The Steiner Traveling Salesman Problem and Its Extensions

Jessica Rodríguez-Pereira[1], Elena Fernández[2], Gilbert Laporte[1], Enrique Benavent[3], and Antonio Martínez-Sykora[4]

[1]Canada Research Chair in Distribution Management, HEC Montréal.
jessica.rodriguez-pereira@hec.ca, gilbert.laporte@cirrelt.ca
[2]Department of Statistics and Operation Research, Universidad de Cádiz, Spain.
elena.fernandez@uca.es
[3]Department of Statistics and Operation Research, Universitat de Valencia, Spain.
enrique.benavent@uv.es
[4]Southampton Business School, University of Southampton, United Kingdom.
A.Martinez-Sykora@soton.ac.uk

April 29, 2019

Abstract

This paper considers the Steiner Traveling Salesman Problem, an extension of the classical Traveling Salesman Problem on an incomplete graph where not all vertices have demand. Some extensions including several depots or location decisions are introduced, modeled and solved. A compact integer linear programming formulation is proposed for each problem, where the routes are represented with two-index decision variables, and parity conditions are modeled using cocircuit inequalities. Exact branch-and-cut algorithms are developed for all formulations. Computational results obtained confirm the good performance of the algorithms. Instances with up to 500 vertices are solved optimally.

Keywords: Steiner Traveling Salesman Problem; integer linear programming; branch-and-cut; valid inequalities.

1 Introduction

The purpose of this paper is to develop effective formulations and efficient exact solution algorithms for the Steiner Traveling Salesman Problem (STSP) and some of its extensions on an undirected network. The STSP is an uncapacitated node-routing problem looking for a minimum-cost route that visits a known set of users with service demand, located at vertices of a given network. The STSP extends the classical TSP in two respects. On the one hand, it is not assumed that the underlying network is complete. Hence, an edge between every pair of vertices may not necessarily exist. On the other hand, some vertices in the network may not have a demand. Still, some non-demand vertices may have to be visited in order to connect two demand vertices served consecutively in a route. This means that the set of vertices that must be traversed in a feasible route is not known in advance.

The relevance of the STSP is not only theoretical but also practical, since it better suits real-life applications in which the underlying network is not complete and where, in addition, not all vertices of the network have a demand. However, additional difficulties arise when studying the STSP, which have to be addressed from an alternative perspective. In particular, the fact that the input graph may not be complete has a direct effect on the degree of the visited vertices in the solutions, which, in contrast to the TSP, is no longer two and cannot be foreseen. This difficulty may in fact be overcome, since it is theoretically possible to operate on a network induced by the demand vertices, in which the edge connecting a pair of vertices represents a minimum-cost path between its end-vertices in the original graph. This operation may, however, not be efficient given that the increase in the number of edges in the resulting network translates directly into the number of decision variables required in mathematical programming formulations. For this reason, in this paper we work on the original incomplete network where not all vertices must have demand.

The STSP was introduced in the same year by two different groups of researchers [10, 15], although it was called the Road Traveling Salesman Problem in [15]. A particular case in which all vertices in the network have to be visited was previously studied by [17, 22]. This particular case is known as the the Graphical Traveling Salesman Problem. Quite surprisingly, the STSP has received little attention after the mid-1980s. To the best of our knowledge Letchford, Nasiri and Theis [20] were the first to propose alternative formulations to the classical formulation from [15]. These authors adapted several well-known formulations of the TSP to the STSP. In particular, they introduced two variants of single-commodity flow model, a multi-commodity flow model and two variants of time-staged formulations for the STSP defined on a directed graph. They proved that the linear programming (LP) relaxation lower bound of their multi-commodity flow formulation coincides with the LP bound of the formulation of [15]. Using their most effective formulations the authors were able to optimally solve STSP instances with up to 250, vertices although the computational burden, both in terms of computing times and memory requirements becomes highly demanding as the size of the instances increases. The STSP is a particular case of the General Routing Problem (GRP), which looks for a minimum-cost route that visits a known set of users with service demand, located at vertices or edges of a given network. The GRP was introduced by Orloff in [23] and its relation to the STSP was studied in [9, 18, 19]. Other related works include [8, 25].

Here we propose a compact integer linear programming (ILP) formulation for the STSP defined on the original undirected graph with a small number of two-index decision variables. Our formulation exploits the well-known property that there exists an optimal STSP solution in which no edge is traversed more than twice; it uses two sets of binary variables, associated with the first and second traversal of edges, respectively. The feasibility of solutions is modeled with two families of constraints of exponential size, one for the connectivity with the depot, and another one for the parity of the visited vertices. For the parity of the vertices we use an adaptation to the STSP of cocircuit constraints [4], which exploits the relationship between our two sets of decision variables. This makes it unnecessary to define additional general integer variables associated with the vertices in order to linearize their parity constraints, as it is usual practice both for node and arc routing problems (see, e.g, [15] for the STSP and [7] for the undirected Rural Postman Problem). While cocircuit inequalities are now often used to model parity in Arc Routing Problems (ARPs), see, e.g. Ghiani and Laporte [16], they are still rarely used to model the parity of the visited vertices in node-routing problem, see, e.g. Belenguer et al. [6] . We prove that the LP bound of the proposed formulation coincides with that of the formulation of [15], which, as mentioned, is the same as the LP bound of the MCF formulation of [20].

As we will see, the performance of the branch-and-cut algorithm based on the proposed formulation is excellent. Since the separation of the two families of constraints with exponential size can be carried out efficiently, we have developed an exact branch-and-cut algorithm for the STSP which allow us to optimally solve instances with up to 500 vertices within modest computing times. Our computing times never exceed 350 seconds for instances with up to 250 vertices and, except for two out of 60 instances, do not exceed our time limit of 7,200 seconds for the larger instances with a number of vertices in the 275–500 range. These results substantially improve those of [20], where the largest instances solved have 250 vertices and the computational burden required is substantially higher.

In the second part of this work we focus on two extensions of the STSP, namely the Steiner Multi-depot Traveling Salesman Problem (SMDTSP) and the Steiner Location Traveling Problem (SLTSP). In particular, we consider different modeling hypotheses for SMDTSPs and we investigate alternative problems that differ from each other in (i) whether or not all the depots must be used and (ii) whether or not each route must traverse a single depot. These models arise naturally as theorical extensions of the STSP, which, in addition, may reflect better potential applications. On the one hand, similarly to other routing problems, when considering large-scale instances, there are usually several depots from which customers demand can be satisfied. On the other hand, the selection of the locations for the depots is a strategic decision, which has proven to have a relevant impact on the routing decisions.

We propose ILP formulations for all the introduced problems, in the spirit of those proposed in [12, 13, 14] for multi-depot ARPs and location-routing ARPs, making use of two sets of two-index decision variables. In all cases one of the sets consists of binary variables. While the variables of the second set are also binary for one of the SMDTSP models and the SLTSP, these variables are general integer for the other SMDTSPs.

Similarly to the STSP, we have developed efficient exact branch-and-cut algorithms to solve the formulations for the SMDTSP variants and the SLTSP. As expected, the computational effort required to solve the instances is now considerably larger than for the STSP. This is not surprising given that both types problems involve additional decisions to the STSP, particularly in the case of the SLTSP where the decision on the locations of the depots is integrated within the model. However, SMDTSP instances with up to 500 vertices and seven depots were solved optimally within the maximum allowed computing times of 7,200 seconds. As for the SLTSP, instances with up to 500 vertices and 10 potential locations were optimally solved within the prescribed time.

The remainder of the paper is organized as follows. Section 2 focuses on the STSP. After introducing some notation, we give a formal definition of the STSP, before presenting our ILP formulation. We then prove that its LP bound coincides with that of [15] and we present some families of valid inequalities that can be used to reinforce its LP relaxation. In Sections 2.3 and 2.4 we present the exact branch-and-cut algorithm, which we use to solve our STSP formulation, and we provide the numerical results of the computational experiments we have carried out. The SMDTSP is introduced in Section 3, where the alternative models are analyzed and compared. Section 4 focuses on the SLTSP. The computational experiments with the solution algorithms for the SMDTSPs and the SLTSP are described in Section 5, where numerical results are also presented and analyzed. The paper closes with some conclusions in Section 6.

2 The Steiner Traveling Salesman Problem

The STSP is defined on an undirected connected graph $G = (V, E)$, not necessarily complete, where V is the vertex set, $|V| = n$, with a distinguished vertex $d \in V$ representing the depot, and E is the edge set, with $|E| = m$. There is a set of vertices with service demand, $V_R \subseteq V$, also referred to as the set of *customers* or set of *required vertices*. With each edge $e \in E$ is associated a non-negative real cost c_e. Feasible solutions to the STSP are tours, starting and ending at the depot and visiting each customer at least once. We say that a customer is *served* by a tour if the tour visits it at least once. The cost of a tour is the sum of the costs of its edges, where the cost of each edge is counted as many times as it is traversed. The STSP consists of finding a minimum cost feasible tour.

In the following we use the notation $\overline{V}_R = V_R \cup \{d\}$ to denote the set of vertices that must be visited. This includes the depot plus the set of required vertices. We will use the following usual notation. For any non-empty vertex subset $S \subset V$, $\delta(S) = \{(u,v) \in E | u \in S, v \in V \setminus S\} = \delta(V \setminus S)$ is the set edges in the cut between S and $V \backslash S$ and $\gamma(S) = \{(u,v) \in E | u,v \in S\}$ the set of edges with both vertices in S. For a singleton $S = \{v\}$, with $v \in V$, we simply write $\delta(v)$ instead of $\delta(\{v\})$. For $H \subset E$ we use $\delta_H(S) = \delta(S) \cap H$ and $\gamma_H(S) = \gamma(S) \cap H$. We also use the standard compact notation $f(A) \equiv \sum_{e \in A} f_e$ where f is a vector defined over a set Ω and $A \subseteq \Omega$. Thus, if x is a vector defined on the edge set E and $H \subseteq E$, then $x(H) = \sum_{e \in H} x_e$. Also if x and y are vectors defined on E and $H \subseteq E$, then $(x + y)(H) = \sum_{e \in H}(x_e + y_e)$.

Proposition 2.1 *In any optimal solution to a given STSP instance all the edges used belong to some minimum cost path in G connecting two vertices of $\overline{V}_R$.*

Proof: Suppose that for a given STSP instance there exists an optimal tour T^* that uses an edge $e = (u, v)$, which does not belong to any minimum-cost path of G between two vertices of $\overline{V}_R$. Let $\overline{u}, \overline{v} \in V_R$ respectively denote the required vertices that are the *neighbors* of u and v in T^*, meaning that $P_{\overline{u},\overline{v}}$, the subpath of T^* connecting $\overline{u}$ and $\overline{v}$, contains no required vertex other than $\overline{u}$ and $\overline{v}$. Let also $\overline{P}_{\overline{u},\overline{v}}$ denote a minimum-cost path connecting $\overline{u}$ and $\overline{v}$. Since $e = (u, v)$ does not belong to any minimum-cost path, $c(\overline{P}_{\overline{u},\overline{v}}) < c(P_{\overline{u},\overline{v}})$. Furthermore, the tour $\widehat{T} = (T \setminus P_{\overline{u},\overline{v}}) \cup \overline{P}_{\overline{u},\overline{v}}$ is also feasible, and $c(\widehat{T}) = c(T^*) - c(P_{\overline{u},\overline{v}}) + c(\overline{P}_{\overline{u},\overline{v}}) < c(T^*)$, contradicting the optimality of T^*. ■

Proposition 2.1 allows us to reduce the graph G by removing the edges not belonging to any minimum cost path between pairs of customers, or between customers and the depot. Hence, in the remainder of this paper we assume that G has been simplified so that set E contains the edges that belong to some minimum path.

