# 1

# Traveling Salesman Problem: An Overview of Applications, Formulations, and Solution Approaches

Rajesh Matai[1], Surya Prakash Singh[2] and Murari Lal Mittal[3]
[1]*Management Group, BITS-Pilani*
[2]*Department of Management Studies, Indian Institute of Technology Delhi, New Delhi*
[3]*Department of Mechanical Engineering, Malviya National Institute of Technology Jaipur, India*

## 1. Introduction

### 1.1 Origin

The traveling salesman problem (TSP) were studied in the 18th century by a mathematician from Ireland named Sir William Rowam Hamilton and by the British mathematician named Thomas Penyngton Kirkman. Detailed discussion about the work of Hamilton & Kirkman can be seen from the book titled Graph Theory (Biggs et al. 1976). It is believed that the general form of the TSP have been first studied by Kalr Menger in Vienna and Harvard. The problem was later promoted by Hassler, Whitney & Merrill at Princeton. A detailed dscription about the connection between Menger & Whitney, and the development of the TSP can be found in (Schrijver, 1960).

### 1.2 Definition

Given a set of cities and the cost of travel (or distance) between each possible pairs, the TSP, is to find the best possible way of visiting all the cities and returning to the starting point that minimize the travel cost (or travel distance).

### 1.3 Complexity

Given *n* is the number of cities to be visited, the total number of possible routes covering all cities can be given as a set of feasible solutions of the TSP and is given as ($n$-1)!/2.

### 1.4 Classification

Broadly, the TSP is classified as symmetric travelling salesman problem (sTSP), asymmetric travelling salesman problem (aTSP), and multi travelling salesman problem (mTSP). This section presents description about these three widely studied TSP.

**sTSP**: Let $V=\{v_1,......,v_n\}$ be a set of cities, $A=\{(r,s):r,s\in V\}$ be the edge set, and $d_{rs}=d_{sr}$ be a cost measure associated with edge $(r,s)\in A$.

The sTSP is the problem of finding a minimal length closed tour that visits each city once. In this case cities $v_i \in V$ are given by their coordinates $(x_i, y_i)$ and $d_{rs}$ is the Euclidean distance between $r$ and $s$ then we have an Euclidean TSP.

**aTSP:** If $d_{rs} \neq d_{sr}$ for at least one $(r,s)$ then the TSP becomes an aTSP.
**mTSP:** The mTSP is defined as: In a given set of nodes, let there are m salesmen located at a single depot node. The remaining nodes (cities) that are to be visited are intermediate nodes. Then, the mTSP consists of finding tours for all m salesmen, who all start and end at the depot, such that each intermediate node is visited exactly once and the total cost of visiting all nodes is minimized. The cost metric can be defined in terms of distance, time, etc. Possible variations of the problem are as follows: *Single vs. multiple depots*: In the single depot, all salesmen finish their tours at a single point while in multiple depots the salesmen can either return to their initial depot or can return to any depot keeping the initial number of salesmen at each depot remains the same after the travel. *Number of salesmen*: The number of salesman in the problem can be fixed or a bounded variable. *Cost*: When the number of salesmen is not fixed, then each salesman usually has an associated fixed cost incurring whenever this salesman is used. In this case, the minimizing the requirements of salesman also becomes an objective. *Timeframe*: Here, some nodes need to be visited in a particular time periods that are called time windows which is an extension of the mTSP, and referred as multiple traveling salesman problem with specified timeframe (mTSPTW). The application of mTSPTW can be very well seen in the aircraft scheduling problems. *Other constraints*: Constraints can be on the number of nodes each salesman can visits, maximum or minimum distance a salesman travels or any other constraints. The mTSP is generally treated as a relaxed vehicle routing problems (VRP) where there is no restrictions on capacity. Hence, the formulations and solution methods for the VRP are also equally valid and true for the mTSP if a large capacity is assigned to the salesmen (or vehicles). However, when there is a single salesman, then the mTSP reduces to the TSP (Bektas, 2006).

## 2. Applications and linkages

### 2.1 Application of TSP and linkages with other problems

**i. Drilling of printed circuit boards**

A direct application of the TSP is in the drilling problem of printed circuit boards (PCBs) (Grötschel et al., 1991). To connect a conductor on one layer with a conductor on another layer, or to position the pins of integrated circuits, holes have to be drilled through the board. The holes may be of different sizes. To drill two holes of different diameters consecutively, the head of the machine has to move to a tool box and change the drilling equipment. This is quite time consuming. Thus it is clear that one has to choose some diameter, drill all holes of the same diameter, change the drill, drill the holes of the next diameter, etc. Thus, this drilling problem can be viewed as a series of TSPs, one for each hole diameter, where the 'cities' are the initial position and the set of all holes that can be drilled with one and the same drill. The 'distance' between two cities is given by the time it takes to move the drilling head from one position to the other. The aim is to minimize the travel time for the machine head.

**ii. Overhauling gas turbine engines**

(Plante et al., 1987) reported this application and it occurs when gas turbine engines of aircraft have to be overhauled. To guarantee a uniform gas flow through the turbines there are nozzle-guide vane assemblies located at each turbine stage. Such an assembly basically consists of a number of nozzle guide vanes affixed about its circumference. All these vanes have individual characteristics and the correct placement of the vanes can result in substantial benefits (reducing vibration, increasing uniformity of flow, reducing fuel consumption). The problem of placing the vanes in the best possible way can be modeled as a TSP with a special objective function.

**iii. X-Ray crystallography**

Analysis of the structure of crystals (Bland & Shallcross, 1989; Dreissig & Uebach, 1990) is an important application of the TSP. Here an X-ray diffractometer is used to obtain information about the structure of crystalline material. To this end a detector measures the intensity of X-ray reflections of the crystal from various positions. Whereas the measurement itself can be accomplished quite fast, there is a considerable overhead in positioning time since up to hundreds of thousands positions have to be realized for some experiments. In the two examples that we refer to, the positioning involves moving four motors. The time needed to move from one position to the other can be computed very accurately. The result of the experiment does not depend on the sequence in which the measurements at the various positions are taken. However, the total time needed for the experiment depends on the sequence. Therefore, the problem consists of finding a sequence that minimizes the total positioning time. This leads to a traveling salesman problem.

**iv. Computer wiring**

(Lenstra & Rinnooy Kan, 1974) reported a special case of connecting components on a computer board. Modules are located on a computer board and a given subset of pins has to be connected. In contrast to the usual case where a Steiner tree connection is desired, here the requirement is that no more than two wires are attached to each pin. Hence we have the problem of finding a shortest Hamiltonian path with unspecified starting and terminating points. A similar situation occurs for the so-called testbus wiring. To test the manufactured board one has to realize a connection which enters the board at some specified point, runs through all the modules, and terminates at some specified point. For each module we also have a specified entering and leaving point for this test wiring. This problem also amounts to solving a Hamiltonian path problem with the difference that the distances are not symmetric and that starting and terminating point are specified.

**v. The order-picking problem in warehouses**

This problem is associated with material handling in a warehouse (Ratliff & Rosenthal, 1983). Assume that at a warehouse an order arrives for a certain subset of the items stored in the warehouse. Some vehicle has to collect all items of this order to ship them to the customer. The relation to the TSP is immediately seen. The storage locations of the items correspond to the nodes of the graph. The distance between two nodes is given by the time needed to move the vehicle from one location to the other. The problem of finding a shortest route for the vehicle with minimum pickup time can now be solved as a TSP. In special cases this problem can be solved easily, see (van Dal, 1992) for an extensive discussion and for references.

