

Models and algorithms for the Asymmetric Traveling Salesman Problem: an experimental comparison

Roberto Roberti · Paolo Toth

Received: 15 October 2011 / Accepted: 25 April 2012 / Published online: 15 May 2012
© Springer-Verlag + EURO - The Association of European Operational Research Societies 2012

Abstract This paper surveys the most effective mathematical models and exact algorithms proposed for finding the optimal solution of the well-known Asymmetric Traveling Salesman Problem (ATSP). The fundamental Integer Linear Programming (ILP) model proposed by Dantzig, Fulkerson and Johnson is first presented, its classical (assignment, shortest spanning r -arborescence, linear programming) relaxations are derived, and the most effective branch-and-bound and branch-and-cut algorithms are described. The polynomial ILP formulations proposed for the ATSP are then presented and analyzed. The considered algorithms and formulations are finally experimentally compared on a set of benchmark instances.

Keywords Asymmetric Traveling Salesman Problem · Integer Linear Programming models · Branch-and-bound algorithms · Branch-and-cut algorithms · Experimental comparison

Introduction

Let $G = (V, A)$ be a given complete digraph, where $V = \{1, \dots, n\}$ is the vertex set and $A = \{(i, j) : i, j \in V\}$ the arc set, and let c_{ij} be the cost associated with arc $(i, j) \in A$ (with $c_{ii} = +\infty$, for $i \in V$). A *Hamiltonian circuit (tour)* of G is a circuit visiting each vertex of V exactly once. The Asymmetric Traveling Salesman Problem (ATSP) is to find a Hamiltonian circuit $G^* = (V, A^*)$ of G whose cost $\sum_{(i,j) \in A^*} c_{ij}$ is minimum. If the considered graph G is undirected, the corresponding problem is denoted as Symmetric Traveling Salesman Problem (STSP).

R. Roberti · P. Toth (✉)
DEIS, University of Bologna, Viale Risorgimento, 2, 40136 Bologna (BO), Italy
e-mail: paolo.toth@unibo.it

R. Roberti
e-mail: roberto.roberti6@unibo.it

The ATSP is known to be NP-hard in the strong sense, and has been intensively studied in the last six decades. In this paper, we will consider and experimentally compare the most effective Integer Linear Programming (ILP) models and exact algorithms proposed for finding the optimal solution of the ATSP. Previous surveys on the subject have been presented by Balas and Toth (1985), Fischetti et al. (2002), Öncan et al. (2009), D’Ambrosio et al. (2010). Several books dealing with the traveling salesman problem and its variations have been published. Among them we mention those by Lawler et al. (1985), Reinelt (1994), Gutin and Punnen (2002), Applegate et al. (2007).

The paper is organized as follows. In “[The Dantzig-Fulkerson-Johnson formulation and its relaxations](#)”, the well-known Dantzig, Fulkerson and Johnson formulation Dantzig et al. (1954) and its “classical” relaxations are described. Polynomial formulations, i.e., formulations requiring a number of constraints polynomial in the number of vertices n , are described in “[Review of polynomial formulations](#)”. “[Exact Algorithms](#)” reviews the most effective branch-and-bound and branch-and-cut algorithms proposed for the ATSP. In “[Transformation of ATSP instances into STSP instances](#)”, the transformation of an ATSP instance into an equivalent STSP instance, proposed by Jonker and Volgenant (1983), is presented. Computational experiments, comparing the considered formulations and exact algorithms on a set of benchmark instances, are described in “[Computational results](#)”.

The Dantzig–Fulkerson–Johnson formulation and its relaxations

Dantzig et al. (1954) proposed the following ILP model (hereafter DFJ), utilizing n^2 binary variables x_{ij} , for the ATSP:

$$(DFJ) \quad \min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (1)$$

$$\text{s.t.} \quad \sum_{i=1}^n x_{ij} = 1, \quad j = 1, \dots, n, \quad (2)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n, \quad (3)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \quad S \subset V : S \neq \emptyset, \quad (4)$$

$$x_{ij} \in \{0, 1\}, \quad i, j = 1, \dots, n, \quad (5)$$

where x_{ij} is equal to 1 if and only if arc (i, j) ($i = 1, \dots, n; j = 1, \dots, n$) is in the optimal tour. Constraints (2) and (3) impose that the in-degree and out-degree of each vertex, respectively, is equal to one, while constraints (4) are Subtour Elimination Constraints (SECs) and impose that no partial circuit exists.

Moreover, it is well known that one can halve the number of SECs (4) by replacing them with

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \quad S \subset V \setminus \{r\} : S \neq \emptyset,$$

where r is any vertex of vertex set V .

Because of constraints (2) and (3), constraints (4) can be equivalently rewritten as *Connectivity constraints*:

$$\sum_{i \in S} \sum_{j \in V \setminus S} x_{ij} \geq 1, \quad S \subset V : S \neq \emptyset. \tag{6}$$

Also in this case, one can halve the number of connectivity constraints (6) by replacing them with

$$\sum_{i \in S} \sum_{j \in V \setminus S} x_{ij} \geq 1, \quad S \subset V : r \in S \tag{7}$$

or with

$$\sum_{i \in S} \sum_{j \in V \setminus S} x_{ij} \geq 1, \quad S \subset V : S \neq \emptyset, r \notin S \tag{8}$$

where r is any fixed vertex.

A valid lower bound on the optimal solution value of the ATSP can be obtained by optimally solving the *Linear Programming* (LP) relaxation of the previous models (1)–(5) or (1)–(3), (5) and (7), obtained by replacing constraints (5) with constraints

$$x_{ij} \geq 0, \quad i, j = 1, \dots, n. \tag{9}$$

Although the considered ILP models require an exponential number of Subtour Elimination or Connectivity constraints, their LP relaxations can be efficiently solved in polynomial time using the effective polynomial *separation procedure* proposed by Padberg and Rinaldi (1990a) for the STSP.

Additional lower bounds can be obtained by considering the different substructures of the ATSP, each associated with a subset of constraints defining a well-structured relaxation.

Constraints (2), (3) and (9), with objective function (1), define the well-known *min-sum Assignment Problem* (AP). Such a problem always has an integer optimal solution and requires the finding of a minimum-cost collection of vertex-disjoint subtours visiting all the vertices of G . Relaxation AP can be solved in $O(n^3)$ time (see, e.g., Lawler 1976; Carpaneto and Toth 1987 for an efficient implementation).

Constraints (2), (7) and (9), with objective function (1), define the well known shortest Spanning r -Arborescence Problem (r -SAP). Such a problem always has an integer optimal solution, and corresponds to find a minimum-cost spanning subdigraph $\bar{G} = (V, \bar{A})$ of G such that (1) the in-degree of each vertex is exactly one, and (2) each vertex can be reached from the root vertex r . Relaxation r -SAP can be solved in $O(n^2)$ time by finding the shortest spanning arborescence rooted at

vertex r (see, e.g., Edmonds 1967; Tarjan 1977; Fischetti and Toth 1993 for an efficient implementation) and adding the minimum-cost arc entering vertex r .

A third substructure, corresponding to constraints (3), (8) and (9), with objective function (1), defines the shortest Spanning r -Antiarborescence Problem (r -SAAP). Such a problem can easily be transformed into r -SAP by simply transposing the input cost matrix, hence it can be solved in $O(n^2)$ time. Different choices of the root vertex r generally produce different values of the lower bounds corresponding to relaxations r -SAP and r -SAAP. In addition, these relaxations can be strengthened by considering the associated Lagrangian relaxations, obtained by embedding, in a Lagrangian fashion, the relaxed constraints (3) for r -SAP, and (2) for r -SAAP, into the objective function (1). Near optimal Lagrangian multipliers, leading to good lower bounds, can be obtained by applying the well-known *subgradient optimization* procedure proposed by Held and Karp (1970) and (1971) for the STSP.

The lower bounds corresponding to relaxations AP, r -SAP and r -SAAP can be also improved, as proposed by Fischetti and Toth (1992), by combining the associated substructures according to the so-called *additive approach* introduced by Fischetti and Toth (1989).

Review of polynomial formulations

In this section, we consider the papers presenting polynomial formulations for the ATSP. For each paper, we focus on the formulation producing the tightest LP-relaxation lower bound. Unlike the exact algorithms described in “[Exact algorithms](#)” the polynomial formulations can be directly solved by a general-purpose ILP solver. Classifications and comparisons of the polynomial formulations for the ATSP have been recently presented in Öncan et al. (2009) and Godinho et al. (2011b).