As noted in [10, 15] there exists an optimal solution to an STSP in which no edge is traversed more than twice. Otherwise, if T is a feasible tour where some edge e is used more than twice, then the tour that results by removing from T two copies of e is also feasible and its cost is at most equal to that of tour T. We exploit this optimality condition to obtain an ILP formulation for the STSP with two sets of binary variables only, associated with the first and second traversals of edges, respectively. In particular, for every $e \in E$, let x_e be a binary variable indicating whether or not edge e is traversed and let y_e be a binary variable equal to one if and only if edge e is traversed twice. The two-index formulation is the following:

$$(2IF) \quad \text{minimize} \quad \sum_{e \in E} c_e(x_e + y_e) \tag{1}$$

subject to

$$(x+y)(\delta(S)) \geq 2 \qquad S \subseteq V \setminus \{d\},\ S \cap V_R \neq \emptyset \tag{2}$$

$$(x-y)(\delta(u) \setminus H) + y(H) \geq x(H) - |H| + 1 \qquad u \in V,\ H \subseteq \delta(u),\ |H| \text{ odd} \tag{3}$$

$$y_e \leq x_e \qquad e \in E \tag{4}$$

$$x_e \in \{0,1\} \qquad e \in E \tag{5}$$

$$y_e \in \{0,1\} \qquad e \in E. \tag{6}$$

Constraints (2) are the well-known connectivity inequalities, which impose that at least two edges must cross the cut-set of any given vertex set containing some customer, but not containing the depot. In particular, when $S = \{u\}$ with $u \in V_R$, they ensure all customers are visited at least once. Moreover, when $S = V \setminus \{d\}$ also ensure that the depot is used. Constraints (3) are an adaptation of cocircuit inequalities [4] which ensure the parity of every vertex. These constraints impose that if an odd number of edges in the cut-set of a vertex is used, then at least one additional traversal must be made. This additional traversal corresponds either to a second traversal of some edge of H, or to a first traversal of some edge in $\delta(S) \setminus H$. Furthermore, taking into account the definition of the decision variables, the additional traversal must either be a second traversal of some of the selected edges or a first traversal of some of the non-selected edges in the cut-set. Constraints (4) impose that an edge cannot be traversed a second time unless it has been traversed a first time. The binary conditions of the variables x and y derived from their definition are imposed through constraints (5) and (6).

This formulation contains $2|E|$ variables, namely $|E|$ of each type x and y. There are $|E|$ constraints (4). The size of the families constraints (2) and (3) is exponential in $|V|$.

If a different preprocess were applied to G to eliminate non-demand vertices by adding shortest-path edges, dominance results similar to those of [16, 25] could be applied, restricting edges that can be used twice in an optimal solution to those of a minimum spanning tree on the transformed graph. As opposed to our transformation based on Proposition 2.1, which reduces the number of edges in the input graph and is independent of where demand is located, the graph resulting from such a transformation would depend on which vertices have a demand. Since we already limit to two the maximum number of edge traversals, the main difference on the outcome of this preprocess could be a reduction on the number of second traversal variables, at the expenses of possibly increasing the number of first traversal edges.

2.1 Comparison of LP bounds

In [20] it is proven that the LP bound of the Fleischmann [15] formulation coincides with that of the multi-commodity flow formulation [20]. To the best of our knowledge this is the best lower bound from an LP relaxation of a formulation for the STSP. Next we compare the lower bound from the LP relaxation of (1)–(6) to that bound.

Proposition 2.2 *The lower bound from the LP relaxation of formulation* (1)–(6) *coincides with the LP lower bound of the Fleischmann [15] formulation.*

Proof: The Fleischmann formulation uses general integer decision variables z_e, $e \in E$ representing the number of times that edge e is traversed. The formulation uses an additional set of auxiliary variables, associated with the vertices, which are used to linearize the parity constraints on the vertices. In particular, for all $u \in V$, h_u is a general integer variable equals to $z(\delta(u))/2$.

The Fleischmann formulation is the following:

$$\text{minimize} \quad \sum_{e \in E} c_e z_e \tag{7}$$

subject to

$$z(\delta(S)) \geq 2 \qquad S \subset V, S \cap V_R \neq \emptyset, V_R \setminus S \neq \emptyset \tag{8}$$

$$z(\delta(u)) = 2h_u \qquad u \in V \tag{9}$$

$$z_e \in \mathbb{Z}_+ \qquad e \in E \tag{10}$$

$$h_u \in \mathbb{Z}_+ \qquad u \in V. \tag{11}$$

Let z_{LP} and z^F_{LP} denote the values of the LP relaxations of formulation (1)–(6) and formulation (7)–(11), respectively. We first show that $z^F_{LP} \leq z_{LP}$. For this, it suffices to see that any feasible solution to the LP relaxation of (1)–(6) induces a feasible solution to the LP relaxation of the Fleishman formulation (7)–(11), and the objective function values of both solutions coincide. Let $(\widehat{x}, \widehat{y})$ be a feasible solution to the LP relaxation of (1)–(6). Consider the solution $(\widehat{z}, \widehat{h})$, where $\widehat{z}_e = \widehat{x}_e + \widehat{y}_e$, $e \in E$, and $\widehat{h}_u = \frac{1}{2}(\widehat{x} + \widehat{y})(\delta(u))$. Indeed, $\sum_{e \in E} c_e \widehat{z}_e = \sum_{e \in E} c_e(\widehat{x}_e + \widehat{y}_e)$. By construction $(\widehat{z}, \widehat{h})$ satisfies (9). It is also clear that $(\widehat{z}, \widehat{h})$ satisfies (8), given that $(\widehat{x}, \widehat{y})$ satisfies the connectivity constraints (2). Therefore, $z^F_{LP} \leq z_{LP}$.

In order to see that $z_{LP} \leq z^F_{LP}$ also holds, first observe that the optimal value of the LP relaxation of the Fleishmann formulation can be determined by only considering continuous solutions to (7)–(11) where $z_e \leq 2$. Therefore, it is enough to show that any feasible solution with $z_e \leq 2$ induces a feasible solution to the LP relaxation of formulation (1)–(6), with the same objective function value. Let $(\widehat{z}, \widehat{h})$ be a feasible solution to the LP relaxation of (7)–(11), with $\widehat{z}_e \leq 2$. Then, consider the solution $(\widehat{x}, \widehat{y})$ with $\widehat{x}_e = \widehat{y}_e = \frac{1}{2}\widehat{z}_e \leq 1$. Given that $\widehat{z}$ satisfies (8), $(\widehat{x}, \widehat{y})$ also satisfies the connectivity constraints (2) by construction. Furthermore, since $\widehat{x}_e = \widehat{y}_e$ for all $e \in E$, the solution $(\widehat{x}, \widehat{y})$ also satisfies all the parity constraints (3). In particular, for any $u \in V$, $H \subseteq \delta(u)$, $|H|$ odd, $(\widehat{x} - \widehat{y})(\delta(u) \setminus H) = 0$ so

$$(\widehat{x} - \widehat{y})(\delta(u) \setminus H) + \widehat{y}(H) = \widehat{y}(H) = \widehat{x}(H) \geq \widehat{x}(H) - |H| + 1.$$

Therefore, $(\widehat{x}, \widehat{y})$ is a feasible solution to the LP relaxation of (1)-(6) with the same objective function value as $(\widehat{z}, \widehat{h})$. Therefore, we also have that $z_{LP} \leq z^F_{LP}$. Hence, $z_{LP} = z^F_{LP}$ and the result follows. ■

Since the lower bound from the LP relaxation of the multi-commodity flow formulation of [20] coincides with the LP bound of the formulation of [15], we also have:

Corollary 2.3 *The lower bound from the LP relaxation of formulation* (1)–(6) *coincides with the LP lower bound from the multi-commodity flow formulation of [20].*

2.2 Valid inequalities

Next we present several families of valid inequalities that can be used to reinforce the LP relaxation of formulation (1)–(6).

Connectivity inequalities for subsets of non-required vertices.

Proposition 2.4 *Let $S \subset V \setminus \overline{V}_R$ be a set of non-required vertices. Then the following connectivity inequalities are valid for the LP relaxation of the formulation* (1)–(6)*:*

$$(x + y)(\delta(S)) \geq 2x_e \qquad e \in \gamma(S). \tag{12}$$

Inequalities (12) impose that if an edge connecting two non-required vertices of S is traversed, then in the cutset of S at least two edges must be traversed or one edge must be traversed, at least twice. Similar inequalities have been used for other routing problems associated with subsets of vertices that do not necessarily have to be visited by a route (see, e.g. [3, 5]).

Parity inequalities for subsets of vertices. For binary solutions, the cocircuit inequalities (3) associated with singletons also guarantee the parity of subsets of vertices. However, it is well-known that when relaxing the integrality constraints on the decision variables, the cocircuit inequalities (3) associated with singletons no longer guarantee the parity of subsets of vertices, which must be satisfied by any feasible solution. The LP relaxation of (1)–(6) can be reinforced by adding parity constraints associated with subsets of vertices.

Proposition 2.5 *Let $\emptyset \subset S \subset V$ be a subset of vertices and $H \subseteq \delta(S)$, $|H|$ odd, a subset of its cut-set with an odd number of edges. Then, the following inequality is valid for the LP relaxation of formulation* (1)–(6)*:*

$$(x - y)(\delta(S) \setminus H) + y(H) \geq x(H) - |H| + 1. \tag{13}$$

These inequalities (13) can be interpreted similarly to constraints (3): if an odd number of edges in the cut-set of a vertex set S are used, then at least one additional traversal must be made, which must be either a second traversal of some of the selected edges or a first traversal of some of the non-selected edges in the cut-set.

Figure 1 illustrates this situation with an example of a fractional solution that satisfies the connectivity constraints (2) as well as the parity constraints (3) associated with singletons, but violates a parity constraint (13) associated with a set of vertices S with $|S| > 1$. The depot is represented by a square vertex, customers are represented by black nodes, and the white nodes have no demand. Straight lines indicate variables with value 1, and dotted lines variables with values 0.5. In particular, we have $x_{d3} = x_{d7} = x_{12} = x_{24} = x_{45} = x_{46} = 1$, $x_{13} = x_{34} = x_{56} = x_{57} = x_{67} = y_{12} = y_{24} = 0.5$. As can be seen, the depicted solution satisfies the connectivity constraints, as well as the parity constraints associated with singletons. However the parity constraint (13) associated with $S = \{1, 2, 3, 4\}$, and $H = \{(d, 3), (4, 5), (4, 6)\}$ is violated, since $x(H) - |H| + 1 = 1$, but $(x - y)(\delta(S) \setminus H) + y(H) = 0$.