**vi. Vehicle routing**

Suppose that in a city $n$ mail boxes have to be emptied every day within a certain period of time, say 1 hour. The problem is to find the minimum number of trucks to do this and the shortest time to do the collections using this number of trucks. As another example, suppose that n customers require certain amounts of some commodities and a supplier has to satisfy all demands with a fleet of trucks. The problem is to find an assignment of customers to the trucks and a delivery schedule for each truck so that the capacity of each truck is not exceeded and the total travel distance is minimized. Several variations of these two problems, where time and capacity constraints are combined, are common in many real-world applications. This problem is solvable as a TSP if there are no time and capacity constraints and if the number of trucks is fixed (say $m$). In this case we obtain an $m$-salesmen problem. Nevertheless, one may apply methods for the TSP to find good feasible solutions for this problem (see Lenstra & Rinnooy Kan, 1974).

**vii. Mask plotting in PCB production**

For the production of each layer of a printed circuit board, as well as for layers of integrated semiconductor devices, a photographic mask has to be produced. In our case for printed circuit boards this is done by a mechanical plotting device. The plotter moves a lens over a photosensitive coated glass plate. The shutter may be opened or closed to expose specific parts of the plate. There are different apertures available to be able to generate different structures on the board. Two types of structures have to be considered. A line is exposed on the plate by moving the closed shutter to one endpoint of the line, then opening the shutter and moving it to the other endpoint of the line. Then the shutter is closed. A point type structure is generated by moving (with the appropriate aperture) to the position of that point then opening the shutter just to make a short flash, and then closing it again. Exact modeling of the plotter control problem leads to a problem more complicated than the TSP and also more complicated than the rural postman problem. A real-world application in the actual production environment is reported in (Grötschel et al., 1991).

### 2.2 Applications of mTSP and connections with other problems

This section is further divided into three. In the first section, the main application of the mTSP is given. The second part relates TSP with other problems. The third part deals with the similarities between the mTSP with other problems (the focus is with the VRP).

#### 2.2.1 Main applications

The main apllication of mTSP arises in real scenario as it is capacble to handle multiple salesman. These situations arise mostly in various routing and scheduling problems. Some reported applications in literature are presented below.

i. **Printing press scheduling problem:** One of the major and primary applications of the mTSP arises in scheduling a printing press for a periodical with multi-editions. Here, there exist five pairs of cylinders between which the paper rolls and both sides of a page are printed simultaneously. There exist three kind of forms, namely 4-, 6- and 8-page forms, which are used to print the editions. The scheduling problem consists of deciding which form will be on which run and the length of each run. In the mTSP vocabulary, the plate change costs are the inter-city costs. For more details papers by Gorenstein (1970) and Carter & Ragsdale (2002) can be referred.
ii. **School bus routing problem:** (Angel et al., 1972) investigated the problem of scheduling buses as a variation of the mTSP with some side constraints. The objective of the scheduling is to obtain a bus loading pattern such that the number of routes is minimized, the total distance travelled by all buses is kept at minimum, no bus is overloaded and the time required to traverse any route does not exceed a maximum allowed policy.
iii. **Crew scheduling problem:** An application for deposit carrying between different branch banks is reported by (Svestka & Huckfeldt, 1973). Here, deposits need to be picked up at branch banks and returned to the central office by a crew of messengers. The problem is to determine the routes having a total minimum cost. Two similar applications are described by (Lenstra & Rinnooy Kan , 1975 and Zhang et al., 1999). Papers can be referred for delaited analysis.

iv. **Interview scheduling problem:** (Gilbert & Hofstra, 1992) found the application of mTSP, having multiperiod variations, in scheduling interviews between tour brokers and vendors of the tourism industry. Each broker corresponds to a salesman who must visit a specified set of vendor booths, which are represented by a set of T cities.

v. **Hot rolling scheduling problem:** In the iron and steel industry, orders are scheduled on the hot rolling mill in such a way that the total set-up cost during the production can be minimized. The details of a recent application of modeling such problem can be read from (Tang et al., 2000). Here, the orders are treated as cities and the distance between two cities is taken as penalty cost for production changeover between two orders. The solution of the model will yield a complete schedule for the hot strip rolling mill.

vi. **Mission planning problem:** The mission planning problem consists of determining an optimal path for each army men (or planner) to accomplish the goals of the mission in the minimum possible time. The mission planner uses a variation of the mTSP where there are n planners, m goals which must be visited by some planners, and a base city to which all planners must eventually return. The application of the mTSP in mission planning is reported by (Brummit & Stentz, 1996; Brummit & Stentz, 1998; and Yu et al., 2002). Similarly, the routing problems arising in the planning of unmanned aerial vehicle applications, investigated by (Ryan et al., 1998), can also be modelled as mTSP.

vii. **Design of global navigation satellite system surveying networks:** A very recent and an interesting application of the mTSP, as investigated by (Saleh & Chelouah, 2004) arises in the design of global navigation satellite system (GNSS) surveying networks. A GNSS is a space-based satellite system which provides coverage for all locations worldwide and is quite crucial in real-life applications such as early warning and management for disasters, environment and agriculture monitoring, etc. The goal of surveying is to determine the geographical positions of unknown points on and above the earth using satellite equipment. These points, on which receivers are placed, are co-ordinated by a series of observation sessions. When there are multiple receivers or multiple working periods, the problem of finding the best order of sessions for the receivers can be formulated as an mTSP. For technical details refer (Saleh & Chelouah, 2004).

### 2.2.2 Connections with other problems

The above-mentioned problems can be modeled as an mTSP. Apart from these above metioned problmes, mTSP can be also related to other problems. One such example is balancing the workload among the salesmen and is described by (Okonjo-Adigwe, 1988). Here, an mTSP-based modelling and solution approach is presented to solve a workload scheduling problem with few additional restrictions. Paper can be referred for detailed description and analysis. Similalry, (Calvo & Cordone, 2003; Kim & Park, 2004) investigated overnight security service problem. This problem consists of assigning duties to guards to perform inspection duties on a given set of locations with subject to constraint such as capacity and timeframe. For more comprehensive review on various application of mTSP authors advise to refer papers by (Macharis & Bontekoning, 2004; Wang & Regan, 2002; Basu et al., 2000).

### 2.2.3 Connections with the VRP

mTSP can be utilized in solving several types of VRPs. (Mole et al., 1983) discuss several algorithms for VRP, and present a heuristic method which searches over a solution space formed by the mTSP. In a similar context, the mTSP can be used to calculate the minimum number of vehicles required to serve a set of customers in a distance-constrained VRP (Laptore et al., 1985; Toth & Vigo, 2002). The mTSP also appears to be a first stage problem in a two-stage solution procedure of a VRP with probabilistic service times. This is discussed further by (Hadjiconstantinou & Roberts, 2002). (Ralphs, 2003) mentions that the VRP instances arising in practice are very hard to solve, since the mTSP is also very complex. This raises the need to efficiently solve the mTSP in order to attack large-scale VRPs. The mTSP is also related to the pickup and delivery problem (PDP). The PDP consists of determining the optimal routes for a set of vehicles to fulfill the customer requests (Ruland & Rodin, 1997). If the customers are to be served within specific time intervals, then the problem becomes the PDP with time windows (PDPTW). The PDPTW reduces to the mTSPTW if the origin and destination points of each request coincide (Mitrović-Minić et al., 2004).

