European Journal of Operational Research 228 (2013) 83–92

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Compact formulations of the Steiner Traveling Salesman Problem and related problems

Adam N. Letchford [a,*], Saeideh D. Nasiri [b], Dirk Oliver Theis [c]

[a] *Department of Management Science, Lancaster University, Lancaster LA1 4YX, United Kingdom*
[b] *STOR-i Doctoral Training Center, Lancaster University, Lancaster LA1 4YF, United Kingdom*
[c] *Faculty of Mathematics, Otto von Guericke University of Magdeburg, Germany*

ARTICLE INFO

Article history:
Received 16 March 2012
Accepted 25 January 2013
Available online 9 February 2013

Keywords:
Traveling salesman problem
Integer programming
Combinatorial optimization

ABSTRACT

The Steiner Traveling Salesman Problem (STSP) is a variant of the TSP that is particularly suitable when routing on real-life road networks. The standard integer programming formulations of both the TSP and STSP have an exponential number of constraints. On the other hand, several *compact* formulations of the TSP, i.e., formulations of polynomial size, are known. In this paper, we adapt some of them to the STSP, and compare them both theoretically and computationally. It turns out that, just by putting the best of the formulations into the CPLEX branch-and-bound solver, one can solve instances with over 200 nodes. We also briefly discuss the adaptation of our formulations to some related problems.

1. Introduction

The *Traveling Salesman Problem* (TSP), in its undirected version, can be defined as follows. We are given a complete undirected graph $G = (V, E)$ and a positive integer cost c_e for each edge $e \in E$. The task is to find a Hamiltonian circuit, or *tour*, of minimum total cost. The best algorithms for solving the TSP to proven optimality, such as the ones described in [1,26,29], are based on a formulation of the TSP as a 0–1 linear program due to Dantzig et al. [7], which we present in Section 2.1 of this paper.

The Dantzig et al. formulation has only one variable per edge, but has an exponentially-large number of constraints, which makes cutting-plane methods necessary (see again [1,26,29]). If one wishes to avoid this complication, one can instead use a so-called *compact* formulation of the TSP, i.e., a formulation with a polynomial number of both variables and constraints. A variety of compact formulations are available (see the surveys [18,27,30] and also Section 2.2 of this paper).

When dealing with routing problems on real-life road networks, however, one is much more likely to encounter the following variant of the TSP. We are given a connected undirected graph $G = (V, E)$, a positive integer cost c_e for each $e \in E$, and a set $V_R \subseteq V$ of *required* nodes. The task is to find a minimum-cost closed walk, not necessarily Hamiltonian, that visits each required node at least once. Nodes may be visited more than once if desired, and edges may be traversed more than once if desired. This variant of the TSP was proposed, apparently independently, by three sets of authors [6,13,28]. (The special case in which all nodes must be visited was considered earlier in [19,24].) We will follow Cornuéjols et al. [6] in calling this variant the *Steiner* TSP, or STSP for short.

As noted in [6,13], it is possible to convert any instance of the STSP into an instance of the standard TSP, by computing shortest paths between every pair of required nodes. So, in principle, one could use any of the above-mentioned TSP formulations to solve the STSP. If, however, the original STSP instance is defined on a sparse graph, the conversion to a standard TSP instance increases the number of variables substantially, which is undesirable. So, in this paper, we provide and analyse some compact formulations for the STSP. Specifically, we introduce single-commodity flow, multi-commodity flow and time-staged formulations, and compare them both theoretically and computationally.

It turns out that, just by putting the best of our formulations into the CPLEX branch-and-bound solver, one can routinely solve instances with over 200 nodes to proven optimality. This is in sharp contrast to the situation with the standard TSP, where sophisticated cutting-plane techniques are needed to solve instances with more than about 50 nodes (see, e.g., [1,26]).

The paper is structured as follows. The relevant literature is reviewed in Section 2. In Section 3, the so-called *commodity-flow* formulations of the TSP are adapted to the Steiner case. In Section 4, the same is done for the so-called *time-staged* formulation. In Section 5, some computational results are given. Then, in Section 6, we briefly discuss the possibility of adapting our approach to other variants of the TSP. Finally, some concluding remarks appear in Section 7.

[*] Corresponding author. Tel.: +44 1524 594719.
E-mail addresses: A.N.Letchford@lancaster.ac.uk (A.N. Letchford), s.d.nasiri@lancaster.ac.uk (S.D. Nasiri), theis@ovgu.de (D.O. Theis).

0377-2217/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2013.01.044

2. Literature review

We now review the relevant literature. We cover the classical formulation of the standard TSP in Section 2.1, compact formulations of the standard TSP in Section 2.2, and the classical formulation of the STSP in Section 2.3.

2.1. The classical formulation of the standard TSP

The classical and most commonly-used formulation of the standard TSP is the following one, due to Dantzig et al. [7]:

$$\min \sum_{e \in E} c_e x_e$$

$$\text{s.t.} \sum_{e \in \delta(\{i\})} x_e = 2 \quad (\forall i \in V) \tag{1}$$

$$\sum_{e \in \delta(S)} x_e \geqslant 2 \quad (\forall S \subseteq V : 2 \leqslant |S| \leqslant |V|/2) \tag{2}$$

$$x_e \in \{0,1\} \quad (\forall e \in E).$$

Here, x_e is a binary variable, taking the value 1 if and only if the edge e belongs to the tour, and, for any $S \subset V$, $\delta(S)$ denotes the set of edges having exactly one end-node inside S. The constraints (1), called *degree* constraints, enforce that the tour uses exactly two of the edges incident on each node. The constraints (2), called *subtour elimination constraints*, ensure that the tour is connected.

We will call this formulation the *DFJ* formulation. A key feature of this formulation is that the subtour elimination constraints (2) are exponential in number.