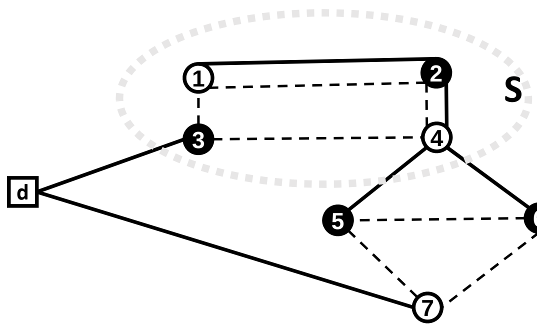


Figure 1: Example of a violated parity constraint (13)

Dead-end inequalities.

We first introduce dead-end inequalities for singletons of non-required vertices and then we extend them to general sets of non-required vertices.

Definition 2.6 *A vertex $u \in V$ is a* dead-end *for a feasible tour T, if T uses only one edge of $\delta(u)$, which is traversed twice.*

It is clear that there exists an optimal solution to a given STSP instance where the only dead-ends are required vertices. Otherwise, by eliminating the two copies of the edge visiting any non-required dead-end, we obtain a new feasible solution of cost not exceeding that of the original route. Therefore, in an optimal solution, any visited non-required vertex $u \in V \setminus \overline{V}_R$ must be connected to at least two different vertices. This means that in any optimal solution there must be at least two different edges incident with u. Hence, we have the following result:

Proposition 2.7 *The following inequalities are valid for the STSP:*

$$x\big(\delta(u) \setminus e\big) \geq x_e \quad u \in V \setminus \overline{V}_R,\ e \in \delta(u). \tag{14}$$

In order to extend the above inequalities to sets of non-required vertices we first provide a more general definition of *dead-end* for subsets of vertices.

Definition 2.8 *A vertex set $S \subset V$ is a* dead-end *for a feasible tour T, if all the edges of $\delta(S)$ used in T are incident with the same vertex $v \in V \setminus S$.*

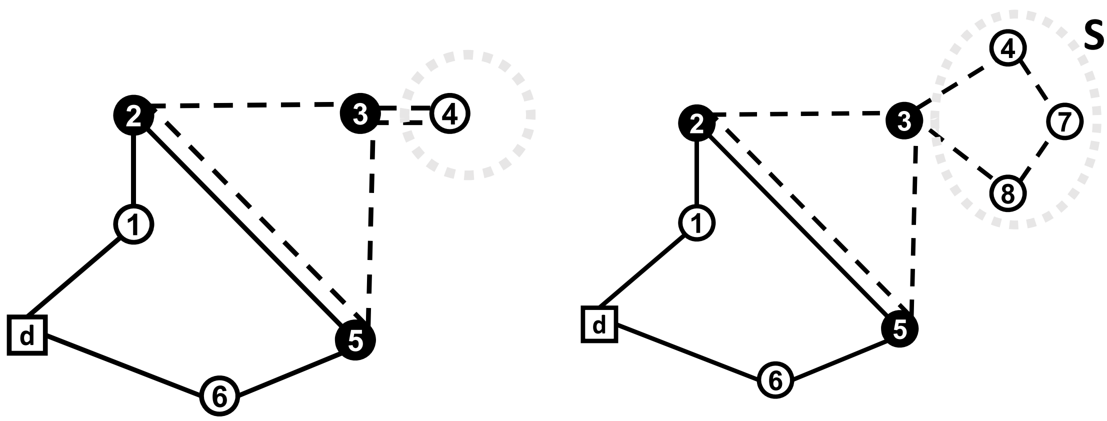
Indeed, there exists an optimal solution to a given STSP instance where any dead-end set contains some required vertex. Therefore we can exclude dead-end sets consisting of non-required vertices only.

Proposition 2.9 *The following inequalities are valid for the STSP:*

$$x\big(\delta(S) \setminus \delta(v)\big) \geq x_e, \qquad S \subseteq V \setminus \overline{V}_R, e = (u, v) \in \delta(S) \text{ with } u \in S, v \notin S. \tag{15}$$

Inequalities (15) impose that if, for a given orientation of the solution, an edge *enters* the set of non-required vertices S *from* vertex v, then an edge incident to a vertex different from v must *leave* S.

Figures 2(a) and 2(b) respectively illustrate inequalities (14) and (15) violated by fractional solutions that satisfy connectivity and parity constraints. In both figures the depot is represented by a square vertex, customers are represented by black nodes, and white nodes have no demand. Straight lines indicate variables at value 1, and dotted lines variables at values 0.5.



(a) Violated inequality (14) for $S = \{4\}$

(b) Violated inequality (15) for $S = \{4, 7, 8\}$

Figure 2: Examples of violated dead-end inequalities

2.3 Solution Algorithm

We have developed an exact branch-and-cut algorithm to solve the STSP based on the formulation proposed above. As usual, initially the integrality conditions are relaxed as well as most of the connectivity and parity constraints (2) and (3), respectively. In particular, initially we only consider the connectivity constraints (2) associated with the singletons $S = \{u\}$ with $u \in \overline{V}_R$, as well as those associated with vertex subsets derived from the connected components induced by the demand vertices. As for the parity constraints (3), for each $u \in V$ initially we only consider the constraints associated with subsets of edges H given by individual edges $e \in \delta(u)$. That is, we consider the subset of constraints (3) with $u \in V$, $e \in \delta(u)$, $H = \{e\}$. At each iteration of the solution algorithm, we solve the current LP and we look for inequalities that are violated by the LP solution, which are then incorporated into the relaxed model.

The branch-and-cut strategy that we apply is the following. If the LP solution is integer, exact separation procedures for (2) and (3) are applied, whereas if the LP solution is fractional, violated inequalities (12), (13), and (14) may be separated as well. In the exploration of the search tree, fractional solutions are only separated at (*i*) the root node; (*ii*) the first three levels; and (*iii*) levels that are multiples of $n/2$. To identify inequalities (2), (12), and (13) violated by fractional solutions, we first apply heuristic separation procedures. Only at the root node, whenever violated inequalities of each of these types are not found heuristically, we also apply exact separation procedures, which can be carried out in polynomial time [21, 24]. For the connectivity inequalities (2), (12) we use standard procedures similar to those that have been applied by many authors both for node and arc routing problems [5, 24]. The procedure that we apply for the parity inequalities (13) has also been used for several arc routing problems. The reader is referred to [1, 2, 3, 12] for further details. Dead-end inequalities (14) are separated by inspection.

At any node of the enumeration tree, we exit the LP solution loop whenever the LP objective value has not improved for a given number of consecutive iterations, independently of whether or not additional violated inequalities can be found.

The specific parameters we use in the implementation of our solution algorithm are the following:

- Heuristic separation of connectivity and parity inequalities (2), (12), and (13): In all cases, the heuristic separation procedures that we apply for these inequalities consist of finding connected components in some ad hoc graph induced by the edges associated with values greater than or equal to a given threshold value ε. The values of ε that we use are 0.2 and 0.1.
- The iterative LP loop terminates when the LP value has not improved for ten consecutive iterations.

2.4 Computational experiments

We now describe the experiments we have carried out in order to assess the behavior of the exact algorithm that we apply to solve our model. We have used two sets of benchmark instances. The first set, S1, was generated and used in [20], so that we can compare our results with those reported in that paper. To further assess the performance of formulation 2IF and analyze its scalability, we have generated and tested a new set S2 of larger instances, with a procedure similar to the one described in [20]. The set S1 consists of sparse graphs of size $n \in \{25, 50, 75, 100, 125, 150, 175, 200, 225, 250\}$, whereas the size of instances of S2 is $n \in \{275, 300, 325, 350, 375, 400, 425, 450, 475, 500\}$. The instances are classified into three groups, which differ from each other in the proportion $r = |V_R|/n$ of required vertices. In the first set $r = 1/3$, whereas in the second and third sets $r = 2/3$ and $r = 1$, respectively. When $r = 1$ all the vertices are required, so that the instances correspond to TSPs on incomplete sparse graphs. Hence, there are in total 60 instances (30 in S1 and 30 in S2), three for each value of n. The sparsity level of the considered instances $s = |E|/\big(n(n-1)/2\big)$ ranges in $[0.01, 0.1]$, where smaller values close to 0.01 correspond to instances of sizes greater than or equal to 200, and larger values close to 0.3 correspond to the smallest instances with $n = 25$. In all cases vertex 0 is set as the depot, independently of whether or not it is a required vertex. Table 1 summarizes information of the benchmark instances. The column under $|V|$ gives the number of vertices. Columns $|E_0|$ and $|E|$ show the number of edges initially and after applying proposition 2.1, respectively. The column under s gives the values of sparsity s. The columns under C_r give, for each demand ratio r, the number of connected components induced by demand nodes.

Table 1: Summary of the test instances for the STSP

Instances	$\|V\|$	$\|E_0\|$	$\|E\|$	s	C_r $r = 1/3$	$r = 2/3$	$r = 1$
S1	25	31	21	0.10	7	2	1
	50	69	66	0.06	7	9	1
	75	105	89	0.04	14	7	1
	100	139	120	0.03	15	8	1
	125	179	161	0.02	20	9	1
	150	215	192	0.02	21	13	1
	175	252	231	0.02	31	12	1
	200	291	266	0.01	35	23	1
	225	326	295	0.01	45	22	1
	250	367	353	0.01	46	19	1
S2	275	350	294	0.01	59	45	1
	300	395	336	0.01	55	42	1
	325	416	345	0.01	63	33	1
	350	464	397	0.01	68	40	1
	375	493	413	0.01	70	44	1
	400	517	443	0.01	67	36	1
	425	563	490	0.01	82	43	1
	450	595	496	0.01	77	46	1
	475	616	553	0.01	99	64	1
	500	666	573	0.01	97	62	1

The branch-and-cut algorithm was implemented in C++ and the experiments were run on a 2.80 GigaHertz Intel Core i7 machine with 16 Gigabytes of memory. We have used the IBM CPLEX 12.7 Concert Technology with default parameters, except for the cuts generated by CPLEX, which were disabled.

The following tables compare our results with those reported in [20] for the different formulations they introduced in that paper. The columns under 2IF give the results obtained with our compact formulation, whereas the remaining columns reproduce the results of [20]: SCF1 and SCF2 refer to two alternative single-commodity flow formulations, MCF refers to a multi-commodity flow formulation, and TSF1 and TSF2 refer to two alternative time-staged formulations. Since the computer on which we have run our experiments is different from that of [20], a precise comparison of the respective computing times is not possible (see, e.g Dongarra [11], or the website of the SPEC [26]). A rather coarse estimation based on their respective characteristics would establish that our computer is twice as fast as that of [20].

Tables 2–6 summarize the information on the number of instances of each group S1-r solved to optimality, integrality gaps, computing times at the root node, total number of explored nodes, and overall computing times. Results for each individual instance can be found in https://www.eio.upc.edu/en/homepages/elena/steiner.