## 3. Mathematical formulations of TSP and mTSP

The TSP can be 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, . . . , 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 $c_{ij}\le c_{ik}+c_{kj}$, 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 $G$.

### 3.1 Integer programming formulation of sTSP

Many TSP formulations are available in literature. Recent surveys by (Orman & Williams, 2006; O¨ncan et al., 2009) can be referred for detailed analysis. Among these, the (Dantzig et al., 1954) formulation is one of the most cited mathematical formulation for TSP. 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 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 of TSP is as follows.
Minimize

$$\sum_{i<j} c_{ij}x_{ij} \tag{1}$$

Subject to

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

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

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

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

$$\sum_{i \in S, j \in V \setminus S, j \in S} x_{ij} \ge 2 \qquad (S \subset V, 3 \le |S| \le n-3) \tag{5}$$

### 3.2 Integer programming formulation of aTSP

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.
Minimize

$$\sum_{i \ne j} c_{ij} x_{ij} \tag{6}$$

Subject to

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

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

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

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

### 3.3 Integer programming formulations of mTSP

Different types of integer programming formulations are proposed for the mTSP. Before presenting them, some technical definitions are as follows. The mTSP is defined on a graph $G = (V, A)$, where V is the set of n nodes (vertices) and A is the set of arcs (edges). Let $C = (c_{ij})$ be a cost (distance) matrix associated with A. The matrix C is said to be symmetric when $c_{ij} = c_{ji}, \forall (i,j) \in A$ and asymmetric otherwise. If $c_{ij} + c_{jk} \ge c_{ik}, \forall i,j,k \in V$, C is said to satisfy the triangle inequality. Various integer programming formulations for the mTSP have been proposed earlier in the literature, among which there exist assignment-based formulations, a tree-based formulation and a three-index flow-based formulation. Assignment based formulations are presented in following subsections. For tree based formulation and three-index based formulations refer (Christofides et al., 1981).

### 3.3.1 Assignment-based integer programming formulations

The mTSP is usually formulated using an assignment based double-index integer linear programming formulation. We first define the following binary variable:

$$x_{ij} = \begin{cases} 1 & \text{If arc (i, j) is used in the tour,} \\ 0 & \end{cases}$$

Otherwise.

Then, a general scheme of the assignment-based directed integer linear programming formulation of the mTSP can be given as follows:

Minimize

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$$

Subject to

$$\sum_{j=2}^{n} x_{1j} = m \tag{11}$$

$$\sum_{j=2}^{n} x_{j1} = m \tag{12}$$

$$\sum_{i=1}^{n} x_{ij} = 1\,, \quad j = 2,.......,n \tag{13}$$

$$\sum_{j=1}^{n} x_{ij} = 1\,,\ i = 2,.......,n \tag{14}$$

$$+ \text{ subtour elimination constraints,} \tag{15}$$

$$x_{ij} \in \{0,1\}\,,\ \forall (i,j) \in A\,, \tag{16}$$

where (13), (14) and (16) are the usual assignment constraints, (11) and (12) ensure that exactly m salesmen depart from and return back to node 1 (the depot). Although constraints (12) are already implied by (11), (13) and (14), we present them here for the sake of completeness. Constraints (15) are used to prevent subtours, which are degenerate tours that are formed between intermediate nodes and not connected to the origin. These constraints are named as subtour elimination constraints (SECs). Several SECs have been proposed for the mTSP in the literature. The first group of SECs is based on that of (Dantzig et al., 1954) originally proposed for the TSP, but also valid for the mTSP. These constraints can be shown as follows:

$$\sum_{i\in S}\sum_{j\in S} x_{ij} \le |S| - 1\,, \qquad \forall S \subseteq V \setminus \{1\}\,, \qquad S \ne \varnothing \tag{17}$$

or alternatively in the following form

$$\sum_{i \notin S} \sum_{j \in S} x_{ij} \geq 1, \qquad \forall S \subseteq V \setminus \{1\}, \qquad S \neq \varnothing \tag{18}$$

Constraints (17) or (18) impose connectivity requirements for the solution, i.e. prevent the formation of subtours of cardinality S not including the depot. Unfortunately, both families of these constraints increase exponentially with increasing number of nodes, hence are not practical for neither solving the problem nor its linear programming relaxation directly. Miller et al. (1960) overcame this problem by introducing $O(n^2)$ additional continuous variables, namely node potentials, resulting in a polynomial number of SECs. Their SECs are given as follows (denoted by MTZ-SECs):

$$u_i - u_j + p x_{ij} \leq p - 1 \text{ for } 2 \leq i \neq j \leq n \tag{19}$$

Here, p denotes the maximum number of nodes that can be visited by any salesman. The node potential of each node indicates the order of the corresponding node in the tour. (Svestka & Huckfeldt, 1973) propose another group of SECs for the mTSP which require augmenting the original cost matrix with new rows and columns. However, (Gavish, 1976) showed that their constraints are not correct for m≥2 and provided the correct constraints as follows:

$$u_i - u_j + (n - m) x_{ij} \leq n - m - 1 \text{ for } 2 \leq i \neq j \leq n \tag{20}$$

Other MTZ-based SECs for the mTSP have also been proposed. The following constraints are due to Kulkarni & Bhave (1985) (denoted by KB-SECs):

$$u_i - u_j + L x_{ij} \leq L - 1 \text{ for } 2 \leq i \neq j \leq n \tag{21}$$

In these constraints, the L is same as p in (19). It is clear that MTZ-SECs and KB-SECs are equivalent.

### 3.3.2 Laporte & Nobert's formulations

(Laporte & Nobert, 1980) presented two formulations for the mTSP, for asymmetrical and symmetrical cost structures, respectively, and consider a common fixed cost f for each salesman used in the solution. These formulations are based on the two-index variable $x_{ij}$ defined previously.

3.3.2.1 Laporte & Nobert's formulation for the asymmetric mTSP

Minimize

$$\sum_{i \neq j} c_{ij} x_{ij} + f_m$$

Subject to

$$\sum_{j=2}^{n} \left( x_{1j} + x_{j1} \right) = 2m \tag{22}$$

$$\sum_{i \neq k} x_{ik} = 1 \qquad k = 2, \ldots\ldots, n \tag{23}$$

$$\sum_{j \neq k} x_{ik} = 1 \qquad k = 2, \ldots\ldots, n \tag{24}$$

$$\sum_{i \neq j; i, j \in S} x_{ij} \leq |S - 1|$$

$$2 \leq |S| \leq n - 2, \qquad S \subseteq V \setminus \{1\} \tag{25}$$

$$x_{ij} \in \{0,1\}, \qquad \forall i \neq j \tag{26}$$

$$m \geq 1 \text{ and integer} \tag{27}$$

This formulation is a pure binary integer where the objective is to minimize the total cost of the travel as well as the total number of salesmen. Note that constraints (23) and (24) are the standard assignment constraints, and constraints (25) are the SECs of (Dantzig et al., 1954). The only different constraints are (22), which impose degree constraints on the depot node.