2.2. Compact formulations of the standard TSP

As mentioned above, a wide variety of compact formulations exist for the standard TSP, and there are several surveys available (e.g., [18,27,30]). For the sake of brevity, we mention here only four of them. All of them start by setting $V = \{1,2,\ldots,n\}$ and viewing node 1 as a 'depot', which the salesman must leave at the start of the tour and return to at the end of the tour. Moreover, all of them can be used for the asymmetric TSP as well as for the standard (symmetric) TSP.

We begin with the formulation of Miller et al. [25], which we call the *MTZ* formulation. For all node pairs (i,j), let $\tilde{x}_{ij}$ be a binary variable, taking the value 1 if and only if the salesman travels from node i to node j. Also, for $i = 2,\ldots,n$, let u_i be a continuous variable representing the position of node i in the tour. (The depot can be thought of as being at positions 0 and n.) The MTZ formulation is then:

$$\min \sum_{i,j=1}^{n} c_{ij} \tilde{x}_{ij} \tag{3}$$

$$\text{s.t.} \sum_{j=1}^{n} \tilde{x}_{ji} = 1 \quad (1 \leqslant i \leqslant n) \tag{4}$$

$$\sum_{j=1}^{n} \tilde{x}_{ij} = 1 \quad (1 \leqslant i \leqslant n) \tag{5}$$

$$\tilde{x}_{ij} \in \{0,1\} \quad (1 \leqslant i,j \leqslant n; i \neq j) \tag{6}$$

$$u_i - u_j + (n-1)\tilde{x}_{ij} \leqslant n-2 \quad (2 \leqslant i,j \leqslant n; i \neq j) \tag{7}$$

$$1 \leqslant u_i \leqslant n-1 \quad (2 \leqslant i \leqslant n). \tag{8}$$

The constraints (4) and (5) ensure that the salesman arrives at and departs from each node exactly once. The constraints (7) ensure that, if the salesman travels from i to j, then the position of node j is one more than that of node i. Together with the bounds (8), this ensures that each non-depot node is in a unique position.

The MTZ formulation is compact, having only $\mathcal{O}(n^2)$ variables and $\mathcal{O}(n^2)$ constraints. Unfortunately, Padberg and Sung [30] show that its LP relaxation yields an extremely weak lower bound, much weaker than that of the DFJ formulation.

The next compact formulation, historically, was the 'time-staged' (TS) formulation proposed by both Vajda [31] and Houck et al. [21] independently. For all $1 \leqslant i,j,\ k \leqslant n$ with $i \neq j$, let r^k_{ij} be a binary variable taking the value 1 if and only if the edge $\{i,j\}$ is the kth edge to be traversed in the tour, and is traversed in the direction going from i to j. We then have:

$$\min \sum_{i=2}^{n} c_{1i} r^1_{1i} + \sum_{k=2}^{n-1} \sum_{i,j=2}^{n} c_{ij} r^k_{ij} + \sum_{i=2}^{n} c_{i1} r^n_{i1}$$

$$\text{s.t.} \sum_{j=2}^{n} r^1_{1j} = 1 \tag{9}$$

$$\sum_{j=2}^{n} r^n_{j1} = 1 \tag{10}$$

$$\sum_{k=1}^{n-1} \sum_{j \neq i} r^k_{ji} = 1 \quad (2 \leqslant I \leqslant n) \tag{11}$$

$$\sum_{j \neq i} r^k_{ji} = \sum_{j \neq i} r^{k+1}_{ij} \quad (2 \leqslant i \leqslant n; 1 \leqslant k \leqslant n-1) \tag{12}$$

$$r^k_{ij} \in \{0,1\} \quad (1 \leqslant i,j,k \leqslant n; i \neq j).$$

The constraints (9) and (10) state that the salesman must leave the depot at the start of the tour and return to it at the end. The constraints (11) ensure that the salesman arrives at each non-depot node exactly once, and the constraints (12) ensure that the salesman departs from each node that he visits.

The TS formulation has $\mathcal{O}(n^3)$ variables and $\mathcal{O}(n^2)$ constraints. It follows from results in [18,30] that the associated lower bound is intermediate in strength between the MTZ and DFJ bounds.

Next, we mention the *single-commodity flow* (SCF) formulation of Gavish and Graves [16]. Imagine that the salesman carries $n-1$ units of a commodity when he leaves node 1, and delivers 1 unit of this commodity to each other node. Let the $\tilde{x}_{ij}$ variables be defined as above, and define additional continuous variables g_{ij}, representing the amount of the commodity (if any) passing directly from node i to node j. The formulation then consists of the objective function (3), the constraints (4)–(6), and the following constraints:

$$\sum_{j=1}^{n} g_{ji} - \sum_{j=2}^{n} g_{ij} = 1 \quad (2 \leqslant i \leqslant n) \tag{13}$$

$$0 \leqslant g_{ij} \leqslant (n-1)\tilde{x}_{ij} \quad (1 \leqslant i,j \leqslant n; j \neq i). \tag{14}$$

The constraints (13) ensure that one unit of the commodity is delivered to each non-depot node. The bounds (14) ensure that the commodity can flow only along edges that are in the tour.

The SCF formulation has $\mathcal{O}(n^2)$ variables and $\mathcal{O}(n)$ constraints. It is proved in [30] that the associated lower bound is intermediate in strength between the MTZ and DFJ bounds. Later on, in [18], it was shown that it is in fact intermediate in strength between the MTZ and TS bounds.