Table 2: Instances solved to optimality

Instances	SCF1	SCF2	MCF	TS1	TS2	2IF
S1-1/3	10	9	10	1	1	10
S1-2/3	8	8	5	1	1	10
S1-3/3	10	9	4	1	1	10

Table 2 reports the number of instances of each group solved to optimality by each formulation. We were able to find optimal solutions for all 30 instances within computing times that are always smaller than 350 seconds. In contrast, the formulation SCF1 of [20] with the best overall performance, found an optimal solution for 28 out of the 30 instances within their computing time limit of 5,000 seconds. Even if we take into account the difference in the performance of the computers, these results clearly show that our alternative formulation outperforms all the formulations presented in [20].

Table 3: Average percent integrality gaps of LP relaxations

Instances	SCF1	SCF2	MCF	TS1	TS2	2IF
S1-1/3	30.65	30.16	0.44	44.30	43.00	0.66
S1-2/3	23.82*	23.36*	1.09*	30.94*	29.80*	1.37
S1-3/3	16.38	16.15	0.85	19.33	18.57	0.67

Table 3 presents average integrality gap percentages at the root node. The entries marked with an asterisk correspond to the formulations of [20] for the group of instances with $r = 2/3$. Since instances 225-2/3 and 250-2/3 could not be optimally solved by any of their formulations, these two instances were excluded for the computation of the averages of the group. Table 3 shows that, among the formulations of [20], the smallest LP gaps are produced by the multi-commodity flow formulation MCF, which produces very tight bounds. In general, 2IF reinforced with inequalities (13) and (14) produces smaller LP gaps than MCF, although in some cases MCF produces smaller gaps. This may seem contradictory with Corolary 2.3, although it can be explained by the termination criterion for the LP loop. As explained, we may exit the loop before reaching the optimal LP solution when there is no improvement in the LP bound, even if there are violated inequalities. Anyhow, our compact formulation 2IF produces the smallest LP gap for 21 of the 30 instances, whereas the MFC formulation produces the smallest LP gap for 16 instances. For seven instances both formulations yield the same outcome.

The effort related to the computation of the lower bounds at the root node is summarized in Table 4, which gives the average computing times needed to solve the LP relaxation of each of the compared formulations for the three sets of instances. As can be seen, the computing times required by multi-commodity flow formulation MCF, for instances with a small proportion of required vertices ($r = 1/3$) are roughly comparable to those required by our formulation 2IF. However, as the proportion of required vertices increases, the computational requirements of MCF tend to explode and their computing times are substantially outperformed by those of 2IF.

Table 4: Average computing times (in seconds) for solving the LP relaxation

Instances	SCF1	SCF2	MCF	TS1	TS2	2IF
S1-1/3	0.04	0.04	18.31	88.74	50.32	17.86
S1-2/3	0.04	0.04	683.91	119.97	68.68	19.61
S1-3/3	0.04	0.05	1134.49	168.66	94.45	5.07

Table 5: Average number of branch-and-bound nodes

Instances	SCF1	SCF2	MCF	TS1	TS2	2IF
S1-1/3	99177.70	72298.33*	31.79	12370.00*	8733.00*	162.30
S1-2/3	59916.75*	324095.00*	158.05*	6576.00*	9908.00*	4339.70
S1-3/3	158331.20	137018.22*	58.50*	78302.00*	38508.00*	1299.70

Tables 5 and 6 summarize information on the overall performance of the formulations. In particular, Table 5 presents the average number of branch-and-bound nodes explored to solve the instances to proven optimality, and Table 6 presents average overall computing times. In Table 5, for each formulation, the averages of the groups where not all instances were optimally solved, have been computed over the subset of instances optimally solved, and the corresponding entries are marked with an asterisk. In Table 6, for each formulation, for the instances that were not optimally solved within that time with the corresponding formulation, we have used the maximum computing time allowed in [20] (5,000 seconds). Table 6 is complemented with Table 7 where the averages are computed across the sizes instead of the proportion of required vertices.

Table 6: Average total computing times over proportion of required vertices (in seconds)

Instances	SCF1	SCF2	MCF	TS1	TS2	2IF
S1-1/3	289.93	636.20	199.66	4507.99	4505.81	21.46
S1-2/3	1086.07	1464.95	2871.12	4510.52	4507.20	59.07
S1-3/3	523.53	745.76	3062.31	4562.40	4527.17	12.31

Among the formulations proposed in [20], MFC again produces the best results, in terms of the number of nodes explored, for the instances that could be solved between the maximum computing time. However, MFC becomes inefficient as the proportion of required vertices increases, due to its memory and computational requirements: five instances with $r = 2/3$ and six instances with $r = 1$ could not be solved. In contrast, our formulation 2IF, which, as already stated, was able to optimally solve all the benchmark instances, explored a moderate number of nodes in all cases.

The superiority of 2IF over all other formulations compared is confirmed in Tables 6 and 7. As can be seen, not only all instances can be solved, but the computing times needed to certify the optimality of the solutions are also very moderate, and much smaller

Table 7: Average total computing times over size (in seconds)

Instances	SCF1	SCF2	MCF	TS1	TS2	2IF
S1-25	0.17	0.04	0.03	269.70	133.95	0.16
S1-50	1.71	0.78	0.63	5000.00	5000.00	0.75
S1-75	2.65	4.28	6.22	5000.00	5000.00	1.53
S1-100	8.68	7.65	208.74	5000.00	5000.00	5.58
S1-125	27.94	44.48	2897.38	5000.00	5000.00	12.64
S1-150	147.90	136.24	3370.49	5000.00	5000.00	28.04
S1-175	234.43	451.53	3343.44	5000.00	5000.00	48.17
S1-200	211.79	1479.07	3340.03	5000.00	5000.00	32.64
S1-225	834.51	699.00	3828.80	5000.00	5000.00	132.55
S1-250	1528.72	3333.36	3447.87	5000.00	5000.00	47.38

than those of any of the formulations used in the comparison.

Table 8: Average results for larger instances S2.

Instances	♯ Instances	TimeLP	GapLP	Time	Gap	Nodes
S2-1/3	9	721.96	2.40	1473.87	0.19	14403.00
S2-2/3	9	295.71	2.26	1282.96	0.02	47918.20
S2-3/3	10	141.69	1.74	1046.59	0.00	180033.00

The results presented in Table 8 show the good performance of the proposed formulation for the set S2 of larger instances. As can be seen, all instances but two were optimally solved within the time limit, now set to 7,200 seconds. The two instances that could not be optimally solved are 500-1/3 and 475-2/3. Their percentage optimality gaps at termination are 0.19% and 0.02%, respectively.

We close this section with an analysis of the performance of the branch-and-cut algorithm. Table 9 presents average results, over the complete set of 60 benchmark instances, on the number of cuts of each type generated at the root node and during the exploration of the enumeration tree. The number of cuts generated is in general quite moderate, with more connectivity constraints than parity cuts. The table also shows that quite a few new dead-end inequalities are also generated.

Table 9: Summary of generated cuts for the two sets of benchmark instances

	Root node			Enumeration tree		
Instances	Connec	Parity	Dead-end	Connec	Parity	Dead-end
S(1+2)-1/3	183.45	18.40	249.77	469.35	61.75	348.85
S(1+2)-2/3	200.35	29.50	128.28	568.15	98.65	189.30
S(1+2)-3/3	228.90	51.10	0.00	201.05	93.85	0.00

3 Steiner Multi-depot Traveling Salesman Problems

SMDTSPs are a natural extension of the STSP in which there are several depots instead of only one. As in the previous section, we consider SMDTSPs defined on a undirected incomplete connected graph $G = (V, E)$, where V is the vertex set, E is the edge set, each of them with an associated traversal cost $c_e \geq 0$, and $V_R \subseteq V$ is the set of customers to visit. We denote by $D \in V$ the set of depots. Broadly speaking, an SMDTSP is to determine a minimum-cost set of non-empty closed routes rooted at depots, that visit each customer at least once.

SMDTSPs are akin to the Multi-Depot Rural Postman Problem (MDRPP) studied in [12] in the sense that the feasibility of its solutions can be established essentially via connectivity with the depots and parity conditions in each route, in addition to the requirement that all customers must be served. Of course, such requirements induce notable differences between these two problems, since now the demand is placed at some nodes of the network instead of on some edges. Moreover, specific modeling hypotheses that we will make, will further condition the characteristics of feasible SMDTSP solutions. In this section we develop alternative SMDTSP models that differ from each other in (i) whether or not all the depots must be used, in the sense that there is a non-empty route starting and ending at each of the depots, and (ii) whether or not each route must traverse a single depot, which naturally defines the root of the route. In the following, SMDTSP solutions that can be decomposed into individual routes, rooted at depots, where no route traverses any other depot different from its root, are called *consistent*. The particular SMDTSP models that we study are the following:

- Consistent + All-depots SMDTSP (CA-SMDTSP): Solutions are *consistent* and all the depots have to be used. Hence solutions can be decomposed in $|D|$ routes, each of which rooted at a different depot, and no route traverses a depot different from its root.
- Consistent + Optional-depots SMDTSP (CO-SMDTSP): Solutions are *consistent* but the condition that all depots have to be used is relaxed. This means that some depots may not be used, but each of the used depots belongs to only one of the routes.
- Non-consistent + Optional-depots SMDTSP (NO-SMDTSP): The condition that the routes are *consistent* is relaxed, as well as the condition that all depots have to be used. This means that some routes may *traverse* several depots and, in fact, any of them can be *deemed* as the depot of the route.

Figure 3 highlights the differences among the above models on a simple example. Figure 3(a) depicts the input graph with three depots, $d1$, $d2$, and $d3$ (represented by square vertices), where customers are represented by black nodes and white nodes have no demand. All horizontal edges have a cost of 1, and all the vertical edges have a cost of $1+\varepsilon$, with $\varepsilon > 0$. Figures 3(b), 3(c), and 3(d) depict optimal solutions for the models CA-SMDTSP, CO-SMDTSP, and NO-SMDTSP, respectively. In Figure 3(b) it can be seen that all depots are used in the optimal solution to CA-SMDTSP and that the routes are consistent. The cost of this solution is $14 + 2\varepsilon$. In the optimal solution to CO-SMDTSP shown in Figure 3(c), the routes are again consistent, but now depot $d3$ is not used. The cost of this solution is also $14 + 2\varepsilon$. The optimal solution to NO-SMDTSP consists of a unique non-consistent route of cost $12 + 2\varepsilon$, traversing all three depots, although only

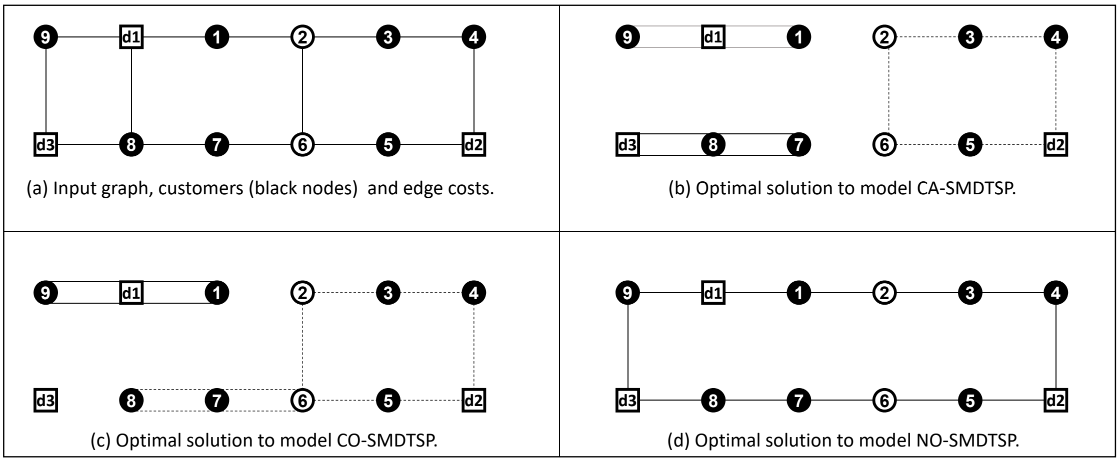


Figure 3: Alternative MSMDTSP models

one of them would be arbitrarily deemed as *its depot*.