3.3.2.2 Laporte & Nobert's formulation for the symmetric mTSP

Minimize

$$\sum_{i < j} c_{ij} x_{ij} + f_m$$

Subject to

$$\sum_{j=2}^{n} x_{1j} = 2m \tag{28}$$

$$\sum_{i < k} x_{ik} + \sum_{j > k} x_{kj} = 2 \qquad k = 2, \ldots\ldots, n \tag{29}$$

$$\sum_{i < j; i, j \in S} x_{ij} \leq |S - 1|$$

$$3 \leq |S| \leq n - 2, \qquad S \subseteq V \setminus \{1\} \tag{30}$$

$$x_{ij} \in \{0,1\}, \qquad 1 < i < j \tag{31}$$

$$x_{1j} \in \{0,1,2\}, \qquad j = 2, \ldots\ldots, n \tag{32}$$

$$m \geq 1 \text{ and integer} \qquad (32)$$

The interesting issue about this formulation is that it is not a pure binary integer formulation due to the variable $x_{1j}$, which can either be 0, 1 or 2. Note here that the variable $x_{1j}$ is only defined for i <j, since the problem is symmetric and only a single variable is sufficient to represent each edge used in the solution. Constraints (28) and (29) are the degree constraints

on the depot node and intermediate nodes, respectively. Other constraints are as previously defined.

## 4. Exact solution approaches

### 4.1 Exact algorithms for the sTSP

When (Dantzig et al., 1954) formulation was first introduced, the simplex method was in its infancy and no algorithms were available to solve integer linear programs. The practitioners 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. (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). (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. 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 (Chv´atal, 1973). Some generalizations of comb inequalities, such as clique tree inequalities (Grötschel & Pulleyblank, 1986) and path inequalities (Cornu´ejols 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 & Padberg, 1980; Grötschel & Padberg, 1985; Padberg & Grötschel, 1985; Padberg & Rinaldi, 1987, 1991; Grötschel & 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). 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 ($n \leq 1000$) were solved within 1 min on a 2.4 GHz ADM Opteron processor. On 21 medium-size TSPLIB instances ($1000 \leq n \leq 2392$), 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 85900 vertices (Applegate et al., 2009).

| N | Type | Sample size | Mean CPU seconds |
|---|---|---|---|
| 100 | random | 10000 | 0.7 |
| 500 | random | 10000 | 50.2 |
| 1000 | random | 1000 | 601.6 |
| 2000 | random | 1000 | 14065.6 |
| 2500 | random | 1000 | 53737.9 |

Table 1. Computation times for Concorde

### 4.2 Exact algorithms for the aTSP

An interesting feature of aTSP is that relaxing the subtour elimination constraints yields a Modified Assignment Problem (MAP), which is an assignment problem. The linear relaxation of this problem always has an integer solution and is easy to solve by means of a specialized assignment algorithm, (Carpaneto & Toth, 1987; Dell'Amico & Toth, 2000 and Burkard et al., 2009). Many algorithms based on the AP relaxation have been devised. Some of the best known are those of (Eastman,1958; Little et al., 1963; Carpaneto & Toth, 1980; Carpaneto et al., 1995 and Fischetti & Toth, 1992). Surveys of these algorithms and others have been presented in (Balas & 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 & 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. The (Fischetti & Toth, 1992) algorithm improves slightly on that of (Carpaneto et al., 1995) by computing better lower bounds at the nodes of the search tree. The Carpanteo, Dell'Amico & 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). A branch- and bound based algorithm for the asymmetric TSP is proposed by (Ali & Kennington, 1986). The algorithm uses a Lagrangean relaxation of the degree constraints and a subgradient algorithm to solve the Lagrangean dual.

### 4.3 Exact algorithms for mTSP

The first approach to solve the mTSP directly, without any transformation to the TSP is due to (Laporte & Nobert, 1980), who propose an algorithm based on the relaxation of some constraints of the mTSP. The problem they consider is an mTSP with a fixed cost f associated with each salesman. The algorithm consists of solving the problem by initially relaxing the SECs and performing a check as to whether any of the SECs are violated, after an integer solution is obtained. The first attempt to solve large-scale symmetric mTSPs to optimality is due to (Gavish & Srikanth, 1986). The proposed algorithm is a branch-and-bound method, where lower bounds are obtained from the following Lagrangean problem constructed by relaxing the degree constraints. The Lagrangean problem is solved using a degree-constrained minimal spanning tree which spans over all the nodes. The results indicate that the integer gap obtained by the Lagrangean relaxation decreases as the problem size increases and turns out to be zero for all problems with $n \geq 400$. (Gromicho et al., 1992) proposed another exact solution method for mTSP. The algorithm is based on a quasi-assignment (QA) relaxation obtained by relaxing the SECs, since the QA-problem is solvable in polynomial time. An additive bounding procedure is applied to strengthen the lower bounds obtained via different r-arborescence and r-anti-arborescence relaxations and this procedure is embedded in a branch-and-bound framework. It is observed that the additive bounding procedure has a significant effect in improving the lower bounds, for which the QA-relaxation yields poor bounds. The proposed branch-and-bound algorithm is superior to the standard branch-and-bound approach with a QA-relaxation in terms of number of nodes, ranging from 10% less to 10 times less. Symmetric instances are observed to yield larger improvements. Using an IBM PS/70 computer with an 80386 CPU running at 25 MHz, the biggest instance solved via this approach has 120 nodes with the number of salesman ranging from 2 to 12 in steps of one (Gromicho, 2003).

## 5. Approximate approaches

There are mainly two ways of solving any TSP instance optimally. The first is to apply an exact approach such as Branch and Bound method to find the length. The other is to calculate the Held-Karp lower bound, which produces a lower bound to the optimal solution. This lower bound is used to judge the performance of any new heuristic proposed for the TSP. The heuristics reviewed here mainly concern with the sTSP, however some of these heuristics can be modified appropriatley to solve the aTSP.

### 5.1 Approximation

Solving even moderate size of the TSP optimally takes huge computtaional time, therefore there is a room for the development and application of approximate algorithms, or heuristics. The approximate approach never guarantee an optimal solution but gives near optimal solution in a reasonable computational effort. So far, the best known approximate algorithm available is due to (Arora, 1998). The complexity of the approximate algorithm is $O\left(n\left(\log_2 n\right)^{O(c)}\right)$ where n is problem size of TSP.

### 5.2 Tour construction approaches

All tour construction algorithms stops when a solution is found and never tries to improve it. It is believed that tour construction algorithms find solution within 10-15% of optimality. Few of the tour construction algorithms available in published literature are described below.

#### 5.2.1 Closest neighbor heuristic

This is the simplest and the most straightforward TSP heuristic. The key to this approach is to always visit the closest city. The polynomial complexity associated with this heuristic approach is $O\left(n^2\right)$. The closest approach is very similar to minimum spanning tree algorithm. The steps of the closest neighbor are given as:

1. Select a random city.
2. Find the nearest unvisited city and go there.
3. Are there any unvisitied cities left? If yes, repeat step 2.
4. Return to the first city.

The Closest Neighbor heuristic approach generally keeps its tour within 25% of the Held-Karp lower bound (Johnson & McGeoch, 1995).