Finally, we mention the *multi-commodity flow* (MCF) formulation of Claus [5]. Here, we imagine that the salesman carries $n-1$ commodities, one unit of each for each customer. Let the $\tilde{x}_{ij}$ variables be defined as above. Also define, for all $1 \leqslant i,\ j \leqslant n$ with $i \neq j$ and all $2 \leqslant k \leqslant n$, the additional continuous variable f^k_{ij}, representing the amount of the kth commodity (if any) passing directly from node i to node j. The formulation then consists of the objective function (3), the constraints (4)–(6), and the following constraints:

$$0 \leqslant f_{ij}^k \leqslant \tilde{x}_{ij} \quad (k = 2, \ldots, n; \{i,j\} \subset \{1, \ldots, n\}) \tag{15}$$

$$\sum_{i=2}^{n} f_{1i}^k = 1 \quad (k = 2, \ldots, n) \tag{16}$$

$$\sum_{i=1}^{n} f_{ik}^k = 1 \quad (k = 2, \ldots, n) \tag{17}$$

$$\sum_{i=1}^{n} f_{ij}^k - \sum_{i=2}^{n} f_{ji}^k = 0 \quad (k = 2, \ldots, n; j \in \{2, \ldots, n\} \setminus \{k\}). \tag{18}$$

The constraints (15) state that a commodity cannot flow along an edge unless that edge belongs to the tour. The constraints (16) and (17) impose that each commodity leaves the depot and arrives at its destination. The constraints (18) ensure that, when a commodity arrives at a node that is not its final destination, then it also leaves that node.

The MCF formulation has $\mathcal{O}(n^3)$ variables and $\mathcal{O}(n^3)$ constraints. It is proved in [30] that the associated lower bound is equal to the DFJ bound. Therefore, this is the strongest of the four compact formulations mentioned.

2.3. The classical formulation of the STSP

The integer programming formulation given in [13] for the STSP is as follows:

$$\min \quad \sum_{e \in E} c_e x_e \tag{19}$$

$$\text{s.t.} \quad \sum_{e \in \delta(i)} x_e \text{ even} \quad (i \in V) \tag{20}$$

$$\sum_{e \in \delta(S)} x_e \geqslant 2 \quad (S \subset V : S \cap V_R \neq \emptyset, V_R \setminus S \neq \emptyset) \tag{21}$$

$$x_e \in \mathbb{Z}_+ \quad (e \in E). \tag{22}$$

Note that, since edges may be traversed more than once if desired, the x variables are now general integers. Also, since nodes may be visited more than once if desired, the degree constraints (20) specify only that the degrees must be even. (Although these constraints are non-linear, they can be easily linearised, using one additional variable for each node.) The crucial point, however, is that the so-called *connectivity constraints* (21) are exponential in number.

3. Flow-based formulations of the STSP

In this section, we adapt the formulations SCF and MCF, mentioned in Section 2.2, to the Steiner case. We also give some results concerned with the strength of the LP relaxations of our formulations.

3.1. Some preliminaries

At this point, we present some additional notation. Let $\widetilde{G} = (V,A)$ be a directed graph, where the set of directed arcs A is obtained from the edge set E by replacing each edge $\{i,j\}$ with two directed arcs (i,j) and (j,i). For each arc $a \in A$, the cost c_a is viewed as being equal to the cost of the corresponding edge. For any node set $S \subset V$, let $\delta^+(S)$ denote the set of arcs in A whose tail is in S and whose head is in $V \backslash S$, and let $\delta^-(S)$ denote the set of arcs in A for which the reverse holds. For readability, we write $\delta^+(i)$ and $\delta^-(i)$ in place of $\delta^+(\{i\})$ and $\delta^-(\{i\})$, respectively. Finally, let $n_R = |V_R|$ denote the number of required nodes.

We will find the following lemma useful:

Lemma 1. *In an optimal solution to the STSP, no edge will be traversed more than once in either direction.*

This lemma is part of the folklore, but an explicit proof can be found in the appendix of [23]. An immediate consequence is that one can define a binary variable $\tilde{x}_a$ for each arc $a \in A$, taking the value 1 if and only if the salesman travels along a.

We will also need the following classical result from network flow theory, due to Gale [15] and Hoffman [20]:

Theorem 1 (*Gale–Hoffman*). *Let $\widetilde{G} = (V,A)$ be a directed graph, $s \in V$ be a source node, $d \in \mathbb{Q}_+^{|V|}$ be a demand vector with $d_s = 0$, and $\ell, u \in \mathbb{Q}_+^{|A|}$ be lower and upper bound vectors with $\ell \leqslant u$. There exists a feasible flow $f \in \mathbb{Q}_+^{|A|}$ satisfying (i) $\ell \leqslant f \leqslant u$ and (ii) $\sum_{a \in \delta^-(i)} f_a = d_i + \sum_{a \in \delta^+(i)} f_a$ for all $i \in V$ if and only if:*

$$\sum_{a \in \delta^-(S)} u_a \geqslant \sum_{i \in S} d_i + \sum_{a \in \delta^+(S)} \ell_a$$

holds for all $S \subset V \backslash \{s\}$.

3.2. An initial single-commodity flow formulation

Without loss of generality, assume that node 1 is required. By analogy with the case of the standard TSP, we imagine that the salesman departs the depot with $n_R - 1$ units of the commodity, and delivers one unit of that commodity to each required node. So, for each arc $a \in A$, let the new variable f_a represent the amount of the commodity passing through a. The single-commodity flow formulation (SCF) may then be adapted to the sparse graph setting as follows:

$$\min \quad \sum_{a \in A} c_a \tilde{x}_a \tag{23}$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(i)} \tilde{x}_a \geqslant 1 \quad (\forall i \in V_R) \tag{24}$$

$$\sum_{a \in \delta^+(i)} \tilde{x}_a = \sum_{a \in \delta^-(i)} \tilde{x}_a \quad (\forall i \in V) \tag{25}$$

$$\sum_{a \in \delta^-(i)} f_a - \sum_{a \in \delta^+(i)} f_a = 1 \quad (\forall i \in V_R \setminus \{1\}) \tag{26}$$

$$\sum_{a \in \delta^-(i)} f_a - \sum_{a \in \delta^+(i)} f_a = 0 \quad (\forall i \in V \setminus V_R) \tag{27}$$

$$0 \leqslant f_a \leqslant (n_R - 1)\tilde{x}_a \quad (\forall a \in A) \tag{28}$$

$$\tilde{x}_a \in \{0,1\} \quad (\forall a \in A). \tag{29}$$

The constraints (24) ensure that the salesman departs from each required node at least once, and the constraints (25) ensure that the salesman departs from each node as many times as he arrives. The constraints (26) impose that one unit of the commodity is delivered to each required node, and the constraints (27) ensure that the amount of commodity on board when leaving a non-required node is equal to the amount when arriving. The bounds (28) ensure that, if any of the commodity passes along an arc, then that arc appears in the tour.