Any feasible solution to a given CA-SMDTSP instance is feasible for the CO-SMDTSP instance with the same parameters. In turn, any feasible solution to a given CO-SMDTSP instance is feasible for the corresponding NO-SMDTSP instance. Therefore, if we respectively denote by $F(\texttt{prob})$ and $v(\texttt{prob})$ the set of feasible solutions and the optimal value to problem `prob`, we have:

Proposition 3.1

- *F(CA-SMDTSP)* $\subseteq$ *F(CO-SMDTSP)* $\subseteq$ *F(NO-SMDTSP).*
- *v(CA-SMDTSP)* $\geq$ *v(CO-SMDTSP)* $\geq$ *v(NO-SMDTSP).*

Optimality conditions from the STSP also apply to each individual route of all three SMDTSP models discussed above. However, as it has been shown in the example that since the graph is not complete, the individual routes are not necessarily disjoint. Therefore, since there exists an optimal solution in which no edge is traversed more than twice in each route, there also exists an optimal SMDTSP solution in which no edge is traversed more than twice the number of depots. Proposition 2.1 still holds for all three SMDTSP models. However, in the case of the CA-SMDTSP, shortest paths connecting a depot and a required vertex must be restricted to those not traversing any other depot.

Taking into account the above properties, a natural option for formulating SMDTSPs is to use three-index binary variables that associate edges with depots, and then define the route of each depot. This strategy allows us to define feasible routes by only imposing connectivity and parity conditions on each of the routes. However, the number of variables involved with this scheme increases with the number of depots. An alternative option that we will adopt consists of considering aggregated two-index variables associated solely with edges. This reduces the number of decision variables, at the expense of possibly losing the feasibility of the routes for some of the above models, when only imposing connectivity and parity constraints. In particular, connectivity with depots of the sets of customers and parity constraints on the visited vertices alone do not guarantee that all the depots are used. These constraints alone neither guarantee that the routes are consistent. Therefore, for some of the SMDTSP models, two-index variables will require the imposition of additional constraints. Below we develop ILP formulations and exact solution algorithms for each of the three SMDTSPs just introduced, which will be compared computationally in Section 5.

3.1 Consistent + All-depots SMDTSP

As mentioned, when using aggregated two-index variables, connectivity and parity constraints alone do not guarantee that all depots are used or that solutions are made up consistent routes. The first condition can be easily satisfied by imposing ad hoc connectivity constraints for the depots. The second difficulty can be overtaken with an adaptation of the return-to-depot constraints introduced in [12] for the MDRPP that we detail below.

Before introducing the inequalities, we define the two sets of variables that we will use in the ILP formulation for the CA-SMDTSP. For each $e \in E \setminus \gamma(D)$, binary variable x_e indicates whether or not edge e is traversed by *some route*. Now the y_e variables represent the number of additional traversals of edge $e \in E \setminus \gamma(D)$. In principle, these are general integer variables, although some can be restricted to take binary values. A consequence of enforcing consistent routes is that the edges incident with depots can only be traversed in routes rooted at their incident depots. Therefore, such edges can be traversed at most twice and their associated y variables are binary. That is, $y_e \in \{0, 1\}$ for all $e \in E_D = \{(u, v) \in E | u \in D \text{ or } v \in D\}$, whereas $y_e \in \mathbb{Z}_+$ for all $e \in E \setminus E_D$. Note that since routes should be consistent, variables corresponding to edges $e \in \gamma(D)$ connecting two depots are not defined.

3.1.1 Return-to-depot inequalities

Definition 3.2 *Consider a vertex set S not containing any depot, i.e. $S \subseteq V \setminus D$, and a subset of depots $\emptyset \neq D' \subseteq D$, $D' = \{d_i\}_{i \in I}$. A subset of edges $H \subseteq \delta(S)$ is an odd-D'-cut for S if $H = \bigcup_{i \in I} H_i$ with $H_i \subseteq \delta(S) \cap \delta(d_i)$ and $|H_i|$ odd, for all $i \in I$.*

When $H \subseteq \delta(S)$ is an odd-D'-cut for S, we denote by $Q_H = (\delta(S) \setminus H) \cap \delta(D')$ the edges of $\delta(S) \setminus H$ incident with some depot of D', and by $\overline{Q}_H = \delta(S) \setminus \delta(D)$ the edges of $\delta(S)$ incident with no depot.

The following result is adapted from [12], and is reproduced here to better understand the rationale supporting the inequalities and their terms.

Proposition 3.3 *Let $S \subseteq V \setminus D$, $D' \subseteq D$, $D' = \{d_i\}_{i \in I}$ with $|D'| \geq 1$ and $H \subseteq \delta(S)$ an odd-D'-cut for S. Then the inequality* (16) *is valid for the CA-SMDTSP.*

$$(x - y)(Q_H) + (x + y)(\overline{Q}_H) + y(H) \geq x(H) - |H| + |D'|. \tag{16}$$

Proof: Let (x, y) be the support vector of a feasible CA-SMDTSP solution and let $H \subseteq \delta(S)$ be an odd-D'-cut for S. Since constraint (16) can only be active when $x(H) - |H| + |D'| > 0$ we consider the following cases:

(*a*) $x(H) = |H|$, i.e., all the edges of H are used (traversed for a first time). Since each $|H_i|$ is odd, in order to guarantee that the route starting at each d_i, $i \in I$ also terminates at the same d_i, at least $|D'|$ additional traversals of edges of $\delta(S)$ are needed. This is imposed by the right-hand side of inequality (16), which takes the value $|D'|$, since $x(H) - |H| = 0$.

The additional traversals can be of one of the following types:

- Additional traversals of edges of H. This is represented by the term $y(H)$ of the left-hand side.
- Traversals of edges of $\delta(S) \setminus H$ incident with no depot. Since such edges can be traversed in the routes rooted at different depots, multiple traversals of these edges may appear in feasible routes. This is represented by the term $(x + y)(\overline{Q}_H)$ of the left-hand side.
- Traversals of edges of $\delta(S) \setminus H$ incident with some depot. In this case, only the edges incident with depots in D' are of interest. Since such edges can be traversed only in routes rooted at the depots to which they are incident, we are only interested in their first traversal (for which the variable associated with the second traversal is binary). This is represented by the term $(x - y)(Q_H)$ of the left-hand side.

Therefore, when all the edges of H are traversed at for a first time, inequality (16) is valid.

(b) $x(H) = |H| - 1$, i.e., $|H| - 1$ edges of H are used. This case can be transformed into case (a) as follows. Let $\hat{\imath} \in D'$ denote the index such that $x(H_{\hat{\imath}}) = |H_{\hat{\imath}}| - 1$. Consider the set of depots $\overline{D}' = D' \setminus \{\hat{\imath}\}$ and note that $H' = H \setminus H_{\hat{\imath}} \subseteq \delta(S)$ is an odd-$\overline{D}'$-cut for S. Since $x(H') = |\overline{D}'|$, its associated inequality (16) is valid since it corresponds to case (a). Note that the right-hand sides of the inequalities associated with H and H' coincide, i.e. $x(H) - |H| + |D'| = x(H') - |H'| + |\overline{D}'|$. Note also that $Q_{H'} \subset Q_H$ and $\overline{Q}_{H'} = \overline{Q}_H$. Therefore, the inequality (16) associated with H is valid since it is dominated by the one associated with H'.

(c) All other cases can be handled similarly.

Therefore, inequality (16) is valid for the CA-SMDTSP and the result follows. ■

The ILP formulation we propose for the CA-SMDTSP is the following:

$$(CA) \quad \text{minimize} \quad \sum_{e \in E} c_e(x_e + y_e) \tag{17}$$

subject to

$$(x+y)(\delta(d)) \geq 2 \qquad d \in D \tag{18}$$

$$(x+y)(\delta(S)) \geq 2 \qquad S \subseteq V \setminus D, (S \cap V_R) \neq \emptyset \tag{19}$$

$$x(\delta(S) \setminus H) + y(H) \geq x(H) - |H| + 1 \qquad S \subset V,\ H \subseteq \delta(S), |H| \text{ odd} \tag{20}$$

$$(x-y)(Q_H) + (x+y)(\overline{Q}_H) + y(H) \geq x\,(H) - |H| + |D'| \qquad S \subset V \setminus D, \emptyset \neq D' \subseteq D,\ H \text{ odd-}D'\text{-cut for } S \tag{21}$$

$$y_e \leq (2|D| - 1)x_e \qquad e \in E \setminus E_D \tag{22}$$

$$y_e \leq x_e \qquad e \in E_D \tag{23}$$

$$x_e \in \{0, 1\} \qquad e \in E \setminus \gamma(D) \tag{24}$$

$$y_e \in \mathbb{Z}_+ \qquad e \in E \setminus \gamma(D). \tag{25}$$

Constraints (18) mean that all depots are used. Connectivity constraints (19) impose that at least two edges cross the cut-set $\delta(S)$ of any given vertex set including some customer but not containing any depot. When $S = \{u\}$ with $u \in V_R$, they also ensure that all demand customers are visited. Constraints (20) ensure the parity of every subset of vertices. Observe that they are weaker than the analogous constraints (13) for the single-depot STSP, since the first term in the left-hand side is now $x(\delta(S) \setminus H)$ instead of $(x-y)(\delta(S)\setminus H)$. This is necessary, given that the y variables may take values greater than one, which invalidates the tighter version (13). Constraints (21) are the return-to-depot inequalities presented above. Similar arguments to those used in [12] for the MDRPP can be applied to prove that, together with connectivity and parity constraints (19) and (20), they guarantee that routes are consistent. Constraints (22) and (23) impose that additional traversals of an edge cannot happen unless the edge has been traversed for a first time. Conditions on variables x and y derived from their definition are reflected in constraints (24) and (25).

The above formulation contains $|E|$ x and y variables. There are $|D|$ constraints (18), $|E \setminus D|$ constraints (22), and $|E_D|$ constraints (23). The size of the families constraints (19), (20), and (21) is exponential in $|V|$.