### 5.2.2 Greedy heuristic

The Greedy heuristic gradually constructs a tour by repeatedly selecting the shortest edge and adding it to the tour as long as it doesn't create a cycle with less than N edges, or increases the degree of any node to more than 2. We must not add the same edge twice of course. Complexity of the greedy heuristic is $O\left(n^2 \log_2(n)\right)$. Steps of Greedy approach are:

1. Sort all edges.
2. Select the shortest edge and add it to our tour if it doesn't violate any of the above constraints.
3. Do we have N edges in our tour? If no, repeat step 2.

The Greedy algorithm normally keeps solution within 15- 20% of the Held-Karp lower bound (Johnson & McGeoch, 1995).

### 5.2.3 Insertion heuristic

Insertion heuristics are quite straight forward, and there are many variants to choose from. The basics of insertion heuristics is to start with a tour of a subset of all cities, and then inserting the rest by some heuristic. The initial subtour is often a triangle. One can also start with a single edge as subtour. The complexity with this type of heuristic approach is given as $O(n^2)$. Steps of an Insertion heuristic are:

Select the shortest edge, and make a subtour of it.

1. Select a city not in the subtour, having the shortest distance to any one of the cities in the subtour.
2. Find an edge in the subtour such that the cost of inserting the selected city between the edge's cities will be minimal.
3. Repeat step 2 until no more cities remain.

### 5.2.4 Christofide heuristic

Most heuristics can only guarantee a feasible soluiton or a fair near optimal solution. Christofides extended one of these heuristic approaches which is known as Christofides heuristic. Complexity of this approach is $O(n^3)$. The steps are gievn below:

1. Build a minimal spanning tree from the set of all cities.
2. Create a minimum-weight matching (MWM) on the set of nodes having an odd degree. Add the MST together with the MWM.
3. Create an Euler cycle from the combined graph, and traverse it taking shortcuts to avoid visited nodes.

Tests have shown that Christofides' algorithm tends to place itself around 10% above the Held-Karp lower bound. More information on tour construction heuristics can be found in (Johnson & McGeoch, 2002).

### 5.3 Tour improvement

After generating the tour using any tour construction heuristic, an improvment heuristic can be further applied to improve the quality of the tour generated. Popularly, 2-opt and 3-opt exchange heuristic is applied for improving the solution. The performance of 2-opt or 3-opt heuristic basically depends on the tour generated by the tour construction heuristic. Other ways of improving the solution is to apply meta-heuristic approaches such as tabu search or simulated annealing using 2-opt and 3-opt.

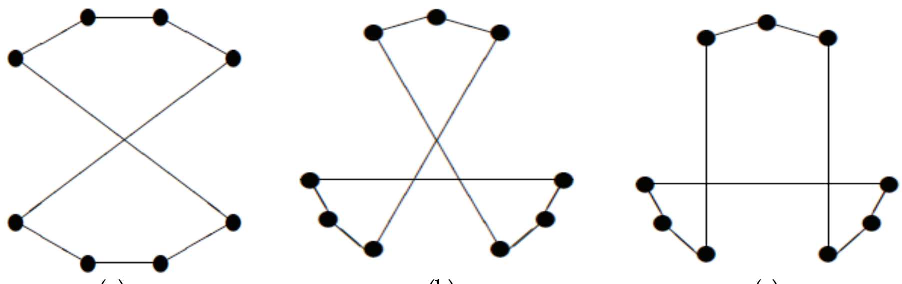



Fig. 1. A 2- opt move and 3-opt move

#### 5.3.1 2-opt and 3-opt

The 2-opt algorithm removes randomly two edges from the already generated tour, and reconnects the new two paths created. This is refered as a 2-opt move. The reconnecting is done such a way to keep the tour valid (see figure 1 (a)). This is done only if the new tour is shorter than older. This is continued till no further improvement is possible. The resulting tour is now 2 optimal. The 3-opt algorithm works in a similar fashion, but instead of removing the two edges it removes three edges. This means there are two ways of reconnecting the three paths into a valid tour (see figure 1(b) and figure 1(c)). Search is completed when no more 3-opt moves can improve the tour quality. If a tour is 3 optimal it is also 2 optimal (Helsgaun). Running the 2-opt move often results in a tour with a length less than 5% above the Held-Karp bound. The improvements of a 3-opt move usually generates a tour about 3% above the Held-Karp bound (Johnson & McGeoch, 1995).

#### 5.3.2 k-opt

In order to improve the already generated tour from tour construction heuristic, k-opt move can be applied (2-opt and 3-opt are special cases of k-opt exchange heuristic) but exchange heuristic having k>3 will take more computational time. Mainly one 4-opt move is used, called "the crossing bridges" (see Figure 2). This particular move cannot be sequentially constructed using 2-opt moves. For this to be possible two of these moves would have to be illegal (Helsgaun).

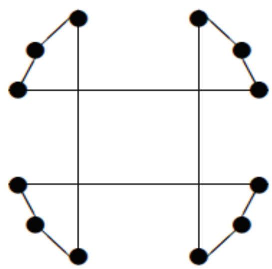


Fig. 2. Double bridge move

### 5.3.3 Lin-Kernighan

Lin & Kernighan constructed an algorithm making it possible to get within 2% of the Held-Karp lower bound. The Lin-Kernighan heuristic (LK) is a variable k-way exchange heuristic. It decides the value of suitable k at each iteration. This makes the an improvement heuristic quite complex, and few have been able to make improvements to it. The time complexity of LK is approximately $O\left(n^{2.2}\right)$ (Helsgaun), making it slower than a simple 2-opt implementation.

### 5.3.4 Tabu search

It is a neighborhood-search algorithm which seacrh the better solution in the neighbourhood of the existing solution. In general, tabu search (TS) uses 2-opt exchange mechanism for searching better solution. A problem with simple neighborhood search approach i.e. only 2-opt or 3-opt exchange heuristic is that these can easily get stuck in a local optimum. This can be avoided easily in TS approach. To avoid this TS keeps a tabu list containing bad solution with bad exchange. There are several ways of implementing the tabu list. For more detail paper by (Johnson & McGeoch, 1995) can be referred. The biggest problem with the TS is its running time. Most implementations for the TSP generally takes $O(n^3)$ (Johnson & McGeoch, 1995), making it far slower than a 2-opt local search.

### 5.3.5 Simulated annealing

Simulated Annealing (SA) has been successfully applied and adapted to give an approximate solutions for the TSP. SA is basically a randomized local search algorithm similar to TS but do not allow path exchange that deteriorates the solution. (Johnson & McGeoch, 1995) presented a baseline implementation of SA for the TSP. Authors used 2-opt moves to find neighboring solutions. In SA, Better results can be obtained by increasing the running time of the SA algorithm, and it is found that the results are comparable to the LK algorithm. Due to the 2-opt neighborhood, this particular implementation takes $O\left(n^2\right)$ with a large constant of proportionality (Johnson & McGeoch, 1995).

### 5.3.6 Genetic algorithm

Genetic Algorithm (GA) works in a way similar to the nature. A basic GA starts with a randomly generated population of candidate solutions. Some (or all) candidates are then mated to produce offspring and some go through a mutating process. Each candidate has a fitness value telling us how good they are. By selecting the most fit candidates for mating and mutation the overall fitness of the population will increase. Applying GA to the TSP involves implementing a crossover routine, a measure of fitness, and also a mutation routine. A good measure of fitness is the actual length of the solution. Different approaches to the crossover and mutation routines are discussed in (Johnson & McGeoch, 1995).