The earliest polynomial formulation of the ATSP is owed to Miller et al. (1960) (hereafter MTZ) and is given by (1)–(3), (5) and

$$u_i - u_j + (n - 1)x_{ij} \leq n - 2, \quad i, j = 2, \dots, n, \quad (10)$$

where $u_i, i = 2, \dots, n$, is an arbitrary real number representing the order of vertex i in the optimal tour, and constraints (10) break subtours. Miller et al. (1960) originally proposed MTZ without any bound on variables u_i . Later on, simple bounds (e.g., $1 \leq u_i \leq n - 1, i = 2, \dots, n$) were introduced to restrict the range of variables u_i . This does not affect the LP bound of MTZ and, in our computational experiments, has shown to increase the computing time, so we leave variables u_i unrestricted.

Gavish and Graves (1978) proposed another formulation (hereafter GG) having LP relaxation stronger than that of MTZ (see Wong 1980; Padberg and Sung 1991) but weaker than that of DFJ (see Gouveia 1995). GG is a *single-commodity flow formulation*, where subtours are broken by introducing $n(n - 1)$ nonnegative variables g_{ij} ($i = 2, \dots, n; j = 1, \dots, n$). GG consists of constraints (1)–(3), (5) and the following constraints:

$$\sum_{j=1}^n g_{ij} - \sum_{j=2}^n g_{ji} = 1, \quad i = 2, \dots, n, \tag{11}$$

$$0 \leq g_{ij} \leq (n - 1)x_{ij}, \quad i = 2, \dots, n, j = 1, \dots, n, \tag{12}$$

where variables g_{ij} can be interpreted as the number of arcs on the path from vertex 1 to arc (i, j) in the optimal tour (see Gouveia and Pires 1999). For fixed values of the variables x_{ij} , constraints (11) and (12) form a network flow problem and, therefore, variables g_{ij} take integer values. The LP relaxation of GG was shown to be equivalent to that of the *two-commodity flow formulation* proposed by Finke et al. (1984) (see Langevin et al. 1990, hereafter FCG).

Fox et al. (1980) proposed three formulations for the *time-dependent traveling salesman problem* that are valid for the ATSP. These formulations present n^3 binary variables, r_{ijk} , that are equal to 1 if and only if arc (i, j) is in position k in the optimal tour. The first formulation (therein P1) has $4n$ constraints, while the second formulation (hereafter P1b) has $3n$ constraints and is obtained from the first one by dropping a set of n constraints. Thus, the LP relaxation of the first formulation is stronger than that of the second formulation; Gouveia and Voss (1995) showed that the LP relaxations of both formulations are stronger than that of GG. The third formulation proposed (therein P2) has only $n + 1$ constraints and is obtained from the first formulation by surrogating the first $3n$ constraints. Öncan et al. (2009) showed that the LP relaxation of the third formulation is weaker than that of the second formulation, whereas Padberg and Sung (1991) proved that the LP relaxation of the third formulation is also weaker than that of DFJ. We have not reported the three formulations because, in our computational experiments, they proved to be ineffective in solving the ATSP.

The first multi-commodity flow (MCF) formulation was proposed by Wong (1980) (hereafter WONG). WONG considers $2(n - 1)$ commodities and introduces $2(n^3 - n^2)$ nonnegative continuous variables and $4n^3 - 2n^2 - 2n$ constraints. This formulation was later modified by Langevin (1988) and by Loulou (1988) to obtain two additional MCF formulations which provide LP relaxations equivalent to that of WONG.

Another MCF formulation with only $n - 1$ commodities was proposed by Claus (1984). This formulation (CLAUS) introduces $n^3 - n^2$ nonnegative continuous variables $w_{ij}^k (i, j = 1, \dots, n; k = 2, \dots, n)$ and $2n^3 - n^2 - n$ constraints and consists of (1)–(3), (5) plus the following constraints to break subtours:

$$\sum_{j=1}^n w_{ij}^k - \sum_{j=1}^n w_{ji}^k = 0, \quad i, k = 2, \dots, n : i \neq k, \tag{13}$$

$$\sum_{j=2}^n w_{1j}^k - \sum_{j=2}^n w_{j1}^k = -1, \quad k = 2, \dots, n, \tag{14}$$

$$\sum_{j=1}^n w_{ij}^i - \sum_{j=1}^n w_{ji}^i = 1, \quad i = 2, \dots, n, \tag{15}$$

$$0 \leq w_{ij}^k \leq x_{ij}, \quad i, j = 1, \dots, n, k = 2, \dots, n, \tag{16}$$

where variable w_{ij}^k is equal to 1 if and only if the commodity going from vertex 1 to vertex k flows on arc (i, j) . Langevin et al. (1990) proved that the LP relaxation of CLAUS is equivalent to that of WONG, whereas Padberg and Sung (1991) proved that the LP relaxation of CLAUS is also equivalent to that of DFJ.

Formulation MTZ was strengthened by Desrochers and Laporte (1990), who proposed formulation DL having LP relaxation stronger than that of MTZ and obtained from MTZ by replacing constraints (10) with the following lifted constraints:

$$u_i - u_j + (n - 1)x_{ij} + (n - 3)x_{ji} \leq n - 2, \quad i, j = 2, \dots, n, \tag{17}$$

$$-u_i + (n - 3)x_{i1} + \sum_{j=2}^n x_{ji} \leq -1, \quad i = 2, \dots, n, \tag{18}$$

$$u_i + (n - 3)x_{i1} + \sum_{j=2}^n x_{ij} \leq n - 1, \quad i = 2, \dots, n. \tag{19}$$

Gouveia and Pires (1999) presented four formulations of the ATSP (therein called RMTZ, L1RMTZ, L2RMTZ, and L3RMTZ). The LP relaxation of all these formulations is stronger than that of MTZ. Here, we present formulation L3RMTZ (hereafter GP), whose LP relaxation is stronger than those of both L1RMTZ and L2RMTZ, which in turn are stronger than that of RMTZ. Gouveia and Pires (1999) also showed that the LP relaxations of formulations L1RMTZ and L2RMTZ are weaker than those of MCF formulations, such as WONG and CLAUS. Formulation GP introduces $(n - 1)^2$ additional nonnegative continuous variables v_{ij} ($i, j = 2, \dots, n$) that are equal to 1 if and only if vertex i is in the path from vertex 1 to vertex j . GP consists of (1)–(3), (5) plus the following $2n^3 - 10n^2 + 18n - 10$ constraints to break subtours:

$$x_{ij} - v_{ij} \leq 0, \quad i, j = 2, \dots, n, \tag{20}$$

$$x_{ij} + v_{ji} \leq 1, \quad i, j = 2, \dots, n, \tag{21}$$

$$x_{ji} + x_{ij} + v_{ki} - v_{kj} \leq 1, \quad i, j, k = 2, \dots, n : i \neq j \neq k, \tag{22}$$

$$x_{kj} + x_{ik} + x_{ij} + v_{ki} - v_{kj} \leq 1, \quad i, j, k = 2, \dots, n : i \neq j \neq k. \tag{23}$$

Gouveia and Pires (2001) presented other formulations for the ATSP, among them a polynomial formulation (therein MCF+) whose LP relaxation is stronger than those of CLAUS and GP. As the LP relaxation of MCF+ is weaker than that of formulation SST (which will be introduced later in this section), as shown by Öncan et al. (2009), and MCF+ has more constraints than SST, we do not report a detailed description of MCF+.