Inequalities similar to those presented in Section 2.2 can also be used to reinforce the above formulation. Now inequalities (12) are valid for vertex sets S that do not contain any depot, i.e. $S \subset V \setminus D$. Dead-end inequalities (14) are only valid for non-required vertices that are not depots, so they become:

$$x\big(\delta(u) \setminus e\big) \geq x_e \quad u \in V \setminus (V_R \cap D),\ e \in \delta(u). \tag{26}$$

3.1.2 Branch-and-cut solution algorithm for the CA-SMDTSP

Based on the above formulation, we have developed an exact branch-and-cut algorithm for the CA-SMDTSP following the scheme of that developed for the STSP. Now, the family of return-to-depot constraints (21), which is of exponential size, is also initially relaxed, jointly with the families of connectivity constraints (19) and parity constraints

(20). Similarly to the MDRPP, violated connectivity constraints (19) are not necessarily associated with minimum cuts in the support graph induced by edges with $G(\overline{x}, \overline{y})$ relative to the capacities vector $\overline{x} + \overline{y}$, where $(\overline{x}, \overline{y})$ is the current LP solution. Instead, in order to solve the separation problem for constraints (19) we have to operate on the subgraph $G^{V \setminus D}(\overline{x}, \overline{y})$ induced by the vertex set $V \setminus D$ and look for minimum cutsets relative to $\overline{x} + \overline{y}$. The real value for $G(\overline{x}, \overline{y})$ of a cutset $S \subset V \setminus D$ can be obtained from its value in $G^{V \setminus D}(\overline{x}, \overline{y})$, $v^{V \setminus D}(S)$. In particular, $(\overline{x} + \overline{y})(\delta(S)) = v^{V \setminus D}(S) + (\overline{x} + \overline{y})\left(\delta(S) \cap \delta(D)\right)$ (see [12] for details). Return-to-facility constraints (21) are handled as lazy constraints, so they are only separated by inspection at the nodes with an integer LP solution.

3.2 Consistent + Optional-depots SMDTSP

The only difference of the CO-SMDTSP with respect to the CA-SMDTSP is that it is possible not to use all the depots of D. Hence, in order to obtain a valid two-index variable formulation for the CO-SMDTSP we only need to remove constraints (18) from formulation (17)–(25). Again, the updated connectivity and the dead-end inequalities (12) and (26) can be used to reinforce its LP relaxation, and an exact branch-and-cut algorithm like the one of CA-SMDTSP, based on the updated formulation, is applied to solve the CA-SMDTSP to proven optimality.

3.3 Non-consistent + Optional-depots SMDTSP

In the NO-SMDTSP we no longer impose that solutions define consistent routes. Hence, the return-to-depot inequalities (21) are no longer valid. Therefore, in order to obtain a valid formulation for the NO-SMDTSP, they must be removed, in addition to constraints (18) forcing to use all depots. Furthermore, now variables associated to edges $e \in \gamma(D)$ connecting two depots are defined, since such edges can be part of feasible solutions.

The NO-SMDTSP satisfies the following properties, which apply neither to the CA-SMDTSP nor to the CO-SMDTSP.

Property 3.4 *There exists an optimal NO-SMDTSP solution that satisfies the following properties:*

- *Each connected component in the graph induced by the solution* defines *one single route with just one depot deemed as the depot of the route. Otherwise, the routes corresponding to different (sets of) depots can be* merged *into a single one, and one of the depots visited in the merged route arbitrarily selected as the deemed depot of the component.*
- *The total number of traversals of any edge is at most two. After merging several routes into a single one as indicated above, if any edge is traversed more than twice, two copies of the edge can be removed while maintaining the feasibility of the solution, and not deteriorating the objective function value. This process can be repeated if needed.*

An implication of Property 3.4 is that there exists an optimal NO-SMDTSP solution in which there is at most one additional traversal of any of the edges that are used. Therefore, in the formulation for the NO-SMDTSP we exploit this property and all y variables are defined as binary. As a consequence, we can use the tight version of the parity constraints (3) instead of the weaker version (20). The exact branch-and-cut algorithm we have developed for the NO-SMDTSP, based on the resulting formulation, is similar to those described above. Details are therefore omitted.

Remark 3.1 We close this section with some comments on the various SMDTSP variants we have studied in this section. While CA-SMDTSP seems a suitable model when the (strategic) decision on the location of the depots has been taken in advance and it is known that the selected depots are needed and suitably placed, models CO-SMDTSP and NO-SMDTSP can somehow be considered as integrated location-routing models in the sense that the location of the depots for the routes is not known in advance, and is defined together with the edges that define the solution routes. Indeed, this is a *task* that can only be done *a posteriori*, after the solutions to the corresponding models are known, by identifying which depots are actually used. Thus, from the location-routing point of view, CO-SMDTSP and NO-SMDTSP entail limitations derived from not knowing in advance the set of depots, like, for example, the possibility of imposing cardinality constraints on the number of depots, or of considering set-up costs for the activated depots. Therefore, more general location-routing models that overcome these difficulties would seem appropriate. This is the topic of the next section.

4 Steiner Location Traveling Salesman Problem

In this section we introduce the SLTSP, which extends SMDTSPs to the case where the depots are not known in advance and their location is explicitly part of the decision to make. Like in the previous cases, we consider the SLTSP defined on a undirected incomplete connected graph $G = (V, E)$ with the same input data as before. Now, $D \in V$ denote the set of potential locations where facilities that will become the depots for the routes may be located. Selected facilities will also be referred to as *open* facilities. We assume that all open facilities will be used, in the sense that there will be at least one non-empty route incident to each of them. In addition, p denotes an upper bound on the number of facilities to be opened. Again we impose that the routes are consistent, although this concept is now updated in order to incorporate locational decisions. In particular, a route rooted at a facility located at vertex $d \subset D$ is consistent if it does not traverse any other open facility different from d. The SLTSP is to find a subset of at most p open facilities together with a set of non-empty consistent routes that visit each customer at least once at minimum routing cost.

It is clear that Proposition 2.1 concerning the edges that may appear in optimal SMDTSP solutions also applies to the SLTSP. Moreover, similar arguments to those of Property 3.4 for the NO-SMDTSP can be applied to conclude that there exists an optimal SLTSP solution in which every connected component of the graph induced by the edges used contains exactly one open facility. As a result, there exists an optimal SLTSP solution in which no edge is traversed more than twice. Hence, similarly to the STSP and the NO-SMDTSP we can derive formulations for the SLTSP in which routes are represented by two sets of two-index binary variables. On the other hand, SLTSPs present some similarities with the Location-Arc Routing Problem introduced in [13], mostly in what concerns to the adaptation of the return-to-depot constraints for integrating location variables. However, substantial differences remain due to the different type of demand that has to be served, as mentioned in Section 3.

Note that there exists a close relationship between the SLTSP and the NO-SMDTSP. The main difference is that in the SLTSP the set of facilities used as depots for the routes is explicit, whereas, as indicated in Remark 3.1, in the NO-SMDTSP this information is implicit in the solution of the problem. Another difference lies in the SLTSP constraint on the maximum number of facilities that can be selected, which is not imposed in the NO-SMDTSP. Of course, when $p = |D|$ both problems are essentially the same. Therefore, if we denote by $v(\text{SLTSP}_p)$ the optimal value to the LSTP for a given parameter value p, we have $v(\text{NO-SMDTSP}) \leq v(\text{SLTSP}_p)$ for all $p < |D|$, and $v(\text{NO-SMDTSP}) = v(\text{SLTSP}_{|D|})$.

The decision variables we use in our formulation for the SLTSP are defined as follows. For each $e \in E$, let x_e be a binary variable indicating whether or not edge e is traversed and let y_e be a binary variable that takes the value one if and only if edge e is traversed twice. For location decisions, for each $d \in D$, let z_d be a binary variable indicating whether or not facility d is opened. Thus, in the following, $z(D') = \sum_{d \in D'} z_d$ indicates number of facilities that are opened within set $D' \subseteq D$.

4.1 Return-to-depot inequalities for the SLTSP

The definition of the SLTSP requires that the routes be consistent. Thus, any valid SLTSP formulation with two-index routing variables must include constraints enforcing this property. Quite similarly to the SMDTSP, we will now consider vertex sets S not containing any potential facility, i.e. $S \subset V \setminus D$, subsets of potential locations $\emptyset \neq D' \subseteq D$, $D' = \{d_i\}_{i \in I}$, and odd-D'-cuts for S, i.e., $H \subset \delta(S)$, $H = \bigcup_{i \in I} H_i$ with $H_i \subseteq \delta(S) \cap \delta(d_i)$ and $|H_i|$ odd, for all $i \in I$. However, the return-to-depot constraints (21) of the SMDTSP models calling for consistent routes are no longer valid for the SLTSP. The reason is that the number of facilities that are opened in a given subset of potential locations D', which appears on the right-hand side of the inequalities, is no longer constant and becomes $z(D')$. Moreover, inequalities (21) with an updated right-hand side $x(H) - |H| + z(D')$ are still not valid for the SLTSP. The reason is again that the edges of $\delta(S) \setminus H$ that can be used to make the constraint valid also depend on the actual set of facilities that are opened in D'. In particular, the edges of $\delta(S) \setminus H$ that can be used in order to satisfy the constraint are of the following types: (*i*) second traversals of edges of H; (*ii*) first traversals of edges not incident with any potential locations; (*iii*) first traversals of edges incident with potential location of D', independently of whether or not the facilities they are incident to are opened; (*iv*) first traversals of edges incident with potential locations in $D \setminus D'$, provided that the corresponding location is not open.

Hence, for a given odd-D'-cut for S, $H \subset \delta(S)$, we now denote by $Q_H = (\delta(S) \setminus H) \setminus \delta(D \setminus D')$ the edges of types (*ii*) and (*iii*) that can be used independently of the locational decisions, by $\overline{Q}_H = (\delta(S) \setminus H) \cap (\delta(D \setminus D')$ the edges of type (*iv*) that can be only used when the corresponding location is not selected, and by $\overline{Q}_H^d = \overline{Q}_H \cap \delta(d)$ the edges of $\overline{Q}_H$ incident with a particular depot $d \in D \setminus D'$. Therefore, the return-to-depot inequalities for the SLTSP become

$$(x - y)(Q_H) + \sum_{d \in D \setminus D'} (1 - z_d)(x - y)(\overline{Q}_H^d) + y(H) \geq x(H) - |H| + z(D').$$

Similar inequalities were introduced in [13] for the the Location-Arc Routing Problem.