## 5.4 Ant colony optimization

Researchers are often trying to mimic nature to solve complex problems, and one such example is the successful use of GA. Another interesting idea is to mimic the movements of ants. This idea has been quite successful when applied to the TSP, giving optimal solutions to small problems quickly (Dorigo & Gambardella, 1996). However, as small as an ant's brain might be, it is still far too complex to simulate completely. But we only need a small part of their behaviour for solving the problem. Ants leave a trail of pheromones when they explore new areas. This trail is meant to guide other ants to possible food sources. The key to the success of ants is strength in numbers, and the same goes for ant colony optimization. We start with a group of ants, typically 20 or so. They are placed in random cities, and are then asked to move to another city. They are not allowed to enter a city already visited by themselves, unless they are heading for the completion of our tour. The ant who picked the shortest tour will be leaving a trail of pheromones inversely proportional to the length of the tour. This pheromone trail will be taken in account when an ant is choosing a city to move to, making it more prone to walk the path with the strongest pheromone trail. This process is repeated until a tour being short enough is found. Consult (Dorigo & Gambardella, 1996) for more detailed information on ant colony optimization for the TSP.

### 5.5 The Held-Karp lower bound

This lower bound if the common way of testing the performance of any new TSP heuristic. Held-Karp (HK) bound is actually a solution to the linear programming relaxation of the integer formulation of TSP (Johnson et al. 1996). A HK lower bound averages about 0.8% below the optimal tour length (Johnson et al., 1996). For more details regarding the HK lower bound, paper by (Johnson et al., 1996) can be referred.

### 5.6 Heuristic solution approaches for mTSP

One of the first heuristics addressing TSP is due to (Russell, 1977). The algorithm is an extended version of the Lin & Kernighan (1973) heuristic. (Potvin et al., 1989) have given another heuristic based on an exchange procedure for the mTSP. (Fogel, 1990) proposed a parallel processing approach to solve the mTSP using evolutionary programming. Problems with 25 and 50 cities were solved and it is noted that the evolutionary approach obtained very good near-optimal solutions. (Wacholder et al., 1989) extended the Hopfield-Tank ANN model to the mTSP but their model found to be too complex to find even feasible soultions. Hsu et al. (1991) presented a neural network (NN) approach to solve the mTSP. The authors stated that their results are better than (Wacholder et al., 1989). (Goldstein, 1990) and (Vakhutinsky & Golden, 1994) presented a self-organizing NN approach for the mTSP. A self-organizing NN for the VRP based on an enhanced mTSP NN model is due to (Torki et al., 1997). Recently, (Modares et al., 1999 and Somhom et al., 1999) have developed a self-organizing NN approach for the mTSP with a minmax objective function, which minimizes the cost of the most expensive route. Utilizing GA for the solution of mTSP seems to be first due to (Zhang et al., 1999). A recent application by (Tang et al., 2000) used GA to solve the mTSP model developed for hot rolling scheduling. (Yu et al., 2002) also used GA to solve the mTSP in path planning. (Ryan et al., 1998) used TS in solving a mTSP with time windows. (Song et al., 2003) proposed an extended SA approach for the mTSP with fixed costs associated with each salesman. (Gomes & Von Zuben, 2002) presented a neuro-fuzzy system based on competitive learning to solve the mTSP along with the capacitated VRP. Sofge et al. (2002) implemented and compared a variety of evolutionary computation algorithms to solve the mTSP, including the use of a neighborhood attractor schema, the shrink-wrap algorithm for local neighborhood optimization, particle swarm optimization, Monte-Carlo optimization, genetic algorithms and evolutionary strategies. For more detailed description, papers mentioned above can be referred.

## 6. References

Ali, A.I. & Kennington, J. L. (1986). The asymmetric *m*-traveling salesmen problem: a duality based branch-and-bound algorithm. *Discrete Applied Mathematics*, Vol. No. 13, pp. 259–76.

Angel, R.D.; Caudle, W.L.; Noonan, R. & Whinston, A. (1972). Computer assisted school bus scheduling. *Management Science,* Vol. 18, pp.279–88.

Applegate, D.L., Bixby, R.E., Chv´atal, V. & Cook, W.J. (2003). Implementing the Dantzig–Fulkerson–Johnson algorithm for large scale traveling salesman problems. *Math Program Ser B* Vol. 97, pp. 91–153.

Applegate, D. L.; Bixby, R. E.; Chvátal, V. & Cook, W. J. (2006), *The Traveling Salesman Problem: A Computational Study*, Princeton University Press, ISBN 978-0-691-12993-8.

Applegate, D.L.; Bixby, R.E.; Chv´atal, V.; Cook, W.J.; Espinoza, D.G.; Goycoolea, M. & Helsgaun, K. (2009). Certification of an optimal TSP tour through 85900 cities. *Operations Reearch Letters.*, Vol. 37, No. 1, pp. 11–15.

Arora, S. (1998). Polynomial Time Approximation Schemes for Euclidian Traveling Salesman and Other Geometric Problems, *Journal of the ACM*, Vol. 45, No. 5, September 1998, pp. 753-782.

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

Basu, A.; Elnagar, A. & Al-Hajj, A. (2000). Efficient coordinated motion. *Mathematical and Computer Modelling,* Vol. 31, pp. 39–53.

Bektas, T. (2006). The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega,* Vol. 34, pp. 206-219.

Biggs N. L.; Lloyd E. Keith & Wilson Robin J. (1986). *Graph Theory 1736-1936,* Clarendon Press, Oxford, ISBN 978-0-19-853916-2.

Bland, R.E., & D.E Shallcross (1989). Large traveling salesman problem arising from experiments in X-ray crystallography: a preliminary report on computation. *Operations Research Letters,* Vol. 8(3), pp. 125-128.

Brummit, B. & Stentz, A. (1996). Dynamic mission planning for multiple mobile robots. *Proceedings of the IEEE international conference on robotics and automation,* April 1996.

Brummit, B. & Stentz, A. (1998). GRAMMPS: a generalized mission planner for multiple mobile robots. *Proceedings of the IEEE international conference on robotics and automation,* May 1998.

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

C. Song; K. Lee & W.D. Lee (2003). Extended simulated annealing for augmented TSP and multi-salesmen TSP. *Proceedings of the international joint conference on neural networks*, Vol. 3, pp. 2340–43.

Calvo, R.W. & Cordone, R. (2003) A heuristic approach to the overnight security service problem. *Computers and Operations Research*, Vol. 30, pp.1269–87.

Carpaneto, G. & Toth, P. (1980). Some new branching and bounding criteria for the asymmetric traveling salesman problem. *Management Science,* Vol. 26, pp. 736–743.

Carpaneto, G. & Toth, P. (1987). Primal-dual algorithms for the assignment problem. Discrete Applied Mathematics, Vol. 18, pp. 137–153.

Carpaneto, G.; Dell'Amico, M. & Toth, P. (1995). Exact solution of large-scale, asymmetric travelling salesman problems. *ACM Transactions on Mathematical Software,* Vol. 21, pp. 394–409.

Carter, A.E. & Ragsdale, C.T. (2002). Scheduling pre-printed newspaper advertising inserts using genetic algorithms. *Omega*, Vol. 30, pp. 415–21.

Christofides, N.; Mingozzi, A. & Toth, P. (1981). Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical Programming*, Vol.20, pp. 255–82.