In what follows, we refer to this SCF formulation as 'SCF1'. Note that it contains $\mathcal{O}(|E|)$ variables and $\mathcal{O}(|E|)$ constraints.

The following theorem characterises the projection of the LP relaxation of SCF1 into the space of the x variables:

Theorem 2. *Let P^1 be the polytope in $(\tilde{x}, f)$-space defined by the constraints (24)–(28) and the trivial bounds $\tilde{x} \in [0,1]^{|A|}$, let P^2 be the projection of P^1 into $\tilde{x}$-space, and let P^3 be the projection of P^2 into x-space, under the linear mapping $x_{ij} = \tilde{x}_{ij} + \tilde{x}_{ji}$ for all $\{i,j\} \in E$. Then P^3 is completely described by the following linear inequalities:*

$$\sum_{e \in \delta(i)} x_e \geqslant 2 \quad (i \in V_R) \tag{30}$$

$$\sum_{e \in \delta(S)} x_e \geqslant 2\frac{|S \cap V_R|}{n_R - 1} \quad (\forall S \subset V \setminus \{1\} : S \cap V_R \neq \emptyset) \tag{31}$$

$$x \in [0,2]^{|E|}. \tag{32}$$

Proof. Setting $\ell_a = 0$ and $u_a = (n_R - 1)\tilde{x}_a$ for all $a \in A$, $s = 1$, $d_i = 0$ for all $i \in V \backslash V_R$ and $d_i = 1$ for all $i \in V_R \backslash \{1\}$ in Theorem 1, we see that P^2 is described by the constraints (24) and (25), the trivial bounds $\tilde{x} \in [0,1]^{|A|}$, and the inequalities

$$(n_R - 1) \sum_{a \in \delta^+(S)} \tilde{x}_a \geqslant |S \cap V_R| \qquad (33)$$

for all $S \subset V \backslash \{1\}$ with $S \cap V_R \neq \emptyset$. Then, using (25), we can re-write (24) as

$$\sum_{a \in \delta^+(i) \cup \delta^-(i)} \tilde{x}_a \geqslant 2 \quad (\forall i \in V_R)$$

and re-write the inequalities (33) in the form:

$$(n_R - 1) \sum_{a \in \delta^+(S) \cup \delta^-(S)} \tilde{x}_a \geqslant 2|S \cap V_R|.$$

The result then follows from the stated mapping of $\tilde{x}$ onto x. □

A consequence of Theorem 2 is the following.

Corollary 1. *The lower bound from the LP relaxation of SCF1 is never stronger than the lower bound from Fleischmann's formulation (19)–(22).*

Proof. The inequalities (30) are a special case of the inequalities (21), and the inequalities (31) are weaker than the inequalities (21). Moreover, an optimal solution to the LP relaxation of Fleischmann's formulation will always satisfy $x_e \leqslant 2$ for all $e \in E$. □

3.3. Strengthened single-commodity flow formulation

It is possible to strengthen SCF1 using the following argument. One can assume that, if any required node is visited more than once by the salesman, then the commodity is delivered on the first visit. Accordingly, for each node $i \in V \backslash \{1\}$, let r_i be the minimum number of required nodes (not including the depot) that the salesman must have visited when he leaves i for the first time. Also, by convention, let $r_1 = 0$. (Note that one can compute r_i for all $i \in V \backslash \{1\}$ efficiently, using Dijkstra's single-source shortest-path algorithm [8]). Now, the constraints (28) can be replaced with the following stronger constraints:

$$0 \leqslant f_{ij} \leqslant (n_R - r_i - 1)\tilde{x}_{ij} \quad (\forall (i,j) \in A). \qquad (34)$$

We call this strengthened SCF formulation 'SCF2'. We do not have a complete characterisation of the projection of the LP relaxation of SCF2 into the space of the x variables, but the following theorem gives a partial description:

Theorem 3. *Let P^1 be the polytope in $(\tilde{x}, f)$-space defined by the constraints (24)–(27), the strengthened constraints (34), and the trivial bounds $\tilde{x} \in [0,1]^{|A|}$. Let P^2 be the projection of P^1 into $\tilde{x}$-space, and let P^3 be the projection of P^2 into x-space. Also, for any set $S \subseteq V \backslash \{1\}$ such that $S \cap V_R \neq \emptyset$, let T(S) be the set of all nodes that are not in S but are adjacent to at least one node in S. Finally, define $L(S) = \min_{i \in T(S)} r_i$ and $U(S) = \max_{i \in T(S)} r_i$. Then P^3 satisfies the inequalities (30) and (32), together with the following inequality for all such sets S and for $k = L(S), \ldots, U(S)$:*

$$(n_R - k - 1) \sum_{e \in \delta(S)} x_e + 2 \sum_{\{i,j\} \in \delta(S): j \in S} \max\{0, k - r_i\} x_{ij} \geqslant 2|S \cap V_R|. \qquad (35)$$