Sherali and Driscoll (2002) strengthened formulation DL by applying a reformulation-linearization technique and introducing $(n - 1)^2$ additional nonnegative continuous variables y_{ij} ($i, j = 2, \dots, n$), where y_{ij} represents the order of arc (i, j) in the optimal tour. The resulting formulation (SD) replaces constraints (17)–(19) with the following $4(n^2 - n)$ constraints:

$$\sum_{j=2}^n y_{ij} + (n - 1)x_{i1} - u_i = 0, \quad i = 2, \dots, n, \tag{24}$$

$$\sum_{i=2}^n y_{ij} - u_j = -1, \quad j = 2, \dots, n, \tag{25}$$

$$x_{ij} - y_{ij} \leq 0, \quad i, j = 2, \dots, n, \tag{26}$$

$$y_{ij} - (n - 2)x_{ij} \leq 0, \quad i, j = 2, \dots, n, \tag{27}$$

$$u_j + (n - 2)x_{ij} + (n - 1)x_{ji} - y_{ij} - y_{ji} \leq n - 1, \quad i, j = 2, \dots, n, \tag{28}$$

$$y_{ij} + y_{ji} - u_j - x_{ji} \leq -1, \quad i, j = 2, \dots, n, \tag{29}$$

$$-x_{1j} + (n - 3)x_{j1} - u_j \leq -2, \quad j = 2, \dots, n, \tag{30}$$

$$(n - 3)x_{1j} - x_{j1} + u_j \leq n - 2, \quad j = 2, \dots, n. \tag{31}$$

Recently, Öncan et al. (2009) showed that the LP relaxation of SD is also stronger than that of GG.

Sarin et al. (2005) studied the ATSP with and without precedence constraints and proposed five polynomial formulations (therein ATSPxy, L1ATSPxy, SL1ATSPxy, L2ATSPxy and ML1ATSPxy) for the ATSP, whose LP relaxations are stronger than that of RMTZ. Moreover, Sarin et al. (2005) showed that the LP relaxation of L1ATSPxy is stronger than that of SL1ATSPxy whose LP relaxation is stronger than that of ATSPxy; the LP relaxations of L1ATSPxy and L2ATSPxy are also stronger than those of formulations L1RMTZ and L2RMTZ by Gouveia and Pires (1999), respectively. Here we report formulation L2ATSPxy (hereafter SSB) only, which, in our computational experiments, was shown to be the best performer. Formulation SSB introduces $(n - 1)^2$ nonnegative continuous variables d_{ij} ($i, j = 2, \dots, n$) and $n^3 - n^2 - n + 1$ constraints, and consists of (1)–(3), (5) and the following constraints to break subtours:

$$d_{ij} - x_{ij} \geq 0, \quad i, j = 2, \dots, n, \tag{32}$$

$$d_{ij} + d_{ji} = 1, \quad i, j = 2, \dots, n : i \neq j, \tag{33}$$

$$x_{1j} + x_{j1} \leq 1, \quad j = 2, \dots, n, \tag{34}$$

$$x_{ij} + d_{jk} + x_{kj} + d_{ki} + x_{ik} \leq 2, \quad i, j, k = 2, \dots, n, \tag{35}$$

Although variable d_{ij} ($i, j = 2, \dots, n$) is continuous, it has a binary connotation and is equal to 1 if and only if vertex i precedes (not necessarily immediately) vertex j in the optimal tour. Godinho et al. (2011b) noticed that the meaning of variables d_{ij} is basically the same of variables v_{ij} introduced in formulation GP. An analysis of all the formulations involving variables x_{ij} and v_{ij} (or d_{ij}) can be found in Gouveia and Pesneau (2006).

Sherali et al. (2006) proposed several polynomial formulations. Here, we present formulation SST (therein ATSP6) only, which uses $(n - 1)^3$ nonnegative continuous variables t_{ij}^k ($i, j, k = 2, \dots, n$), and consists of (1)–(3), (5), (33) and the following $3n^3 - 11n^2 + 17n - 9$ constraints:

$$d_{ij} + x_{ji} + d_{jk} + d_{ki} \leq 2, \quad i, j, k = 2, \dots, n, \tag{36}$$

$$d_{ij} - x_{1i} \geq 0, \quad i, j = 2, \dots, n, \tag{37}$$

$$d_{ji} - x_{i1} \geq 0, \quad i, j = 2, \dots, n, \tag{38}$$

$$0 \leq t_{ij}^k \leq x_{ik}, \quad i, j, k = 2, \dots, n : i \neq j \neq k, \tag{39}$$

$$x_{ij} + \sum_{k=2; k \neq j}^n t_{ij}^k = d_{ij}, \quad i, j = 2, \dots, n, \tag{40}$$

$$x_{1k} + \sum_{k=2; i \neq j}^n t_{ij}^k = d_{kj}, \quad k, j = 2, \dots, n, \tag{41}$$

where t_{ij}^k is equal to 1 if and only if, in the optimal tour, arc (i, k) is used and vertex k precedes vertex j . Godinho et al. (2011b) pointed out that variables t_{ij}^k can be interpreted in the same way as the flow variables w_{ij}^k used in formulation CLAUS. Öncan et al. (2009) showed that the LP relaxation of SST is stronger than that of MCF+, whereas Sherali et al. (2006) proved that the LP relaxation of SST is stronger than that of L1ATSPxy.

Godinho et al. (2011b) also described, in the context of the ATSP, a sophisticated formulation, called EC-MCF, that was originally proposed for the time-dependent traveling salesman problem by Godinho et al. (2011a). Let z_{ij}^h ($i, j, h = 1, \dots, n$) be a binary variable that is equal to 1 if arc (i, j) is traversed in position h in the optimal tour, and let r_{ij}^{hk} ($i = 1, \dots, n; j, k = 2, \dots, n; h = 1, \dots, n - 1$) be a binary variable that is equal to 1 if arc (i, j) is traversed in position h in the first part of the optimal tour that links vertex 1 to vertex k . Formulation EC-MCF involves the following $n^4 - 4n^3 + 9n^2 - 7n + 2$ variables and $2n^3 - 9n^2 + 15n - 8$ constraints:

$$\begin{aligned} & \min \sum_{i \in V} \sum_{j \in V} c_{ij} \sum_{h=1}^n z_{ij}^h \\ & \text{s.t. } \sum_{j \in V} r_{1j}^k = 1, \quad k \in V \setminus \{1\}, \\ & \sum_{j \in V \setminus \{1\}} r_{ij}^{h+1,k} - \sum_{j \in V \setminus \{k\}} r_{ji}^{hk} = 0, \quad h = 1, \dots, n - 2, k, i \in V \setminus \{1\} : i \neq k, \\ & \sum_{j \in V \setminus \{k\}} z_{kj}^{h+1} - \sum_{j \in V \setminus \{k\}} r_{jk}^{hk} = 0, \quad h = 1, \dots, n - 1, k = 2, \dots, n, \\ & \sum_{j \in V \setminus \{k\}} (z_{ij}^{h+1} - r_{ij}^{h+1,k}) - \sum_{j \in V \setminus \{1\}} (z_{ji}^h - r_{ji}^{hk}) = 0, \\ & \quad h = 2, \dots, n - 1, k, i \in V \setminus \{1\} : i \neq k, \\ & z_{ij}^h \in \{0, 1\}, \quad i, j, h = 1, \dots, n, \\ & r_{ij}^{hk} \in \{0, 1\}, \quad j, k \in V \setminus \{1\}, i \in V \setminus \{k\}, h = 1, \dots, n - 1. \end{aligned}$$

Godinho et al. (2011b) showed that the LP relaxation of EC-MCF is stronger than those of SD, P1, and CLAUS, and is unrelated with that of SST.

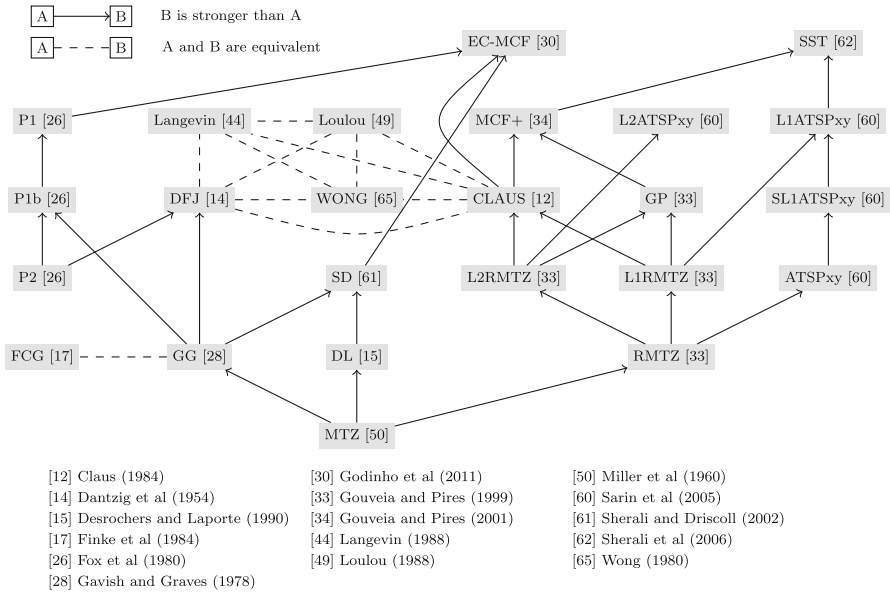


Fig. 1 Relations among the linear relaxations of the polynomial formulations and of formulation DFJ

In Fig. 1, we summarize the relationships among the linear relaxations of the polynomial formulations reviewed in this section. A link going from formulation A to formulation B means that the LP relaxation of B is stronger than that of A, whereas a dashed line connecting two formulations means that the relative LP relaxations are equivalent.