Chv´atal, V. (1973). Edmonds polytopes and weakly Hamiltonian graphs. *Mathematical Programming*, Vol. 5, pp. 29–40.

Cornu´ejols, G.; Fonlupt, J. & Naddef, D. (1985). The travelling salesman problem on a graph and some related integer polyhedra. *Mathematical Programming*, Vol. 33, pp. 1–27.

Crowder, H. & Padberg, M.W. (1980). Solving large-scale symmetric traveling salesman problems to optimality. *Management Science,* Vol. 26, pp. 495–509.

D. Applegate; W. Cook & A. Rohe (2000). *Chained Lin-Kernighan for large traveling salesman problems*, July 27, 2000.

Dantzig, G.B.; Fulkerson, D.R. & Johnson, S.M. (1954). Solution of a large-scale traveling salesman problem. *Operations Research*, Vol. 2, pp.393–410.

Dell'Amico, M. & Toth, P. (2000). Algorithms and codes for dense assignment problems: The state of the art. *Discrete Applied Mathematics,* Vol. 100, No. 1-2, pp. 17–48.

Dorigo, M. & Gambardella, L.M. (1996). "*Ant Colonies for the Traveling Salesman Problem*", University Libre de Bruxelles, Belgium.

Dreissig, W. & W. Uebaeh (1990). *Personal communication*.

Eastman, W.L. (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, Vol.* 69B, p. 125–130.

Fischetti, M. & Toth, P. (1992). An additive bounding procedure for the asymmetric traveling salesman problem. *Mathematical Programming: Series A and B, Vol.* 53(2) , pp. 173–197.

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

Fogel, D.B. (1990). A parallel processing approach to a multiple travelling salesman problem using evolutionary programming. In: *Proceedings of the fourth annual symposium on parallel processing. Fullerton*, CA, pp. 318–26.

Gavish, B. (1976). A note on the formulation of the *m*salesman traveling salesman problem. *Management Science*, Vol. 22, No. 6, pp.704–5.

Gavish, B. & Srikanth, K. (1986). An optimal solution method for large-scale multiple traveling salesman problems. *Operations Research*, Vol. 34, No. 5, pp. 698–717.

Gilbert, K.C. & Hofstra, R.B. (1992). A new multiperiod multiple traveling salesman problem with heuristic and application to a scheduling problem. *Decision Sciences*, Vol. 23, pp.250–9.

Goldstein, M. (1990). Self-organizing feature maps for the multiple traveling salesmen problem. In: *Proceedings of the IEEE international conference on neural network.* San Diego, CA, pp. 258–61.

Gomes, L.D.T. & Von Zuben, F.J. (2002). Multiple criteria optimization based on unsupervised learning and fuzzy inference applied to the vehicle routing problem. *Journal of Intelligent and Fuzzy Systems,* Vol. 13, pp. 143–54.

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

Gorenstein, S. (1970). Printing press scheduling for multi-edition periodicals. *Management Science,* Vol. 16, No. 6, pp.B373–83.

Grötschel, M. & Padberg, M.W. (1985). Polyhedral theory. In: Lawler E.L.; Lenstra J.K.; Rinnooy Kan AHG & Shmoys D.B. (eds). *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley: Chichester, pp 251–305.

Grötschel, M. & Pulleyblank, W.R. (1986). Clique tree inequalities and the symmetric Travelling Salesman Problem. *Mathematics of Operations Research*, Vol. 11, No. 4, (November, 1986), pp. 537–569.

Grötschel, M. & O. Holland (1991). Solution of Large-scale Symmetric Traveling Salesman Problems. *Mathematical Programming,* Vol. 51, pp.141-202.

Grötschel, M.; M. Jünger & G. Reinelt (1991). Optimal Control of Plotting and Drilling Machines: A Case Study. *Mathematical Methods of Operations Research,* Vol. 35, No. 1, (January, 1991), pp.61-84.

Gromicho, J.; Paixão, J. & Branco, I. (1992). Exact solution of multiple traveling salesman problems. In: Mustafa Akgül, et al., editors. *Combinatorial optimization. NATO ASI Series*, Vol. F82. Berlin: Springer; 1992. p. 291–92.

Gromicho, J. (2003). *Private communication*.

Hadjiconstantinou, E. & Roberts, D. (2002). Routing under uncertainty: an application in the scheduling of field service engineers. In: Paolo Toth, Daniele Vigo, editors. *The vehicle routing problem. SIAM Monographs on Discrete Mathematics and Applications*, Philadelphia, pp. 331–52.

Hsu, C.; Tsai, M. & Chen, W. (1991). A study of feature-mapped approach to the multiple travelling salesmen problem. *IEEE International Symposium on Circuits and Systems,* Vol. 3, pp. 1589–92.

Johnson D.S. & McGeoch L.A. (1995). *The Traveling Salesman Problem: A Case Study in Local Optimization*, November 20, 1995.

Johnson, D.S; McGeoch, L.A. & Rothberg E.E (1996). Asymptotic Experimental Analysis for the Held- Karp Traveling Salesman Boun. *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 341-350.

Johnson, D.S. & McGeoch, L.A.(2002). Experimental Analysis of Heuristics for the STSP, *The Traveling Salesman Problem and its Variations*, Gutin & Punnen (eds), Kluwer Academic Publishers, pp. 369-443.

K. Helsgaun. *An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic*, Department of Computer Science, Roskilde University.

Kim, K.H. & Park, Y. (2004). A crane scheduling method for port container terminals. *European Journal of Operational Research*, Vol. 156, pp. 752–68.

Kulkarni, R.V. & Bhave, P.R. (1985). Integer programming formulations of vehicle routing problems. *European Journal of Operational Research*, Vol. 20, pp. 58–67.

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

Land, A.H. & Doig, A.G. (1960). An automatic method of solving discrete programming problems. *Econometrica, Vol.* 28, pp. 497–520.

Laporte, G. & Nobert, Y. (1980). A cutting planes algorithm for the *m*-salesmen problem. *Journal of the Operational Research Society*, Vol. 31, pp.1017–23.

Laporte, G.; Nobert, Y. & Desrochers, M. (1985). Optimal routing under capacity and distance restrictions. *Operations Research*, Vol. 33, No. 5, pp.1050–73.

Laporte, G. (1992). The vehicle routing problem: an overview of exact and approximate algorithms. *European Journal of Operational Research*, Vol. 59, No. 3, pp. 345–58.

Laporte, G. (1992). The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, Vol. 59, No. 2, pp. 231-247.

Lenstra, J.K. & A.H.G. Rinnooy Kan (1974). *Some Simple Applications of the Travelling Salesman Problem.* BW 38/74, Stichting Mathematisch Centrum, Amsterdam.

Lenstra, J.K,. & Rinnooy Kan, A.H.G. (1975). Some simple applications of the traveling salesman problem. *Operational Research Quarterly*, Vol. 26, pp. 717–33.

Little, J.D.C.; Murty, K.G.; Sweeney, D.W. & Karel, C. (1963). An algorithm for the traveling salesman problem. *Operations Research*, Vol. 11, pp. 972–989.

Lin, S. & Kernighan, B. (1973). An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, Vol. 21, pp. 498–516.

Macharis, C. & Bontekoning, Y.M.( 2004) Opportunities for OR in intermodal freight transport research: a review. *European Journal of Operational Research*, Vol. 153, pp.400–16.