Proof. The fact that P^3 satisfies (30) and (32) follows from Theorem 2 and the fact that SCF2 dominates SCF1. Now, setting $\ell_{ij} = 0$ and $u_{ij} = (n_R - r_i - 1)\tilde{x}_{ij}$ for all $(i,j) \in A$, $s = 1$, $d_i = 0$ for all $i \in V \backslash V_R$ and $d_i = 1$ for all $i \in V_R \backslash \{1\}$ in Theorem 1, we see that P^2 is described by the constraints (24) and (25), the trivial bounds $\tilde{x} \in [0,1]^{|A|}$, and the inequalities

$$\sum_{(i,j) \in \delta^-(S)} (n_R - r_i - 1)\tilde{x}_{ij} \geqslant |S \cap V_R|$$

for all $S \subset V \backslash \{1\}$ with $S \cap V_R \neq \emptyset$. Together with non-negativity on $\tilde{x}$ and the Eq. (25), this implies that P^2 satisfies

$$(n_R - k - 1) \sum_{(i,j) \in \delta^+(S) \cup \delta^-(S)} \tilde{x}_{ij} + 2 \sum_{(i,j) \in \delta^-(S)} \max\{0, k - r_i\}(\tilde{x}_{ij} + \tilde{x}_{ji})$$
$$\geqslant 2|S \cap V_R|$$

for all such S. The result then follows from the mapping of $\tilde{x}$ onto x. □

Note that, as one would expect, the inequalities (35) generalise and dominate the inequalities (31). (To see this, just set $k = L(S)$ and note that $L(S)$ will never exceed $n_R - 1 - |S \cap V_R|$.) We conjecture that the constraints (30), (32) and (35) give a complete description of the projection. We also conjecture that the lower bound from formulation SCF2 always lies between the one from formulation SCF1 and the one from Fleischmann's formulation. (In practice, it is usually only slightly better than that of SCF1. See Table 3 in Section 5.)

3.4. Multi-commodity flow formulation

Now we move onto a multi-commodity flow (MCF) formulation. Similar to the MCF formulation for the standard TSP, we assume that the salesman leaves the depot (node 1) with one unit of commodity for each required node. Accordingly, let the binary variable g_a^k be 1 if and only if commodity k passes through arc a, for every $k \in V_R \backslash \{1\}$ and $a \in A$. The resulting formulation then consists of minimising (23) subject to the following constraints:

$$\sum_{a \in \delta^+(i)} \tilde{x}_a \geqslant 1 \quad (\forall i \in V_R) \qquad (36)$$

$$\sum_{a \in \delta^+(i)} \tilde{x}_a = \sum_{a \in \delta^-(i)} \tilde{x}_a \quad (\forall i \in V) \qquad (37)$$

$$\sum_{a \in \delta^-(i)} g_a^k - \sum_{a \in \delta^+(i)} g_a^k = 0 \quad (\forall i \in V \setminus \{1\}; k \in V_R \setminus \{1, i\}) \qquad (38)$$

$$\sum_{a \in \delta^-(k)} g_a^k - \sum_{a \in \delta^+(k)} g_a^k = 1 \quad (\forall k \in V_R \setminus \{1\}) \qquad (39)$$

$$\sum_{a \in \delta^-(1)} g_a^k - \sum_{a \in \delta^+(1)} g_a^k = -1 \quad (\forall k \in V_R \setminus \{1\}) \qquad (40)$$

$$\tilde{x}_a \geqslant g_a^k \quad (\forall a \in A;\ k \in V_R \setminus \{1\}) \qquad (41)$$

$$\tilde{x}_a \in \{0,1\} \quad (\forall a \in A) \qquad (42)$$

$$g_a^k \in \{0,1\} \quad (\forall a \in A \ \&\ k \in V_R \setminus \{1\}). \qquad (43)$$

The constraints are interpreted along similar lines to those of the formulations already seen.

This MCF formulation has $\mathcal{O}(n_R|E|)$ variables and $\mathcal{O}(n_R|E|)$ constraints. As for the projection into the space of x variables, we have the following result:

Theorem 4. *Let P^1 be the polytope in $(\tilde{x}, g)$-space defined by the constraints (37)–(41) and the trivial bounds $\tilde{x} \in [0,1]^{|A|}$, let P^2 be the projection of P^1 into $\tilde{x}$-space, and let P^3 be the projection of P^2 into x-space. Then P^3 is completely described by the inequalities (21), (30) and (32).*

Proof. Let k be an arbitrary node in $V_R \backslash \{1\}$. The classical *max-flow min-cut* theorem [14] implies that imposing the constraints (38)–(41) for the given k is equivalent to imposing the following inequalities:

$$\sum_{a \in \delta^+(S)} \tilde{x}_a^* \geqslant 1 \quad (\forall S \subset V_R \setminus \{1\} : k \in S).$$

Therefore, P^2 is completely described by (37), the trivial bounds $\tilde{x} \in [0,1]^{|A|}$ and the constraints

$$\sum_{a \in \delta^+(S)} x_a \geqslant 1 \quad (\forall S \subset V_R \setminus \{1\} : S \cap V_R \neq \emptyset).$$

Using (37), these last inequalities can be re-written in the form:

$$\sum_{a \in \delta^+(S) \cup \delta^-(S)} x_a \geqslant 2.$$

The result then follows from the mapping of $\tilde{x}$ onto x. □

An immediate consequence of Theorem 4 is the following:

Corollary 2. *The lower bound from the LP relaxation of MCF is equal to the lower bound from Fleischmann's formulation (19)–(22).*

4. Time-staged formulations of the STSP

In this section, we adapt the TS formulation for the standard TSP, mentioned in Section 2.2, to the Steiner case. A simple formulation is presented in the following subsection. A method to reduce the number of variables is presented in Section 4.2. Then, in Section 4.3, we evaluate the total number of variables and constraints in each of the formulations that we have considered.