Exact algorithms

Many branch-and-bound algorithms have been proposed to find the optimal solution of the ATSP. After the seminal paper by Little et al. (1963), where for the first time the term “branch-and-bound” was coined, other algorithms were proposed by Bellmore and Malone (1971), Garfinkel (1973), Smith et al. (1977), Carpaneto and Toth (1980), Balas and Christofides (1981), Miller and Pekny (1989), Pekny et al. (1991), Pekny and Miller (1992), Fischetti and Toth (1992), Carpaneto et al. (1995). In the following, two of the most effective branch-and-bound algorithms for the ATSP, i.e., those proposed by Carpaneto et al. (1995) and by Fischetti and Toth (1992), are briefly reviewed. The algorithm proposed by Pekny and Miller (1992) exhibits, on the whole, a performance comparable with that of the approach described in Carpaneto et al. (1995).

The algorithm proposed by Carpaneto et al. (1995) is a lowest first branch-and-bound method based on the AP relaxation and the *subtour elimination branching scheme*. At the root node of the decision tree, the AP relaxation of the original problem is solved, the *patching heuristic* algorithm proposed by Karp (1979) is

applied to determine an initial tour of cost z^* , and a *reduction procedure* based on the AP reduced costs (c'_{ij}) is executed to transform the original complete graph into a sparse one (by setting $x_{ij} = 0$ if $V(\text{AP}) + c'_{ij} \geq z^*$, where $V(\text{AP})$ is the value of the optimal solution of the AP relaxation). At each of the other nodes of the decision tree, the AP relaxation of the subproblem associated with the considered node is solved, through an effective parametric technique, in $O(n^2)$ time. If $V(\text{AP}) \geq z^*$ the node is fathomed. Otherwise, if the AP solution contains no subtour (i.e. a feasible solution has been found) the best solution thus far is updated, z^* is set equal to $V(\text{AP})$ and the node is fathomed. If neither of the two previous cases occur, the subtour elimination branching scheme proposed by Carpaneto and Toth (1980) is applied: the subtour S of the AP solution having the minimum number, say h , of not “imposed” arcs is selected, and h descending nodes are generated so as to forbid, by “imposing” and “excluding” proper arc subsets, subtour S for each descending node.

The algorithm proposed by Fischetti and Toth (1992) is a lowest first branch-and-bound method based on the branching scheme introduced by Carpaneto and Toth (1980), and, at each node of the decision tree, computes the corresponding lower bound by applying the additive approach combining the AP, r-SAP, and r-SAAP relaxations.

More recently, two effective branch-and-cut algorithms for the ATSP have been proposed by Fischetti and Toth (1997) and by Fischetti et al. (2003).

The algorithm proposed by Fischetti and Toth (1997) is based on the DFJ model (1)–(5), and exploits additional classes of facet-inducing inequalities for the ATSP polytope that proved to be of crucial importance for the solution of some real-world instances.

An ATSP inequality $\alpha x \leq \alpha_0$ is called *symmetric* when $\alpha_{ij} = \alpha_{ji}$ for all $(i, j) \in A$. Symmetric inequalities can be thought of as derived from valid inequalities for the STSP defined on the complete undirected graph $G' = (V, E)$. Indeed, let $y_e = 1$ if edge $e \in E$ belongs to the optimal STSP solution, $y_e = 0$ otherwise. Every inequality $\sum_{e \in E} \alpha_e y_e \leq \alpha_0$ valid for the STSP can be transformed into a valid ATSP inequality by simply replacing y_e by $x_{ij} + x_{ji}$ for all edges $e = \{i, j\} \in E$. This produces the symmetric inequality $\alpha x \leq \alpha_0$, where $\alpha_{ij} = \alpha_{ji} = \alpha_{\{i, j\}}$ for all $i, j \in V, i \neq j$. Conversely, every symmetric ATSP inequality $\alpha x \leq \alpha_0$ corresponds to the valid STSP inequality $\sum_{\{i, j\} \in E} \alpha_{ij} y_{\{i, j\}} \leq \alpha_0$. The above correspondence implies that every separation algorithm for the STSP can be used, as a “black box”, for the ATSP as well. Several exact/heuristic separation algorithms for the STSP have been proposed in recent years, all of which can be used for the ATSP. Only two such separation tools are used by Fischetti and Toth (1997), namely (1) the Padberg and Rinaldi (1990a) exact algorithm for SECs; and (2) the simplest heuristic scheme for 2-matching constraints, i.e., for combs with 2-node teeth, where each component H of the graph induced by the edges $e \in E$ with fractional y_e^* is heuristically considered, in turn, as the handle of the comb. Having fixed H , the most violated 2-matching constraint with handle H is easily found by sorting the edges having one extreme node in H and the other extreme node in $V \setminus H$ by nonincreasing y_{ij}^k , and by taking the first k such edges to act as teeth, for $k = 1, 3, 5, \dots$

In addition, Fischetti and Toth (1997) considered the so-called D_k^+ and D_k^- inequalities proposed by Grötschel and Padberg (1985), and the odd *close alternative trail* (odd CAT) inequalities proposed by Balas (1989) (and analyzed by Fischetti 1991). The separation problem for the classes of the D_k^+ and D_k^- inequalities is a combinatorial optimization problem that can be effectively solved in practice by an implicit enumeration scheme enhanced by suitable pruning conditions (see Fischetti and Toth 1997). As for the detection of violated odd CAT inequalities, Balas (1989) showed that these inequalities correspond to odd cycles in an auxiliary undirected “incompatibility” graph. An effective heuristic separation algorithm, based on the computation of a minimum-weight odd cycle going through a given edge was proposed by Fischetti and Toth (1997). In addition, *clique lifting* (see Balas and Fischetti 1993) and *shrinking* (see Padberg and Rinaldi 1990b) procedures are applied to simplify the considered separation problems. A detailed analysis of the polyhedral structure of the ATSP can be found in Balas and Fischetti (1993).

Pricing is an important ingredient of branch-and-cut codes, since it allows one to effectively handle LP relaxations involving a huge number of variables. In order to keep the size of the LP relaxation as small as possible, the following pricing scheme is commonly used. We determine a (small) *core* set of arcs, say \tilde{A} , and decide to temporarily fix $x_{ij} = 0$ for all $(i, j) \in A \setminus \tilde{A}$. We then solve the corresponding restricted LP problem, compute the associated LP reduced costs \bar{c}_{ij} , and check whether $\bar{c}_{ij} \geq 0$ for all $(i, j) \in A \setminus \tilde{A}$. If this is the case, then the LP relaxation has been solved to optimality. Otherwise, the current core set \tilde{A} is enlarged by adding (some of) the arcs with negative reduced cost, and the whole procedure is iterated.

Fischetti and Toth (1997) proposed an improved pricing technique, called *AP pricing*, in which the pricing condition is strengthened by exploiting the fact that any feasible solution of the current LP relaxation cannot select the arcs with negative reduced cost in an arbitrary way, as the degree equations, among other constraints, have to be fulfilled.

The exact algorithm proposed by Fischetti and Toth (1997) is a lowest-first branch-and-cut method. At each node of the branching tree, the LP relaxation is initialized by taking all the constraints present in the last LP solved at the father node (for the root node, only the degree equations are taken). As for the variables, one retrieves from a scratch file the optimal basis associated with the last LP solved at the father node, and initializes the core variable set, \tilde{A} , by taking all the arcs belonging to this basis (for the root node, \tilde{A} contains the $2n - 1$ variables in the optimal AP basis found by solving AP on the original costs c_{ij}). In addition, \tilde{A} contains all the arcs of the best known ATSP solution. Starting with the above advanced basis, one iteratively solves the current LP relaxation, applies the AP pricing procedure and repeats if needed.