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

Miliotis, P. (1976). Integer programmi.ng approaches to the travelling salesman problem. *Mathematical Programming,* Vol. 10, pp. 376–378.

Miliotis, P. (1978). Using cutting planes to solve the symmetric travelling salesman problem. *Mathematical Programming,* Vol. 15, pp. 177–188.

Miller, C.E.; Tucker, A.W. & Zemlin, R.A.(1960). Integer programming formulation of traveling salesman problems. *Journal of Association for Computing Machinery*, Vol. 7, pp. 326–9.

Mitrović-Minić, S.; Krishnamurti, R. & Laporte, G. (2004). Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. *Transportation Research*, Vol. 28, No. 8, pp. 669–85.

Modares, A.; Somhom, S. & Enkawa, T.(1999). A self-organizing neural network approach for multiple traveling salesman and vehicle routing problems. *International Transactions in Operational Research,* Vol. 6, pp. 591–606.

Mole, R.H.; Johnson, D.G. & Wells, K. (1983). Combinatorial analysis for route first-cluster second vehicle routing. *Omega*, Vol. 11, No. 5, pp. 507–12.

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

Okonjo-Adigwe, C. (1988). An effective method of balancing the workload amongst salesmen. *Omega*, Vol. 16, No. 2, pp. 159–63.

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

O¨ncan, T.; Altınel, I.K. & Laporte, G. (2009). A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers & Operations Research, Vol.* 36, pp. 637–654.

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

Padberg, M.W. & Hong, S. (1980). On the symmetric travelling salesman problem: A computational study. *Mathematical Programming Study,* Vol. 12, pp. 78–107.

Padberg, M.W. & Rinaldi, G. (1987). Optimization of a 532-city symmetric traveling salesman problem by branch and cut. *Operations Research Letters,* Vol. 6, No. 1, pp. 1–7.

Padberg, M.W. & Rinaldi, G. (1991). A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review,* Vol. 33, pp. 60–100.

Plante, R.D.; T.J. Lowe & R. Chandrasekaran (1987). The Product Matrix Traveling Salesman Problem: An Application and Solution Heuristics. *Operations Research,* Vol. 35, pp. 772-783.

Potvin, J.; Lapalme, G. & Rousseau, J. (1989). A generalized k-opt exchange procedure for the MTSP. *Information Systems* 1989;27(4): 474–81.

Ralphs, T.K. (2003). Parallel branch and cut for capacitated vehicle routing. *Parallel Computing*, Vol. 29, pp. 607–29.

Ratliff, H.D. & A.S. Rosenthal (1983). Order-Picking in a Rectangular Warehouse: A Solvable Case for the Travelling Salesman Problem. *Operations Research,* Vol. 31, pp. 507-521.

Ruland, K.S. & Rodin, E.Y.(1997). The pickup and delivery problem. *Computers and Mathematics with Applications,* Vol. 33, No. 12, pp. 1–13.

Russell, R.A. (1997). An effective heuristic for the *m*-tour traveling salesman problem with some side conditions. *Operations Research*, Vol. 25, No. 3, pp. 517–24.

Ryan, J.L.; Bailey, T.G.; Moore, J.T & Carlton, W.B. (1998). Reactive Tabu search in unmanned aerial reconnaissance simulations. *Proceedings of the 1998 winter simulation conference,* Vol. 1, pp . 873–9.

Saleh, H.A. & Chelouah, R. (2004). The design of the global navigation satellite system surveying networks using genetic algorithms. *Engineering Applications of Artificial Intelligence,* Vol. 17, pp. 111–22.

Schrijver, A. (1960). On the history of combinatorioal optimization.

Sofge, D.; Schultz, A. & De Jong, K. (2002). Evolutionary computational approaches to solving the multiple traveling salesman problem using a neighborhood attractor schema. *Lecture notes in computer science*, Vol. 2279, pp. 51–60.

Somhom, S.; Modares, A. & Enkawa, T. (1999). Competition-based neural network for the multiple traveling salesmen problem with minmax objective. *Computers and Operations Research,* Vol. 26, No. 4, pp. 395–407.

Svestka, J.A. & Huckfeldt, V.E. (1973). Computational experience with an *m*-salesman traveling salesman algorithm. *Management Science*, Vol. 19, No. 7, pp. 790–9.

Tang, L.; Liu, J.; Rong, A. & Yang, Z. (2000).A multiple traveling salesman problem model for hot rolling scheduling in Shangai Baoshan Iron & Steel Complex. *European Journal of Operational Research*, Vol. 124, pp. 267–82.

Torki, A.; Somhon, S. & Enkawa, T. (1997). A competitive neural network algorithm for solving vehicle routing problem. *Computers and Industrial Engineering*, Vol. 33, No. 3–4, pp. 473–6.

Toth, P. & Vigo, D. (2002). Branch-and-bound algorithms for the capacitated VRP. In: Paolo Toth, Daniele Vigo, editors. The vehicle routing problem. *SIAM Monographs on Discrete Mathematics and Applications*, Philadelphia, pp.29–51.

van Dal, R. (1992). *Special Cases of the Traveling Salesman Problem*, Wolters-Noordhoff, Groningen.

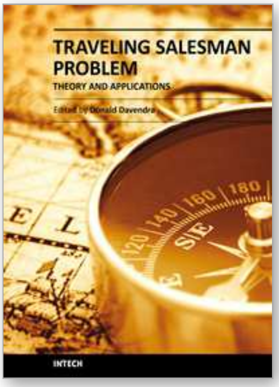
Vakhutinsky, I.A. & Golden, L.B. (1994). Solving vehicle routing problems using elastic net. *Proceedings of the IEEE international conference on neural network*, pp. 4535–40.

Wacholder, E.; Han, J. & Mann, R.C. (1989). A neural network algorithm for the multiple traveling salesmen problem. *Biology in Cybernetics*, Vol. 61, pp. 11–9.

Wang, X. & Regan, A.C. (2002). Local truckload pickup and delivery with hard time window constraints. *Transportation Research* Part B, Vol. 36, pp. 97–112.

Yu, Z.; Jinhai, L.; Guochang, G.; Rubo, Z. & Haiyan, Y. (2002). An implementation of evolutionary computation for path planning of cooperative mobile robots.

*Proceedings of the fourth world congress on intelligent control and automation*, Vol. 3, pp. 1798–802.
Zhang, T.; Gruver, W.A. & Smith, M.H. (1999). Team scheduling by genetic search. *Proceedings of the second international conference on intelligent processing and manufacturing of materials*, Vol. 2,. pp. 839–44.




**Traveling Salesman Problem, Theory and Applications**
Edited by Prof. Donald Davendra

ISBN 978-953-307-426-9
Hard cover, 298 pages
**Publisher** InTech
**Published online** 30, November, 2010
**Published in print edition** November, 2010

This book is a collection of current research in the application of evolutionary algorithms and other optimal algorithms to solving the TSP problem. It brings together researchers with applications in Artificial Immune Systems, Genetic Algorithms, Neural Networks and Differential Evolution Algorithm. Hybrid systems, like Fuzzy Maps, Chaotic Maps and Parallelized TSP are also presented. Most importantly, this book presents both theoretical as well as practical applications of TSP, which will be a vital tool for researchers and graduate entry students in the field of applied Mathematics, Computing Science and Engineering.



**InTech Europe**
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

**InTech China**
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821