4.1. An initial time-staged formulation

In this context, it is natural to have one time stage for each time that an edge of G is traversed (in either direction). In terms of the classical STSP formulation given in Section 2.3, the total number of time stages will then be equal to $\sum_{e \in E} x_e$. The problem here is that we do not know this value in advance. Observe, however, that Lemma 1 implies that it cannot exceed $2|E|$.

Now, let A be defined as in Section 3.1, and recall that $|A| = 2|E|$. For all $a \in A$ and all $1 \leqslant k \leqslant |A|$, let the binary variable r_a^k take the value 1 if and only if arc a is the kth arc to be traversed in the tour. Our TS formulation for the STSP is as follows:

$$\min \quad \sum_{k=1}^{|A|} \sum_{a \in A} c_a r_a^k \tag{44}$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(1)} r_a^1 = 1 \tag{45}$$

$$r_a^1 = 0 \quad (a \in A \setminus \delta^+(1)) \tag{46}$$

$$\sum_{k=1}^{|A|} \sum_{a \in \delta^+(1)} r_a^k = \sum_{k=1}^{|A|} \sum_{a \in \delta^-(1)} r_a^k \tag{47}$$

$$\sum_{k=1}^{|A|} \sum_{a \in \delta^+(i)} r_a^k \geqslant 1 \quad (\forall i \in V_R) \tag{48}$$

$$\sum_{a \in \delta^-(i)} r_a^k = \sum_{a \in \delta^+(i)} r_a^{k+1} \quad (\forall i \in V \setminus \{1\}; 1 \leqslant k \leqslant |A| - 1) \tag{49}$$

$$r_a^k \in \{0,1\} \quad (\forall a \in A, 1 \leqslant k \leqslant |A|). \tag{50}$$

Constraints (45) and (46) ensure that the salesman departs from the depot in the first time stage, and constraint (47) ensures that he arrives at the depot as many times as he leaves it. Constraints (48) ensure that each required node is visited at least once. Constraints (49) ensure that, if the salesman arrives at a non-depot node in any given time stage, then he must depart from it in the subsequent time stage. Finally, constraints (50) are the usual binary conditions.

We call this TS formulation 'TS1'. Note that TS1 has $\mathcal{O}(|E|^2)$ variables and $\mathcal{O}(n|E|)$ constraints. We will see in Section 5 that the lower bound from this formulation is very weak in practice.

4.2. Bounding the number of edge traversals

Clearly, one could reduce the number of variables and constraints in the TS formulation if one had a better upper bound on the total number of times that the salesman traverses an edge of G. The following theorem provides such a bound:

Theorem 5. *There exists an optimal STSP solution in which the total number of edge traversals (in either direction) does not exceed $2(n-1)$.*

For the proof of this theorem, we will use the following lemma.

Lemma 2. *Let H be a connected graph with k nodes and ℓedges. If $\ell > 2(k-1)$, then there exists a cycle C in H such that the graph arising when the edges of C are deleted from H is still connected.*

Proof. Let T be a spanning tree in H, and let H' be the graph resulting if the edges of T are deleted from H. For the number ℓ' of edges of H' we have $\ell' = \ell - (k-1)$, which, by the hypothesis in the lemma, is greater than $k-1$. Clearly, the number of nodes of H' is equal to k.

Now, let T' be a spanning forest in H'. Note that H' may fail to be connected. Firstly, if one of the connected components of H' contains an edge e other than those in T', then let C be the cycle defined by taking e and the path in T' connecting the end-nodes of e. Clearly, deleting the edges of C from H leaves a connected graph because connectivity is assured by the tree T.

But, secondly, it is impossible that all connected components of H' contain no other edges except those in T': In that case, H' would be a forest, and hence have at most $k-1$ edges. But the number of edges of H' is greater than $k-1$, a contradiction. □

We can now complete the proof of the theorem.

Proof of Theorem 5.. Let x be an optimal solution to the STSP, which has, among all optimal solutions, the smallest number of edge traversals.

Construct a graph H by starting with the node set V, and precisely x_e copies of the edge e, for all $e \in E$. Then delete every isolated node from H. The number of nodes k of H is at most n, and the number of edges is $\ell := \sum_{e \in E} x_e$.

For the sake of contradiction, we assume that $\ell > 2(n-1)$. If that is the case, then Lemma 2, is applicable. Let C be a cycle with the property given in the lemma, and let F be its edge set. For every $e \in E$, denote by y_e the number of times the edge e occurs in C. The fact that after deleting the edges of C from H, a connected graph remains, implies that $x - y$ is a solution to the STSP, whose total cost is at most that of x. Thus, $x - y$ is an optimal solution in which the total number of edge traversals is smaller than in x, contradicting the choice of x.

Thus, we conclude that $\sum_e x_e = \ell \leqslant 2(n-1)$. □

An immediate consequence of this theorem is that one does not need to define the variables r_a^k in the TS formulation when $k > 2(n-1)$. The constraints in which $k > 2(n-1)$ can be dropped as well. As a result, the number of variables and constraints in the TS formulation can be reduced to $\mathcal{O}(n|E|)$ and $\mathcal{O}(n^2)$, respectively. We call this smaller TS formulation 'TS2'. Since TS2 is obtained from TS1 by eliminating variables, the lower bound from the LP relaxation of TS2 is no worse than the one from TS1. We will see in Section 5 that it is usually slightly better (though still poor).

4.3. Summary

Table 1 displays, for each of the STSP formulations that we have considered, bounds on the total number of variables and

Table 1
Alternative STSP formulations and their size.