On exit of the pricing loop the separation algorithms are applied to find, if any, ATSP inequalities that cut off the current LP optimal solution x^* . When violated cuts are found, one adds them to the current LP relaxation and repeats.

When separation fails and x^* is integer, the current best ATSP solution is updated, and a backtracking step occurs. If x^* is fractional, instead, the current LP

basis is saved in a file, and one branches on the variable x_{ij} with $0 < x_{ij}^* < 1$ that maximizes the score $\sigma(i, j) = c_{ij} \min\{x_{ij}^*, 1 - x_{ij}^*\}$. As a heuristic rule, a large priority is given to the variables with $0.4 \leq x_{ij}^* \leq 0.6$ (if any), so as to produce a significant change in both descending nodes.

This branching scheme has been enhanced by Fischetti et al. (2003) through the so-called *Fractional Persistency* mechanism proposed by Fischetti et al. (2001) for the solution of crew scheduling and vehicle scheduling problems. The corresponding branch-and-cut algorithm will be denoted as FLT in “[Computational results](#)”.

Transformation of ATSP instances into STSP instances

It is easy to see that a code for the ATSP can be invoked to solve symmetric TSP instances. In fact, the reverse also stands by means of the following two transformations:

- the *3-node* transformation proposed by Karp (1972). A complete undirected graph with $3n$ vertices is obtained from the original complete directed one by adding two copies, $n + i$ and $2n + i$, of each vertex $i \in V$, and by (1) setting to 0 the cost of edges $(i, n + i)$ and $(n + i, 2n + i)$ for each $i \in V$, (2) setting to c_{ij} the cost of edge $(2n + i, j)$, $i, j \in V$, and (3) setting to $+\infty$ the costs of all remaining edges;
- the *2-node* transformation proposed by Jonker and Volgenant (1983) (see also Jünger et al. 1995). A complete undirected graph with $2n$ vertices is obtained from the original complete directed one by adding a copy, $n + i$, of each vertex $i \in V$, and by (1) setting to 0 the cost of the edge $(i, n + i)$ for each $i \in V$, (2) setting to $c_{ij} + M$ the cost of edge $(n + i, j)$, $i, j \in V$, where M is a sufficiently large positive value, and (3) setting to $+\infty$ the costs of all the remaining edges. The transformation value nM has to be subtracted from the STSP optimal cost.

The most effective branch-and-cut algorithm for the STSP is currently the one by Applegate et al. (2007), and the corresponding code, Applegate et al. (1999), is publicly available. In Fischetti et al. (2002), this code was used to test the effectiveness of the approach based on the ATSP-to-STSP transformation. The code has been used with default parameters. The results have shown that the 2-node transformation is, in general, more effective than the 3-node one.

Computational results

As a testbed for our computational experiments, we took the 27 ATSP instances collected in TSPLIB Reinelt (1991) and 5 real-world instances provided by Balas (2000). All the instances have integer nonnegative costs.

Tables 1 and 2 report on the computational performance of eight polynomial formulations (namely, MTZ, GG, CLAUS, DL, GP, SD, SSB, and SST) for the ATSP, described in “[Review of polynomial formulations](#)”, on ten small-size

Table 1 CPU times for solving the LP relaxations of 8 polynomial formulations using the primal, dual or barrier method

Formulation	Primal		Dual		Barrier	
	Solved	T_{LP}	Solved	T_{LP}	Solved	T_{LP}
MTZ	10	0.16	10	0.03	10	16.30
GG	10	0.25	10	0.56	10	0.14
CLAUS	3	1,018.11	10	353.84	10	95.05
DL	10	0.30	10	0.05	10	17.23
GP	10	48.01	10	3.48	10	232.78
SD	10	7.45	10	5.10	10	0.75
SSB	9	360.49	10	6.68	10	103.59
SST	0	1,200.00	1	1,128.43	3	1,004.75

instances from the TSPLIB (namely, instances *ftv33*, *ftv35*, *ftv38*, *ftv44*, *ftv47*, *ftv55*, *ftv64*, *ftv70*, *ft70*, and *ft53*), which is the set of instances used by Öncan et al. (2009) in their computational study. The computational results reported in Tables 1 and 2 were obtained on an Intel Core2 Duo@2.26 GHz by running CPLEX11.2 as LP and ILP solver. All the computing times reported in the following tables are expressed in seconds.

Even though none of the LP-bounds of the eight formulations considered in Tables 1 and 2 is theoretically stronger than the LP-bound of formulation EC-MCF, we do not report any computational results of EC-MCF because the corresponding memory requirements were such that we could not compute the LP-bound on any of the considered test instances. For the computational performance of formulation EC-MCF, the reader is referred to Godinho et al. (2011a, b).

In Table 1, we compare, on the ten considered instances, the performance of the eight polynomial formulations for computing the corresponding LP relaxations using three different LP methods (*Primal*, *Dual* and *Barrier*). For each method and each formulation, Table 1 reports the number of LP-bounds computed within the time limit of 1,200 s (column *Solved*) and the average computing time (column T_{LP}). In the computation of the average computing time, we considered a computing time of 1,200 s whenever the LP relaxation of an instance could not be computed within the time limit or the instance could not be run due to exceeded memory limits for CPLEX.

Table 1 shows that, for a given formulation, the computing times may significantly vary using different LP methods. The dual method turned out to be the best performing method when solving MTZ, DL, GP and SSB, whereas the barrier method was the best performing one when solving GG, CLAUS, SD and SST. All the formulations but SST could solve the LP relaxation of all the 10 instances within the time limit imposed.

Table 2 reports, for each of the eight formulations and each of the ten instances, the percentage gap (computed with respect to the optimal solution value) left after solving the LP relaxation (column LP), the percentage gap left at the root node of the search tree of CPLEX (column Rt), and the computing times for solving the LP

Table 2 Performance of 8 polynomial formulations on 10 ATSP instances from the TSPLIB

Inst	MTZ				GG				CLAUS				DL				
	T_{LP}	T_{LP}	T_{Rt}	T_{ILP}	T_{LP}	T_{LP}	T_{Rt}	T_{ILP}	T_{LP}	T_{LP}	T_{Rt}	T_{ILP}	T_{LP}	T_{LP}	T_{Rt}	T_{ILP}	
	LP	LP	Rt	T_{ILP}	LP	T_{LP}	Rt	T_{ILP}	LP	T_{LP}	Rt	T_{ILP}	LP	T_{LP}	Rt	T_{ILP}	
ftv33	7.64	0.02	2.77	0.22	4.91	7.03	0.06	0.78	0.83	5.51	0.00	7.20	0.00	9.44	0.00	9.44	14.88
ftv35	6.12	0.02	4.15	0.09	6.27	5.59	0.06	0.63	3.51	11.93	1.06	9.17	1.00	19.02	1.00	19.02	79.36
ftv38	5.87	0.02	2.91	0.14	9.02	5.44	0.08	0.80	2.29	21.46	1.02	12.50	0.96	31.01	0.96	31.01	74.88
ftv44	5.55	0.02	2.68	0.14	16.40	5.18	0.10	1.74	2.47	18.02	1.74	23.71	1.27	77.38	1.27	77.38	tl
ftv47	6.77	0.02	2.39	0.28	36.04	6.54	0.11	1.84	1.83	58.27	1.54	37.43	1.52	256.06	1.52	256.06	tl
ftv55	10.55	0.03	3.89	0.27	43.74	10.31	0.16	4.97	1.19	91.16	1.49	87.38	1.43	324.61	1.43	324.61	tl
ftv64	6.31	0.05	4.21	0.28	114.50	5.84	0.22	4.00	1.98	322.56	1.71	182.06	1.63	1,368.27	1.63	1,368.27	tl
ftv70	9.26	0.05	3.94	0.59	168.70	8.75	0.27	7.06	2.51	1,125.57	2.10	285.50	1.99	tl	1.99	tl	tl
ft70	1.77	0.06	1.16	0.64	tl	1.10	0.25	0.85	3.74	326.07	0.05	237.47	0.04	1,500.99	0.04	1,500.99	tl
ft53	14.04	0.05	10.88	0.44	312.38	12.45	0.14	0.00	3.65	37.95	0.00	68.13	0.00	255.02	0.00	255.02	265.67
Avg	7.39	0.03	3.90	0.31	251.20	6.82	0.14	2.27	2.40	201.85	1.07	95.05	0.98	564.18	0.98	564.18	1,123.48
					(9)					(10)							(4)