The formulation we propose for the SLTSP is:

$$
\begin{aligned}
(L2IF) \quad & \text{minimize} \quad \sum_{e\in E} c_e(x_e + y_e) && && (27)\\
& \text{subject to} \\
& (x+y)(\delta(d)) \geq 2z_d && d \in D \setminus V_R && (28)\\
& (x+y)(\delta(S)) \geq 2(1 - z(S\cap D)) && S \subseteq V : (S \cap V_R) \neq \emptyset && (29)\\
& (x-y)(\delta(S)\setminus H) + y(H) \geq x(H) - |H| + 1 \\
& && S \subset V,\ H \subseteq \delta(S), \\
& && |H| \text{ odd} && (30)\\
& (x-y)(Q_H) + \sum_{d\in D\setminus D'} (1-z_d)(x-y)(\overline{Q})^d_H + y(H) \geq x(H) - |H| + z(D') \\
& && S \subset V \setminus D, \emptyset \neq D' \subseteq D, \\
& && H \text{ odd-}D'\text{-cut for } S && (31)\\
& y_e \leq x_e && e \in E && (32)\\
& z(D) \leq p && && (33)\\
& x_e \in \{0,1\} && e \in E \setminus \gamma(D) && (34)\\
& y_e \in \{0,1\} && e \in E \setminus \gamma(D) && (35)\\
& z_d \in \{0,1\} && d \in D. && (36)
\end{aligned}
$$

Constraints (28) ensure that all open facilities are used. Constraints (29) and (30) are connectivity and parity inequalities, respectively. Note that the tight version of parity constraints (30) is valid, given that all y variables are binary. Constraints (31) are the return-to-depot inequalities discussed in Section 4.1. These constraints contain bilinear terms, which can be easily linearized as we will explain below. Constraints (32) impose that an edge cannot be traversed a second time unless it also has been traversed for the first time. The limit on the maximum number of facilities that can be opened is imposed by (33). Conditions of the variables x, y, and z derived from their definition are reflected in constraints (34), (35), and (36), respectively.

In order to integrate the set of constraints (31) within an ILP formulation it is necessary to linearize the bilinear terms that they include. For this we define additional decision variables representing the products $h_{ed} = (1 - z_d)(x_e - y_e)$ for the edges $e \in \delta(d)$, with $d \in D$. These variables will take the value 1 if and only if edge e, which is incident with potential facility d, is traversed exactly once and the facility located d is not open. The new set of variables h and variables x, y, and z are related throught the usual linearizing constraints:

$$
\begin{aligned}
& h_{ed} + z_d \leq 1 && d \in D, e \in \delta(d) && (37)\\
& h_{ed} + y_e \leq x_e && d \in D, e \in \delta(d) && (38)\\
& x_e \leq z_d + y_e + h_{ed} && d \in D, e \in \delta(d). && (39)
\end{aligned}
$$

4.2 Branch-and-cut solution algorithm for the SLTSP

The branch-and-cut algorithm developed for the SLTSP formulation follows the same idea and structure as the previous ones. Initially, the families of inequalities of exponential size, (29), (30), and (31) are relaxed, and only a small subset of constraints (29), (30) is considered, namely those associated with singletons, i.e., $S = \{u\}$, $u \in V_R \setminus D$ for (29), and those associated with singletons $S = \{u\}$, $u \in V$ and $H = \{e\} \in \delta(S)$ for (30). The separation of the connectivity constraints (29), which include location variables, can be easily adapted from the separation procedure for classical connectivity constraints. Details can be found, for instance, in [13]. The return-to-depot inequalities are again handled as lazy constraints and separated by inspection at integer nodes.

5 Computational experiments for SMDTSPs and SLTSP

We now present the results of the computational experiments we have conducted for the SMDTSP models and the SLTSP introduced in Sections 3 and 4, respectively. In order to assess the behavior of our solution algorithms we have used the sets of instances S1 and S2 with $r \in \{1/3, 2/3, 1\}$ used in our experiments for the STSP presented in Section 2.4. Below we describe how we have proceeded to select the sets of depots or potential depots for the different models we have tested. First, for each instance we randomly selected a subset of vertices $R \subseteq V$, with $|R| = 10$. Then, from this set R we randomly generated three subsets, $R7$, $R4$ and $R2$ successively smaller, i.e., $R2 \subset R4 \subset R7 \subset R$, with $|R2| = 2$, $|R4| = 4$ and $|R7| = 7$, which determined the sets of depots for the different SMDTSP data sets. The set R was taken as the set of potential locations for the facilities of the SLTSP instances. All computational experiments were run in the same environment and conditions as for the STSP experiments. As before, the maximum computing time for the solution of each instance was set to 7,200 seconds.

5.1 Steiner Multi-depot Traveling Salesman Problems

As already mentioned, for these experiments we have used the sets of instances S1 and S2 with $r \in \{1/3, 2/3, 1\}$ and, for each instance, we have considered the three sets of depots $R2$, $R4$, and $R7$ generated as described above, so the number of depots $|D| \in \{2, 4, 7\}$. The set of benchmark instances contains in total 180 data sets, 20 for each fixed value of r and $|D|$.

Tables 10, 11 and 12 summarize the results for the three SMDTSP models CA-SMDTSP, CO-SMDTSP and NO-SMDTSP, respectively. Results for each individual instance can be found in https://www.eio.upc.edu/en/homepages/elena/steiner. Each line in these tables presents results concerning the set S-r-$|D|$, which contains the 20 instances with the given values of r and $|D|$ and varying number of vertices. The columns under ♯ *Instances* indicate the total number of instances in the group solved to proven optimality within the allowed computing time. The entries in columns under *TimeLP* and Gap_0 respectively give average values, over all the instances in the corresponding S-r-$|D|$ set, of the computing times (in seconds), and the percentage optimality gaps at the root node of the enumeration tree, whereas the entries under *Time* and *Gap* give the same information at termination. The average number of nodes of the enumeration tree explored in each case is given under columns *Nodes*. Finally, column under ♯ D gives the average number of depots actually used in the obtained solutions.

Table 10: Average results for CA-SMDTSP

Instances	♯ Instances	Time LP	Gap_0	Time	Gap	Nodes	♯ D
S-1/3-2	20	5.98	5.06	341.75	0.00	16146.00	2.00
S-1/3-4	19	5.68	5.36	715.58	0.02	26374.80	4.00
S-1/3-7	18	4.70	5.38	920.67	0.05	64497.65	7.00
S-2/3-2	20	5.24	3.36	575.98	0.00	50078.20	2.00
S-2/3-4	20	6.63	3.49	396.65	0.00	43218.60	4.00
S-2/3-7	20	4.65	3.87	380.38	0.00	64497.65	7.00
S-3/3-2	18	7.48	1.92	1112.43	0.03	172389.50	2.00
S-3/3-4	18	7.66	2.06	1200.00	0.03	213170.75	4.00
S-3/3-7	17	5.72	2.26	1705.18	0.10	316756.45	7.00

As could be expected, the SMDSTP is for the same value of ratio r, more difficult to solve than the STSP. This is already reflected by the percentage optimality gaps at the root node, which, even if they are very moderate and on average usually below 5%, are larger for the SMDSTP than for the STSP.

Table 10 summarizes the performance of the solution algorithm for the proposed CA-SMDSTP formulation. As expected, for fixed values of the ratio r, an increase in the number of depots yields an increase in the difficulty for optimally solving the instances. The exception is ratio $r = 2/3$, for which the set of instances with the largest number of depots was the easiest to solve to optimality.

Table 11: Average results for the CO-SMDTSP

Instances	♯ Instances	Time LP	Gap_0	Time	Gap	Nodes	♯ D
S-1/3-2	20	5.37	5.37	371.72	0.00	20318.65	1.70
S-1/3-4	19	7.03	5.49	757.78	0.02	23365.65	3.15
S-1/3-7	19	6.50	5.29	657.18	0.04	26696.60	5.70
S-2/3-2	19	6.81	3.53	611.44	0.04	44689.60	2.00
S-2/3-4	19	6.29	3.53	607.71	0.03	40947.15	3.65
S-2/3-7	19	6.23	3.81	908.29	0.02	91938.85	6.20
S-3/3-2	19	8.16	1.90	849.19	0.03	135690.45	1.95
S-3/3-4	18	8.65	2.06	1456.13	0.04	229222.50	4.00
S-3/3-7	16	7.03	2.30	1940.96	0.09	280374.55	6.85

As shown in Table 11, a similar pattern of increasing difficulty as the number of depots rises also applies, in general, to the CO-SMDTSP model, the exception being now ratio $r = 1/3$ for which the most difficult instances correspond to set S-1/3-4 with the intermediate value $|D| = 4$. Nevertheless, the results of Table 12 for the NO-SMDTSP model indicate that the mentioned pattern now only holds for the ratio $r = 1$.

Table 12: Average results for the NO-SMDTSP

Instances	♯ Instances	Time LP	Gap_0	Time	Gap	Nodes	♯ D
S-1/3-2	20	8.70	4.81	414.13	0.00	15129.95	1.60
S-1/3-4	19	8.17	5.25	606.06	0.00	14950.90	3.05
S-1/3-7	20	6.57	4.84	182.04	0.00	15219.20	4.60
S-2/3-2	20	9.75	3.21	397.83	0.00	29882.80	1.80
S-2/3-4	20	7.20	3.21	359.70	0.00	27235.85	3.25
S-2/3-7	20	6.91	3.22	128.37	0.00	17053.75	4.95
S-3/3-2	20	14.81	1.80	773.33	0.00	118712.45	1.70
S-3/3-4	20	18.81	1.92	834.90	0.00	129084.10	3.15
S-3/3-7	19	8.12	1.83	901.10	0.01	188664.05	4.70

The results of the previous tables show that the proposed solution algorithms perform well for all three models, solving to optimality at least 93% of the tested instances within reasonable computing times. However, a comparison of the models reveals that NO-SMDTSP produces the smallest percentage optimality gaps at the root node. This can be explained by the fact that both sets of variables in that model, x and y, are binary, allowing the tight version of parity constraints, (30). This effect can also be observed in Table 13, which presents average results on the number of cuts of each type generated at the root node and during the exploration of the enumeration tree, over each ratio r and each of the models. While the solution algorithms for CA-SMDTSP and CO-SMDTSP mostly add connectivity constrains at the root node, the solution algorithm for NO-SMDTSP basically reinforces its LP bound by only adding parity constraints (30). Table 13 shows that the number of cuts generated in the exploration of the enumeration tree is in general quite moderate, with more parity than other type of cuts. Furthermore, we can also see that not many return-to-depot cuts (Ret D) are needed.

Table 13: Summary of generated cuts for the SMDTSP

Instances	Model	Root node			Enumeration tree		
		Connec	Parity	Ret D	Connec	Parity	Ret D
S-1/3	CA	104.86	38.36	0.16	364.47	1783.73	8.61
	CO	103.46	39.53	0.10	385.38	1898.58	7.60
	NO	82.57	154.47	0.00	265.44	1661.64	0.00
S-2/3	CA	100.52	44.16	0.08	213.42	1793.23	13.87
	CO	86.95	40.75	0.09	197.10	1812.19	10.81
	NO	67.03	175.94	0.00	138.30	1597.13	0.00
S-3/3	CA	73.42	56.70	0.11	87.16	1541.81	12.52
	CO	73.23	91.84	0.11	91.77	1613.05	12.52
	NO	54.16	373.76	0.00	63.78	1163.32	0.00

Columns $\sharp D$ in Tables 10–12 provide information on the structure of the optimal solutions produced by the different models. Not surprisingly, the number of depots actually used in optimal solutions decreases as the flexibility, in terms of the consistency requirements of the routes, increases. While in CA-SMDTSP all depots must be used, the solutions to CO-SMDTSP and NO-SMDTSP models use fewer and fewer depots, respectively. In both cases, the highest reduction, relative to CA-SMDTSP, in the number of depots actually used is attained for instances with ratio $r = 1/3$, while the smallest reduction corresponds to $r = 1$ and $r = 2/3$ for models CO-SMDTSP and NO-SMDTSP, respectively.

