling salesman problem: A case study. In: Aarts E and Lenstra JK (eds). *Local Search in Combinatorial Optimization*. Wiley: Chichester, pp 215–310.

Johnson DS and McGeoch LA (2002). Experimental analysis of heuristics for the STPS. In: Gutin G and Punnen AP (eds). *The Traveling Salesman Problem and Its Variations*. Kluwer: Boston. pp 369–443.

Jünger M, Reinelt G and Rinaldi G (1995). The traveling salesman problem. In: Ball MO, Magnanti TL, Monma CL and Nemhauser GL (eds). Network Models Handbooks in Operations Research and Management Science, vol. 7. North-Holland: Amsterdam, pp 225–330.

Karp RM (1979). A patching algorithm for the nonsymmetric traveling-salesman problem. *SIAM J Comput* **8**: 561–573.

Karp RM and Steele JM (1985). Probabilistic analysis of heuristics. In: Lawler EL, Lenstra JK, Rinnooy Kan AHG and Shmoys DB (eds). *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley: Chichester, pp 181–205.

Land AH (1979). *The solution of some 100-city traveling salesman problems*. Technical report, London School of Economics, London.

Land AH and Doig AG (1960). An automatic method of solving discrete programming problems. *Econometrica* **28**: 497–520.

Laporte G (1992). The traveling salesman problem: An overview of exact and approximate algorithms. *Eur J Opl Res* **59**: 231–247.

Lawler EL, Lenstra JK, Rinnooy Kan AHG and Shmoys DB (eds) (1985). *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley: Chichester.

Lin S (1965). Computer solutions of the traveling salesman problem. *Bell Syst Tech J* **44**: 2245–2269.

Lin S and Kernighan BW (1973). An effective heuristic algorithm for the traveling salesman problem. *Opns Res* **21**: 498–516.

Little JDC, Murty KG, Sweeney DW and Karel C (1963). An algorithm for the traveling salesman problem. *Opns Res* **11**: 972–989.

Martin GT (1966). *Solving the traveling salesman problem by integer programming*. Working Paper, CEIR, New York.

Miliotis P (1976). Integer programming approaches to the travelling salesman problem. *Math Program* **10**: 376–378.

Miliotis P (1978). Using cutting planes to solve the symmetric travelling salesman problem. *Math Program* **15**: 177–188.

Naddef D (2002). Polyhedral theory and branch-and-cut algorithms for the symmetric TSP. In: Gutin G and Punnen AP (eds). *The Traveling Salesman Problem and Its Variations*. Kluwer: Boston, pp 29–116.

Öncan T, Altınel IK and Laporte G (2009). A comparative analysis of several asymmetric traveling salesman problem formulations. *Comput Opns Res* **36**: 637–654.

Or I (1976). *Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking*. PhD thesis, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Il.

Orman AJ and Williams HP (2006). A survey of different integer programming formulations of the travelling salesman problem. In: Kontoghiorghes E and Gatu C (eds). *Optimisation, Econometric and Financial Analysis Advances in Computational Management Science*. Springer: Berlin, Heidelberg, pp 91–104.

Padberg MW and Grötschel M (1985). Polyhedral computations. In: Lawler EL, Lenstra JK, Rinnooy Kan AHG and Shmoys DB (eds). *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley: Chichester, pp 307–360.

Padberg MW and Hong S (1980). On the symmetric travelling salesman problem: A computational study. *Math Program Study* **12**: 78–107.

Padberg MW and Rinaldi G (1987). Optimization of a 532-city symmetric traveling salesman problem by branch and cut. *Opns Res Lett* **6**: 1–7.

Padberg MW and Rinaldi G (1991). A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Rev* **33**: 60–100.

Toth P and Vigo D (eds) (2002). The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia.

Received April 2009;
accepted May 2009 after one revision