Formulation	Classical	SCF	MCF	TS1	TS2
Variables	$\lvert E\rvert$	$\mathcal{O}(\lvert E\rvert)$	$\mathcal{O}(n_R\lvert E\rvert)$	$\mathcal{O}(\lvert E\rvert^2)$	$\mathcal{O}(n\lvert E\rvert)$
Constraints	$\mathcal{O}(2^{n_R})$	$\mathcal{O}(\lvert E\rvert)$	$\mathcal{O}(n_R\lvert E\rvert)$	$\mathcal{O}(n\lvert E\rvert)$	$\mathcal{O}(n^2)$

constraints. Here, 'classical' refers to the formulation of Fleischmann [13] mentioned in Section 2.3, 'SCF' refers to either of the single-commodity flow formulations SCF1 and SCF2, given in Sections 3.2 and 3.3, 'MCF' refers to the multi-commodity flow formulation given in Section 3.4, 'TS1' refers to the time-staged formulation given in Section 4.1, and 'TS2' refers to the reduced time-staged formulation given in Section 4.2.

Observe that, in the case of real road networks, the graph G is typically very sparse, and we have $|E| = \mathcal{O}(n)$. Moreover, in many cases, n_R is small relative to n. So none of the formulations is particularly large. Therefore, the main thing determining whether or not the formulations are useful in practice is likely to be the amount of computing time and memory taken to solve them by branch-and-bound. This in turn is influenced by the quality of the LP relaxations and the time taken to solve those. The next section explores these computational issues.

We close this section with a remark on the MTZ formulation of the TSP. The MTZ formulation is based on the idea of determining the order in which the nodes are visited. Since nodes can be visited multiple times in the Steiner case, a unique order cannot be determined. As a result, it does not appear possible to adapt the MTZ formulation to the STSP. This is not a problem, though, given the extreme weakness of the MTZ formulation.

5. Computational results

Some experiments were conducted, in order to compare empirically our compact formulations of the STSP. The starting point was the creation of ten sparse graphs, with $n \in \{25, 50, 75, \ldots, 250\}$. The following procedure was used, which is designed to lead to graphs that resemble real-life road networks:

1. Set $V = \{1,\ldots,n\}$ and $E = \emptyset$.
2. Place the n nodes at random in a circle of radius 100.
3. Define the set of *potential edges* $P = \{\{i,j\}: 1 \leqslant i < j \leqslant n\}$.
4. For each $\{i,j\} \in P$, let the cost c_{ij} be the Euclidean distance between the end-nodes, rounded to the nearest integer.
5. Sort the potential edges in non-decreasing order of cost.
6. Examine each edge in P in turn. Insert it into E if both of the following conditions are satisfied:
   (a) It does not cross one of the edges already inserted into E.
   (b) It does not form an angle of less than 60° with an edge that has already been inserted into E and with which it shares an end-node.
7. If there are no isolated nodes in the resulting graph, output the graph. Otherwise, repeat the procedure.

For each of the ten graphs created, three STSP instances were created, by varying the proportion of required nodes. In the first such instance, each node has a probability of 1/3 of being required. In the second instance, the probability is 2/3, and in the third instance, all nodes are required. In each case, one of the required nodes was selected at random to be the depot. Fig. 1 shows the instance that was obtained for $n = 200$ and a probability of 2/3. The required and non-required nodes are represented by small red and green[1] circles, respectively, and the depot is represented by the small black square. The optimal solution for this instance can be deduced from Fig. 2, in which nodes that are not visited and edges that are not traversed have been omitted.

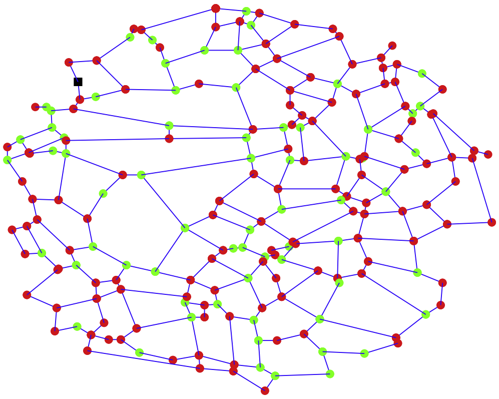

Fig. 1. STSP instance with $n = 200$ and $n_R = 133$.

A program was written in Microsoft Visual C that reads an instance from a file, and then outputs five integer programs, corresponding to the formulations SCF1, SCF2, MCF, TS1 and TS2. Each of the resulting 150 ($10 \times 3 \times 5$) integer programs was fed into the branch-and-bound solver of IBM CPLEX version 12.3. The computer used was a PC with a 1.6 GHz Intel Core i7 processor, with 8 GB of RAM, operating under Windows 7. Default CPLEX settings were used.

For each integer program, a time limit of 5000 s was imposed. We were able to find the optimal solutions to 28 out of the 30 instances within the time limit. The two that were not solved were the ones with $n \in \{225, 250\}$ and 2/3 of the nodes being required.

Table 2 shows, for each of the thirty instances and each of the five formulations, the time taken to solve the LP relaxation. We see that the LPs associated with the SCF formulations are very easy to solve, whereas the LPs associated with the other formulations consume more time. This is probably due to the fact that the SCF formulations have the fewest variables (see again Table 1). For a similar reason, the LP relaxation of TS2 can be solved more quickly than that of TS1.

Table 3 shows, for the 28 instances that were solved to optimality, and for the same five formulations, the *percentage integrality gap*, i.e., the difference between the lower bound given by the LP solution and the cost of the true integer optimum, expressed as a percentage of the latter. We see that the gaps are far smaller for the MCF formulation than for the SCF and TS formulations. The superiority of the MCF formulation over the formulation SCF1

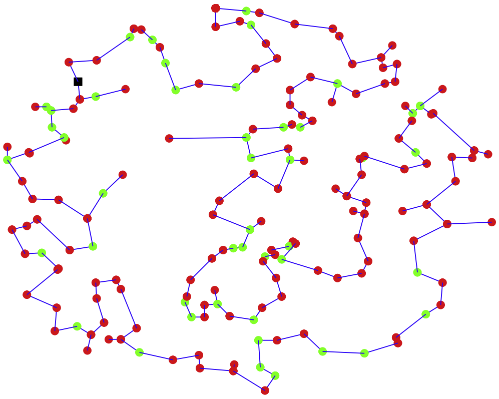

Fig. 2. Optimal tour for the STSP instance with $n = 200$ and $n_R = 133$.