Inst	GP				SD				SSB				SST				
	T_{LP}	T_{LP}	T_{Rt}	T_{ILP}	T_{LP}	T_{LP}	T_{Rt}	T_{ILP}	T_{LP}	T_{LP}	T_{Rt}	T_{ILP}	T_{LP}	T_{LP}	T_{Rt}	T_{ILP}	
	LP	LP	Rt	T_{ILP}	LP	T_{LP}	Rt	T_{ILP}	LP	T_{LP}	Rt	T_{ILP}	LP	T_{LP}	Rt	T_{ILP}	
ftv33	0.00	0.85	0.00	1.97	8.21	4.78	0.25	4.53	0.61	26.91	0.00	0.41	0.00	0.71	0.77	0.77	0.00
ftv35	1.29	0.85	1.29	3.32	914.01	3.90	0.28	3.35	0.70	49.75	1.09	0.46	1.02	2.64	14.88	14.88	0.65
ftv38	1.24	1.32	1.24	5.38	tl	3.26	0.39	3.15	1.22	81.67	1.05	0.65	0.98	2.48	31.50	31.50	0.64
ftv44	1.84	1.58	1.26	8.24	tl	2.43	0.44	2.03	0.97	97.31	1.74	0.88	0.92	21.36	91.24	91.24	-
ftv47	1.86	1.63	1.53	6.27	tl	2.75	0.72	2.71	1.51	109.82	1.86	2.52	1.54	32.08	1,247.54	1,247.54	-
ftv55	2.33	3.85	1.12	83.12	tl	5.89	0.53	5.62	4.18	1,105.22	2.33	2.80	1.41	261.42	999.26	999.26	-
ftv64	3.30	6.64	2.99	189.60	tl	4.00	0.84	3.68	4.79	309.85	2.88	14.92	2.70	691.29	tl	tl	-
ftv70	3.73	8.11	3.37	53.06	tl	4.64	2.22	4.45	6.99	525.08	3.12	27.54	2.63	476.80	tl	tl	-
ft70	1.15	7.24	1.09	44.94	tl	0.80	1.36	0.74	7.46	236.81	0.55	14.66	0.48	38.21	tl	tl	-
ft53	10.69	2.75	9.33	16.73	tl	11.39	0.48	11.26	1.47	tl	10.63	1.94	9.22	8.18	tl	tl	-
Avg	2.74	3.48	2.32	41.26	1,532.22	4.38	0.75	4.15	2.99	434.24	2.53	6.68	2.09	153.51	958.52	958.52	0.43
					(2)					(9)							(6)
																	(1)

Table 3 Comparison between branch-and-bound algorithms, branch-and-cut algorithms and polynomial formulations

Inst	CDT		FT		FLT		Concorde		MTZ		GG		DL	
	LB	T	LB	T	LB	T	LB	T	LB	T	LB	T	LB	T
br17	100.00	3.6	0.00	0.0	0.00	0.0	0.00	0.2	94.23	10.9	68.91	8.2	43.59	34.6
ft53	14.11	tl	1.56	0.2	0.00	0.1	0.00	0.6	14.04	3,123.8	12.45	379.5	12.93	tl
ft70	1.80	0.4	0.57	0.3	0.02	0.2	0.01	3.2	1.77	tl	1.10	3,260.7	0.88	652.2
ftv33	7.85	0.0	3.73	0.1	0.00	0.0	0.00	0.3	7.64	49.1	7.03	55.1	5.35	34.0
ftv35	6.25	0.0	3.53	0.2	0.88	0.4	0.68	9.0	6.12	62.7	5.59	119.3	4.04	52.3
ftv38	6.01	0.0	3.01	0.3	0.85	0.6	0.52	14.5	5.87	90.2	5.44	214.6	3.45	57.1
ftv44	5.70	0.0	4.46	0.1	0.37	0.5	0.12	9.1	5.55	164.0	5.18	180.2	2.43	31.7
ftv47	6.98	0.1	3.49	0.4	1.01	0.5	0.62	23.4	6.77	360.4	6.54	582.7	2.83	120.0
ftv55	10.76	1.1	6.59	1.5	0.81	1.4	0.44	9.0	10.55	437.4	10.31	911.6	6.05	194.8
ftv64	6.42	0.8	4.89	1.3	1.36	2.6	0.33	20.8	6.31	1,145.0	5.84	3,225.6	4.24	325.3
ftv70	9.44	3.3	6.92	3.7	0.92	1.1	0.26	17.8	9.26	1,687.0	8.75	11,255.7	4.69	524.0
ftv90	6.33	0.7	3.17	2.5	0.25	0.5	0.06	14.7	6.24	340.4	6.21	tl	1.72	498.1
ftv100	6.60	16.6	3.91	29.6	0.39	2.2	0.00	12.6	6.53	1,083.1	6.49	tl	3.11	1,237.5
ftv110	5.87	5.0	4.49	38.4	0.77	7.4	0.05	25.6	5.81	5,964.9	5.71	tl	2.41	tl
ftv120	6.51	50.1	5.77	99.4	0.97	13.1	0.28	54.4	6.44	8,755.0	6.32	tl	2.18	tl
ftv130	4.46	6.4	3.77	16.6	0.35	1.6	0.00	16.6	4.42	1,938.2	4.34	tl	1.31	600.7
ftv140	4.92	13.9	4.26	38.8	0.25	2.1	0.00	25.6	4.87	1,127.6	4.85	tl	1.41	287.3
ftv150	3.91	3.1	3.03	17.0	0.27	2.6	0.00	27.0	3.87	920.6	3.86	tl	1.31	402.7
ftv160	4.58	93.8	4.03	316.2	0.67	3.8	0.30	55.7	4.55	12,731.4	4.54	tl	2.35	6,000.6
ftv170	4.50	tl	4.14	tl	0.87	4.1	0.40	41.9	4.48	tl	4.47	tl	2.05	tl
kro124p	6.22	tl	2.73	135.7	0.04	1.0	0.00	9.9	6.13	2,296.6	5.47	tl	3.46	2,316.8
p43	97.37	tl	0.37	tl	0.16	9.3	0.16	22.7	97.34	tl	85.23	tl	96.16	tl
rfg323	0.00	0.1	0.00	0.3	0.00	0.4	0.00	23.9	0.00	tl	0.00	tl	0.00	8,481.1

Table 3 continued

Inst	CDT		FT		FLT		Concorde		MTZ		GG		DL		
	LB	T	LB	T	LB	T	LB	T	LB	T	LB	T	LB	T	
rbg358	0.00	0.1	0.00	0.5	0.00	0.5	0.00	29.3	0.00	0.00	0.00	0.00	0.00	0.00	tl
rbg403	0.00	0.1	0.00	1.0	0.00	1.3	0.00	49.3	0.00	0.00	0.00	0.00	0.00	0.00	tl
rbg443	0.00	0.1	0.00	1.2	0.00	1.4	0.00	34.5	0.00	0.00	0.00	0.00	0.00	0.00	tl
ry48p	13.21	tl	2.94	20.3	0.53	0.8	0.35	22.9	12.88	1,316.0	11.17	11.17	4.25	3,417.3	tl
balas84	14.07	tl	5.53	986.6	1.01	15.7	1.01	78.0	13.98	tl	12.34	12.34	9.67	tl	tl
balas108	25.00	tl	9.87	tl	1.97	89.0	2.63	1,416.0	24.80	tl	18.14	18.14	19.50	tl	tl
balas120	21.68	tl	13.29	tl	1.05	1,276.3	1.05	7,186.9	21.43	tl	18.58	18.58	16.86	tl	tl
balas160	19.40	tl	11.34	tl	1.26	671.1	1.26	7,848.0	19.18	tl	16.16	16.16	18.83	tl	tl
balas200	15.63	tl	8.68	tl	1.24	1,712.8	0.74	2,294.2	15.55	tl	12.86	12.86	15.31	tl	tl
Avg	13.61	9.06	4.06	65.85	0.57	119.51	0.35	606.18	13.33	2,180.21	11.37	11.37	9.14	1,329.90	(19)
		(22)		(26)		(32)		(32)		(20)				(11)	
Avg	-	318.7	-	241.0	-	88.6	-	145.4	-	794.9	-	-	-	826.6	681.7
1,000 sec		(22)		(26)		(30)		(28)		(9)				(8)	(14)

relaxation (column T_{LP}), the root node (column T_{Rt}), and the ILP formulation itself (column T_{ILP}). For each formulation, we used the LP-method that turned out to be the best one in the experiments summarized in Table 1. The time limit for solving the ILP formulation was set equal to 1,800 s. In the rows labeled “Avg”, Table 2 reports average values and, in parenthesis, the number of instances solved to optimality within the time limit. In the computation of the average computing times, we considered a computing time equal to 1,200 or 1,800 s whenever the LP model or the ILP model, respectively, could not be solved within the time limit. For formulation SST the averages refer to the first three instances since the remaining ones could not be run due to exceeded memory limits for CPLEX.