Finally, the results of Table 14 allow us to compare the optimal values of all three models. The entries of this table give average values, over all the instances in each set S-r-$|D|$, with fixed r (rows) and $|D|$ (columns), of the percentages of the ratios of the optimal values of model `mod` over the optimal values of the most restrictive model CA-SMDTSP, i.e., $100\, v(\texttt{mod})/v(CA - SMDTSP)$. Of course, all values are 100 when `mod` is CA-SMDTSP. As anticipated from Proposition 3.1, it can be clearly seen that $v(CA - SMDTSP) \geq v(CO - SMDTSP) \geq v(NO - SMDTSP)$. In general, the decrease in the optimal values as the models become more relaxed are very moderate, although some exceptions can be found. The largest reduction corresponds to instance $25 - 1/3 - 4$ where both CO-SMDTSP and NO-SMDTSP produce a solution whose value is 75.26% of the solution produced by CA-SMDTSP, by removing one depot.

Table 14: Summary of objective function values percentatges

Instances	Model	$\|D\|$		
		2	4	7
S-1/3	CA	100.00	100.00	100.00
	CO	99.88	98.21	98.73
	NO	99.83	98.15	98.48
S-2/3	CA	100.00	100.00	100.00
	CO	100.00	99.00	99.71
	NO	99.79	98.97	98.99
S-3/3	CA	100.00	100.00	100.00
	CO	99.96	100.00	99.99
	NO	99.51	99.25	99.50

5.2 Steiner Location Traveling Salesman Problem

In order to analyze the performance of the algorithm for the SLTSP formulation the same sets of instances as before were used. However, in this case the size of the set of potential locations was fixed to $|R| = 10$, and three diferent values of p were tested for the maximum number of facilities to be located: $p \in \{2, 4, 10\}$. We now denote by *S-r-p* to the set of 20 instances with ratio r and maximum number of open facilities p.

Table 15 summarizes the results of the SLSTP, for which 94% of the instances were

Table 15: Average results for the SLTSP

Instances	♯ Instances	Time LP	Gap_0	Time	Gap	Nodes	$\sum z_d^*$
S-1/3-2	18	87.94	6.51	1569.28	0.07	37191.45	2.00
S-1/3-4	20	68.03	4.29	2868.11	0.00	19797.00	3.95
S-1/3-10	20	52.36	3.20	194.98	0.00	10476.05	5.85
S-2/3-2	17	29.47	3.58	1666.71	0.08	103523.10	2.00
S-2/3-4	20	37.45	2.50	785.62	0.00	112542.60	3.90
S-2/3-10	20	24.08	1.96	153.76	0.00	19171.15	6.55
S-3/3-2	18	11.38	2.36	1569.99	0.02	371747.75	2.00
S-3/3-4	17	11.34	1.87	1748.60	0.03	527214.20	3.85
S-3/3-10	20	7.00	1.55	416.31	0.00	91180.50	6.50

optimally solved. A surprising outcome from the obtained results is that in general, the SLSTP is easier to solve as the maximum number of p open facilities increases. In comparison with the two multi-depot models CO-SMDTSP and NO-SMDTSP which somehow involve location decisions, this behavior is different from that of CO-SMDTSP, whereas it seems in accordance with the results produced by NO-SMDTSP for $r = 1/3$ and $r = 2/3$.

Similarly to previous models the percentage optimality gaps at the root node are nearly always smaller than 5%, and these gaps tend to decrease as the value of the ratio r increases. As can be observed from column $\sum z_d^*$, the number of facilities that are opened in optimal solutions is exactly p for the smallest value $p = 2$, nearly always p for the intermediate value $p = 4$, but notably smaller than p for the largest value $p = 10$. These results are consistent with those of the *Optional-depots* SMDSTP models. Note, however, that even if at most p facilities can be opened in the SLTSP, the set R of candidate facilities contains 10 elements. Hence, there is a quite wide range of possibilities as for the potential combinations of p plants to open. This contrasts with respect to CO-SMDTSP and NO-SMDTSP where the set of *candidate depots* contains exactly $|D|$ elements. This also justifies that, in general, the number of open facilities in optimal SLTSP solutions (column $\sum z_d^*$ in Table 15) tends to be higher than the number of depots actually used in optimal solutions to CO-SMDTSP and NO-SMDTSP (columns $\sharp D$ in Tables 11 and 12).

Table 16: Summary of generated cuts for the SLTSP

Instances	Root node				Enumeration tree			
	Connec	Parity	Dead-end	Ret D	Connec	Parity	Dead-end	Ret D
S-1/3	162.86	9.32	228.37	0.03	1460.02	75.28	318.37	10.64
S-2/3	167.85	13.25	84.87	0.41	1460.15	116.26	129.20	11.90
S-3/3	205.56	17.99	0.00	0.78	297.35	127.98	0.00	14.47

The average number of cuts of each type generated at the root node and in the exploration of the enumeration tree are presented in Table 16. The entries of this table indicate that, similarly to the SMDTSP models, the number of cuts generated for solving SLTSP instances is in general quite moderate. However, we can observe that the SLTSP requires more connectivity cuts than of any other type of cuts, as was the case for the

STSP. Furthermore, some dead-end inequalities are again generated.

6 Conclusions

We have introduced, modeled and solved the Steiner Traveling Salesman Problem (STSP) and we have presented some new extensions that consider several depots or locational decisions. All considered problems are defined on incomplete undirected graphs and service demand is placed at the vertices of the network, although not all the vertices have to be necessarily visited. An integer linear programming formulation was proposed for each problem, where the parity conditions are modeled using cocircuit inequalities. While such constraints are now quite common for ARPs, they are still rarely used to model the parity of the visited vertices in node-routing problems. All formulations make use of two-index decision variables. This implies that for the problems that consider several depots or potential facilities specific return-to-depot constraints may be needed to guarantee that routes are well defined. An exact branch-and-cut algorithm was developed for each of the presented formulations in which the families of constraints of exponential size are separated. The numerical results of our computational experiments outperform existing results for the STSP. Moreover, the obtained results indicate that the modeling techniques that we have applied are also very effective for dealing with the proposed extensions. All solution algorithms are capable of solving instances involving up to 500 nodes within reasonable computing times.

Acknowledgements

This research was partially supported by the Spanish Ministry of Economy and Competitiveness through grants EEBB-I-16-10670, MTM2015-68097-P, and MTM2015-63779-R (MINECO/FEDER), and by the Canadian Natural Sciences and Engineering Research Council under the grant 2015-06189. This support is gratefully acknowledged. Thanks are due to the referees for their valuable comments.

References

[1] J. Aráoz, E. Fernández, and C. Franquesa. The clustered prize-collecting arc routing problem. *Transportation Science*, 43(3):287–300, 2009.

[2] J. Aráoz, E. Fernández, and C. Franquesa. The generalized arc routing problem. *TOP*, 25(3):497–525, 2017.

[3] J. Aráoz, E. Fernández, and O. Meza. Solving the prize-collecting rural postman problem. *European Journal of Operational Research*, 196(3):886–896, 2009.

[4] F. Barahona and M. Grötschel. On the cycle polytope of a binary matroid. *Journal of Combinatorial Theory, Series B*, 40(1):40–62, 1986.

[5] J. M. Belenguer and E. Benavent. The capacitated arc routing problem: Valid inequalities and facets. *Computational Optimization and Applications*, 10(2):165–187, 1998.

[6] J. M. Belenguer, E. Benavent, C. Prins, C. Prodhon, and R. Wolfler Calvo. A branch-and-cut method for the capacitated location-routing problem. *Computers and Operations Research*, 38(6):931–941, 2011.

[7] N. Christofides, V. Campos, Á. Corberán, and E. Mota. An algorithm for the rural postman problem. Technical report, Imperial College Report IC.O.R.81.5, 1981.

[8] A. Corberán, A. N. Letchford, and J. M. Sanchis. A cutting plane algorithm for the general routing problem. *Mathematical programming*, 90(2):291–316, 2001.

[9] Á. Corberán and J. M. Sanchis. The general routing problem polyhedron: Facets from the RPP and GTSP polyhedra. *European Journal of Operational Research*, 108(3):538–550, 1998.

[10] G. Cornuéjols, J. Fonlupt, and D. Naddef. The traveling salesman problem on a graph and some related integer polyhedra. *Mathematical Programming*, 33(1):1–27, 1985.

[11] J. Dongarra. Performance of various computers using standard linear equations software. *Computer Architecture News*, 20, 11 2002.

[12] E. Fernández, G. Laporte, and J. Rodríguez-Pereira. A branch-and-cut algorithm for the multi-depot rural postman problem. *Transportation Science*, 52(2):353–369, 2018.

[13] E. Fernández, G. Laporte, and J. Rodríguez-Pereira. Exact solution of several families of location-arc routing problems. *Transportation Science*, to appear 2018.

[14] E. Fernández and J. Rodríguez-Pereira. Multi-depot rural postman problems. *TOP*, 25(2):340–372, 2017.

[15] B. Fleischmann. A cutting plane procedure for the travelling salesman problem on road networks. *European Journal of Operational Research*, 21(3):307–317, 1985.

[16] G. Ghiani and G. Laporte. A branch-and-cut algorithm for the undirected rural postman problem. *Mathematical Programming*, 87(3):467–481, 2000.

[17] W.W. Hardgrave and G.L. Nemhauser. On the relation between the traveling-salesman and the longest-path problems. *Operations Research*, 10(5):647–657, 1962.

[18] A. N. Letchford. New inequalities for the general routing problem. *European Journal of Operational Research*, 96:317–322, 1997.

[19] A. N. Letchford. The general routing polyhedron: A unifying framework. *European Journal of Operational Research*, 112:122–133, 1999.

[20] A. N. Letchford, S.D. Nasiri, and D. O. Theis. Compact formulations of the Steiner traveling salesman problem and related problems. *European Journal of Operational Research*, 228(1):83–92, 2013.

[21] A. N. Letchford and J. J. Salazar-González. Stronger multi-commodity flow formulations of the capacitated vehicle routing problem. *European Journal of Operational Research*, 244(3):730–738, 2015.

[22] P. Miliotis, G. Laporte, and Y. Nobert. Computational comparison of two methods for finding the shortest complete cycle or circuit in a graph. *RAIRO-Operation Research*, 15(3):233–239, 1981.

[23] C. S. Orloff. A fundamental problem in vehicle routing. *Networks*, 4(1):35–64, 1974.

[24] M.W. Padberg and M. Grötschel. Polyhedral computations. In *The Traveling Saliesman Prolbem: A Guided Tour of Combinatorial Optimization*, pages 307–360. E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys (editons), Wiley, Chichester, 1985.

[25] G. Reinelt and D.O. Theis. Transformation of facets of the general routing problem polytope. *SIAM Journal on Optimization*, 16(1):220–234, 2005.

[26] Standard Performance Evaluation Corporation (SPEC). https://www.spec.org/, December 2018.