As to the LP-bounds, Table 2 shows that SST obtains tight lower bounds but requires larger computing times than the other formulations. Moreover, SST was able to provide the lower bounds, within the time limit, on only three of the ten instances considered. Among the other formulations, the one that provides the tightest LP-bounds is CLAUS, followed by SSB and GP. The other four formulations (MTZ, GG, DL and SD) obtain, on average, worse LP-bounds than the previous ones but within shorter computing time (less than one second on average). GG was the only formulation able to solve, to optimality, all of the ten instances within the time limit imposed, whereas MTZ, DL and SD could solve, to optimality, all but one of the instances.

The results reported in Table 2 show that formulations MTZ, GG, and DL are the best formulations to be directly used within CPLEX. This is probably due to two main reasons: (1) the limited number of constraints required to break subtours allows CPLEX to compute the LP-bounds effectively, and (2) the large variety of cuts embedded in CPLEX lead to root lower bounds competitive with those that can be obtained with other formulations having stronger LP-bounds but within much shorter computing times. We stress that the computational analysis reported in Table 2 is aimed at comparing the suitability of each formulation to be directly solved with CPLEX. Therefore, any conclusion that can be drawn from the results reported in Table 2 cannot consider the fitness of each polynomial formulation as the starting point for more complicated exact algorithms for solving the ATSP. We also stress that, for each class of formulations, we report only the results obtained by the formulation providing, on average, the tightest LP-bounds because the other formulations (i.e., those appearing in Fig. 1 but neglected in this section) were computationally outperformed by at least one of the eight formulations considered in this study.

In Table 3, we compare, on the 27 ATSP instances from TSPLIB and the 5 instances provided by Balas, the performance of (a) the branch-and-bound algorithm by Carpaneto et al. (1995) (hereafter CDT) described in “Exact algorithms”, (b) the branch-and-bound algorithm based on the additive bounding procedure by Fischetti and Toth (1992) (hereafter FT) described in “Exact algorithms”, (c) the branch-and-cut algorithm by Fischetti et al. (2003) (hereafter FLT) described in “Exact algorithms”, (d) the branch-and-cut algorithm (hereafter Concorde) solving the STSP instances obtained from the transformation described in “Transformation of ATSP instances into STSP instances”, and (e) the polynomial formulations MTZ, GG and DL (see “Review of polynomial formulations”). For each instance and each

algorithm, we report the percentage gap of the corresponding lower bound, or the value of the LP relaxation of the considered formulation, computed at the root node of the decision tree (column LB) and the computing time for solving the instance to optimality (column T). The rows labeled “Avg” report average values and, in parenthesis, the number of instances solved to optimality within the corresponding time limit. Different time limits were imposed on the algorithms: 1,000 s for CDT and FT, 10,000 s for FLT and Concorde, 1,800 s for MTZ, GG and DL. For this reason, in the computation of the average computing time, only the instances solved to optimality within the imposed time limit are considered. The rows labeled “Avg 1,000 s” report the average computing time and, in parenthesis, the number of instances solved to optimality within the time limit of 1,000 s (a computing time equal to 1,000 s is considered when this time limit is reached).

The computational results reported for CDT, FT, FLT and Concorde are taken from Fischetti et al. (2002) and were obtained on a Digital Alpha 533 MHz with CPLEX6.5.3 as LP solver, whereas the computational results relative to formulations MTZ, GG and DL were obtained on an Intel Core2 Duo@2.26 GHz by running CPLEX11.2. We have experimentally found that the latter machine is approximately 10-12 times faster than that used by Fischetti et al. (2002); therefore, the computing times reported for MTZ, GG and DL were multiplied by 10 in order to have a fair comparison.

Table 3 shows that FLT and Concorde were the only two exact methods able to solve, within the imposed time limit, all the 32 instances to optimality, and are clearly better performing than the other 5 exact methods considered. Although, on the considered instances, Concorde generally obtains better lower bounds at the root node, FLT is always faster. MTZ, GG and DL proved not to be competitive with either FLT or Concorde, since they solved only 20, 11 and 19 instances, respectively, and, on the solved instances, their computing times are always much longer. As for the branch-and-bound algorithms CDT and FT, for which a time limit of 1,000 s was imposed, it can be noted that they dominate, by considering the instances solved to optimality within this time limit and the computing times, the polynomial formulations MTZ, GG and DL. By considering the values of the lower bounds at the root node, Table 3 shows that (a) the lower bound of CDT (given by the value of the AP relaxation) is only slightly worse than that of MTZ (i.e., the addition of constraints (10) to the AP relaxation only marginally improves the corresponding LP relaxation), (b) the lower bound of FT (corresponding to the additive bounding procedure) is always better than those of MTZ and GG, and globally better than that of DL.

Acknowledgments Work supported by Ministero dell’Istruzione, dell’Università e della Ricerca (M.I.U.R.), Italy. Thanks are due to two referees for making several suggestions that improved the presentation of the paper.

References

Applegate D, Bixby R, Chvátal V, Cook W (1999) Concorde—a code for solving traveling salesman problems. <http://www.tsp.gatech.edu/concorde/index.html>

- Applegate D, Bixby R, Chvátal V, Cook W (2007) The traveling salesman problem: a computational study. Princeton series in applied mathematics. Princeton University Press, Princeton
- Balas E (1989) The asymmetric assignment problem and some new facets of the traveling salesman polytope on a directed graph. *SIAM J Discret Math* 2(4):425–451
- Balas E (2000) Personal communication
- Balas E, Christofides N (1981) A restricted Lagrangian approach to the traveling salesman problem. *Math Programm Ser A* 21(1):19–46
- Balas E, Fischetti M (1993) A lifting procedure for the asymmetric traveling salesman polytope and a large new class of facets. *Math Programm Ser A* 58(1-3):325–352
- Balas E, Toth P (1985) Branch and bound methods. In: Lawler E, Lenstra J, Rinnooy Kan A, Shmoys D (eds) *The Traveling Salesman Problem: a guided tour of combinatorial optimization*. Wiley, Chichester, pp 361–401
- Bellmore M, Malone J (1971) Pathology of traveling-salesman subtour-elimination algorithms. *Oper Res* 19(2):278–307
- Carpaneto G, Toth P (1980) Some new branching and bounding criteria for the asymmetric travelling salesman problem. *Manage Sci* 26(7):736–743
- Carpaneto G, Toth P (1987) Primal-dual algorithms for the assignment problem. *Discret Appl Math* 18(2):137–153
- Carpaneto G, Dell’Amico M, Toth P (1995) Exact solution of large-scale asymmetric traveling salesman problems. *ACM Transact Math Softw* 21(4):394–409
- Claus A (1984) A new formulation for the travelling salesman problem. *SIAM J Algebraic Discret Methods* 5(1):21–25
- D’Ambrosio C, Lodi A, Martello S (2010) Combinatorial traveling salesman problem algorithms. In: *Wiley encyclopedia of operations research and management science*, vol 1. Wiley, New York, pp 738–747
- Dantzig G, Fulkerson D, Johnson S (1954) Solutions of a large-scale traveling-salesman problem. *Oper Res* 2(4):393–410
- Desrochers M, Laporte G (1990) Improvements and extensions to the Miller–Tucker–Zemlin subtour elimination constraints. *Oper Res Lett* 10(1):27–36
- Edmonds J (1967) Optimum branchings. *J Res Natl Bur Stand Sec B* 71B(4):233–240
- Finke G, Claus A, Gunn E (1984) A two commodity network flow approach to the travelling salesman problem. *Congressus Numerantium* 41:167–178
- Fischetti M (1991) Facets of the asymmetric traveling salesman polytope. *Math Oper Res* 16(1):42–56
- Fischetti M, Toth P (1989) An additive bounding procedure for combinatorial optimization problems. *Oper Res* 37(2):319–328
- Fischetti M, Toth P (1992) An additive bounding procedure for the asymmetric travelling salesman problem. *Math Program Ser A* 53(1-3):173–197
- Fischetti M, Toth P (1993) An efficient algorithm for the min-sum arborescence problem. *ORSA J Comput* 5:426–434
- Fischetti M, Toth P (1997) A polyhedral approach to the asymmetric traveling salesman problem. *Manage Sci* 43(11):1520–1536
- Fischetti M, Lodi A, Martello S, Toth P (2001) A polyhedral approach to simplified crew scheduling and vehicle scheduling problems. *Manage Sci* 47(6):833–850
- Fischetti M, Lodi A, Toth P (2002) Exact methods for the asymmetric traveling salesman problem. In: Gutin G, Punnen A (eds) *The traveling salesman problem and its variations*. Kluwer Academic Publishers, The Netherlands, pp 169–205
- Fischetti M, Lodi A, Toth P (2003) Solving real-world ATSP instances by branch-and-cut. *Lecture Notes in Computer Science 2570 Combinatorial Optimization- Eureka, you shrink!* Springer, Berlin, pp 64–77
- Fox K, Gavish B, Graves S (1980) An n-constraint formulation of the (time-dependent) traveling salesman problem. *Oper Res* 28(4):1018–1021
- Garfinkel R (1973) On partitioning the feasible set in a branch-and-bound algorithm for the asymmetric traveling-salesman problem. *Oper Res* 21(1):340–343
- Gavish B, Graves S (1978) The travelling salesman problem and related problems, working Paper GR-078-78. Operations Research Center, Massachusetts Institute of Technology, Cambridge
- Godinho MT, Gouveia L, Pesneau P (2011a) Natural and extended formulations for the time-dependent traveling salesman problem. *Discret Appl Math* doi:[10.1016/j.dam.2011.11.019](https://doi.org/10.1016/j.dam.2011.11.019)

- Godinho MT, Gouveia L, Pesneau P (2011b) On a time-dependent flow-based formulation and an updated classification of formulations for the travelling salesman problem. In: Progress in combinatorial optimization, chap 7. Wiley, New York, pp 223–254
- Gouveia L (1995) A result on projection for the vehicle routing problem. *Eur J Oper Res* 85(3):610–624
- Gouveia L, Pesneau P (2006) On extended formulations for the precedence constrained asymmetric traveling salesman problem. *Networks* 48(2):77–89
- Gouveia L, Pires J (1999) The asymmetric travelling salesman problem and a reformulation of the Miller–Tucker–Zemlin constraints. *Eur J Oper Res* 112:134–146
- Gouveia L, Pires J (2001) The asymmetric travelling salesman problem: on generalizations of disaggregated Miller–Tucker–Zemlin constraints. *Discret Appl Math* 112(1–3):129–145
- Gouveia L, Voss S (1995) A classification of formulations for the (time-dependent) travelling salesman problem. *Eur J Oper Res* 83(1):69–82
- Grötschel M, Padberg M (1985) Polyhedral theory. In: Lawler E, Lenstra J, Rinnooy Kan A, Shmoys D (eds) *The Traveling Salesman Problem: A guided tour of combinatorial optimization*. Wiley, Chichester, pp 251–305
- Gutin G, Punnen A (2002) *The traveling salesman problem and its variations*. Kluwer, Dordrecht
- Held M, Karp R (1970) The traveling-salesman problem and minimum spanning trees. *Oper Res* 18(6):1138–1162
- Held M, Karp R (1971) The traveling-salesman problem and minimum spanning trees: part ii. *Math Program Ser A* 1(1):6–25
- Jonker R, Volgenant T (1983) Transforming asymmetric into symmetric traveling salesman problems. *Oper Res Lett* 2(4):161–163
- Jünger M, Reinelt G, Rinaldi G (1995) The traveling salesman problem. In: Ball M, Magnanti T, Monma C, Nemhauser G (eds) *Network models, handbooks in operations research and management science*, vol 7. North Holland, Amsterdam, pp 255–330
- Karp R (1972) Reducibility among combinatorial optimization problems. In: Miller R, Thatcher J (eds) *Complexity of Computer Computations*. Plenum Press, New York, pp 85–103
- Karp R (1979) A patching algorithm for the nonsymmetric traveling-salesman problem. *SIAM J Comput* 8(4):561–573
- Langevin A (1988) *Planification des tournées de véhicules*. PhD thesis, École Polytechnique de Montréal, Montreal
- Langevin A, Soumis F, Desrosiers J (1990) Classification of travelling salesman problem formulations. *Oper Res Lett* 9(2):127–132
- Lawler E (1976) *Combinatorial optimization: networks and matroids*. Holt, Rinehart and Winston, New York
- Lawler E, Lenstra J, Rinnooy Kan A, Shmoys D (1985) *The Traveling Salesman Problem: A guided tour of combinatorial optimization*. Wiley, Chichester
- Little J, Murty K, Sweeney D, Karel C (1963) An algorithm for the traveling salesman problem. *Oper Res* 11(6):972–989
- Loulou R (1988) On commodity flow formulations for the TSP, working paper. McGill University, Montréal
- Miller C, Tucker A, Zemlin R (1960) Integer programming formulation of traveling salesman problems. *J Ass Comput Mach* 7(4):326–329
- Miller D, Pekny J (1989) Results from a parallel branch and bound algorithm for the asymmetric traveling salesman problem. *Oper Res Lett* 8(3):129–135
- Öncan T, Kuban Altinel I, Laporte G (2009) A comparative analysis of several asymmetric traveling salesman problem formulations. *Comput Oper Res* 36(3):637–654
- Padberg M, Rinaldi G (1990a) An efficient algorithm for the minimum capacity cut problem. *Math Program Ser A* 47(1–3):19–36
- Padberg M, Rinaldi G (1990b) Facet identification for the symmetric traveling salesman polytope. *Math Program Ser A* 47(1–3):219–257
- Padberg M, Sung T (1991) An analytical comparison of different formulations of the travelling salesman problem. *Math Program Ser A* 52(1–3):315–357
- Pekny J, Miller D (1992) A parallel branch and bound algorithm for solving large asymmetric traveling salesman problems. *Math Program Ser A* 55(1–3):17–33
- Pekny J, Miller D, Stodolsky D (1991) A note on exploiting the hamiltonian cycle problem substructure of the asymmetric traveling salesman problem. *Oper Res Lett* 10(3):173–176

- Reinelt G (1991) TSPLIB - a traveling salesman problem library. *ORSA J Comput* 3(4):376–384. <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>
- Reinelt G (1994) The traveling salesman problem: computational solutions for TSP applications. Lecture Notes in Computer Science. Springer, Berlin
- Sarin S, Sherali H, Bhootra A (2005) New tighter polynomial length formulations for the asymmetric travelling salesman problem with and without precedence constraints. *Oper Res Lett* 33(1):62–70
- Sherali H, Driscoll P (2002) On tightening the relaxations of Miller–Tucker–Zemlin formulations for asymmetric traveling salesman problems. *Operat Res* 50(6):656–669
- Sherali H, Sarin S, Tsai PF (2006) A class of lifted path and flow-based formulations for the asymmetric travelling salesman problem with and without precedence constraints. *Discret Optim* 3(1):20–32
- Smith T, Srinivasan V, Thompson G (1977) Computational performance of three subtour elimination algorithms for solving asymmetric traveling salesman problems. In: PL Hammer BK EL Johnson, Nemhauser G (eds) *Studies in Integer Programming*, Annals of Discrete Mathematics, vol 1. Elsevier, Amsterdam, pp 495–506
- Tarjan R (1977) Finding optimum branchings. *Networks* 7(1):25–35
- Wong R (1980) Integer programming formulations of the traveling salesman problem. In: *Proceedings of the IEEE international conference of circuits and computers*, pp 149–152