[1] For interpretation of colour in Figs. 1 and 2, the reader is referred to the web version of this article.

Table 2
Time taken to solve LP relaxation, in seconds.

Instance	SCF1	SCF2	MCF	TS1	TS2
25-1/3	0.01	0.02	0.01	0.19	0.12
50-1/3	0.02	0.01	0.11	1.00	0.73
75-1/3	0.02	0.03	0.45	3.63	1.90
100-1/3	0.03	0.03	1.19	9.25	5.15
125-1/3	0.03	0.03	2.93	21.86	11.48
150-1/3	0.03	0.05	9.38	33.66	18.80
175-1/3	0.05	0.05	30.33	66.27	38.80
200-1/3	0.05	0.06	20.09	184.16	93.90
225-1/3	0.06	0.06	65.81	240.76	136.52
250-1/3	0.08	0.08	52.80	326.57	195.77
25-2/3	0.02	0.01	0.01	0.23	0.16
50-2/3	0.02	0.02	0.44	1.44	0.94
75-2/3	0.02	0.02	2.09	5.58	2.54
100-2/3	0.02	0.03	7.33	10.30	6.33
125-2/3	0.03	0.03	12.74	39.78	17.71
150-2/3	0.03	0.05	17.30	51.92	28.58
175-2/3	0.06	0.06	190.31	106.21	58.6
200-2/3	0.06	0.06	379.52	181.63	94.72
225-2/3	0.06	0.06	707.65	286.29	149.31
250-2/3	0.09	0.08	5521.67	516.30	327.87
25-3/3	0.02	0.02	0.06	0.34	0.30
50-3/3	0.01	0.02	1.34	1.48	1.15
75-3/3	0.02	0.03	3.96	5.43	3.51
100-3/3	0.03	0.03	8.80	14.10	8.19
125-3/3	0.05	0.03	18.49	35.51	22.53
150-3/3	0.05	0.05	33.20	76.24	40.89
175-3/3	0.06	0.05	634.25	151.26	78.83
200-3/3	0.06	0.06	2238.60	219.71	114.58
225-3/3	0.06	0.08	2839.66	454.76	237.78
250-3/3	0.08	0.09	5566.54	727.42	436.74

can be explained by comparing Corollaries 1 and 2. We also see that the gaps for TS2 are a little better than those for TS1. This is due to the fact that TS2 can be obtained from TS1 by fixing some variables to zero. Finally, observe that, as the proportion of required nodes increases, the gaps decrease for the SCF and TS formulations. The reason for this phenomenon is not clear.

Table 3
Percentage integrality gaps.

Instance	SCF1	SCF2	MCF	TS1	TS2
25-1/3	24.42	22.51	0.00	43.91	42.01
50-1/3	22.81	22.01	0.38	41.70	40.20
75-1/3	24.50	23.82	0.41	43.64	41.76
100-1/3	30.33	30.12	0.00	51.61	49.72
125-1/3	34.51	34.20	0.09	44.80	43.71
150-1/3	33.42	33.21	1.70	45.60	44.43
175-1/3	35.44	35.32	0.04	44.84	43.83
200-1/3	32.20	31.90	0.00	40.05	39.22
225-1/3	35.51	35.40	1.20	45.01	44.12
250-1/3	33.40	33.22	0.54	41.82	40.98
25-2/3	18.91	17.21	0.00	33.33	31.31
50-2/3	19.60	19.01	0.66	30.36	28.76
75-2/3	20.81	20.50	1.91	25.59	24.57
100-2/3	24.13	23.82	0.23	31.29	30.19
125-2/3	26.42	26.21	1.59	33.18	32.10
150-2/3	27.44	27.20	1.48	33.48	32.39
175-2/3	24.62	24.53	2.01	28.19	27.68
200-2/3	28.61	28.41	0.87	32.08	31.40
25-3/3	14.21	13.71	1.87	19.40	18.44
50-3/3	18.52	18.19	0.95	24.16	22.64
75-3/3	14.03	13.82	0.00	17.99	16.83
100-3/3	15.42	15.20	1.00	19.31	18.17
125-3/3	18.24	17.87	1.26	22.09	21.23
150-3/3	17.41	17.30	0.75	20.18	19.46
175-3/3	16.63	16.54	0.99	17.93	17.61
200-3/3	17.10	16.91	0.35	17.68	17.46
225-3/3	14.32	14.19	0.50	15.28	14.98
250-3/3	17.92	17.88	0.82	19.27	18.87

Table 4
Number of branch-and-bound nodes.

Instance	SCF1	SCF2	MCF	TS1	TS2
25-1/3	32	0	0	12,370	8733
50-1/3	650	529	0	–	–
75-1/3	2416	2530	0	–	–
100-1/3	4331	1414	0	–	–
125-1/3	4133	5588	0	–	–
150-1/3	20,931	139,198	73	–	–
175-1/3	64,354	21,491	0	–	–
200-1/3	12,040	37,628	0	–	–
225-1/3	786,162	442,307	210	–	–
250-1/3	96,728	–	34	–	–
25-2/3	0	0	0	6576	9908
50-2/3	655	606	0	–	–
75-2/3	955	3354	37	–	–
100-2/3	861	2383	0	–	–
125-2/3	19,505	63,716	753	–	–
150-2/3	143,990	72,724	–	–	–
175-2/3	76,311	153,114	–	–	–
200-2/3	237,057	2,296,863	–	–	–
25-3/3	53	199	0	78,302	38,508
50-3/3	1812	623	0	–	–
75-3/3	613	637	0	–	–
100-3/3	5590	6884	234	–	–
125-3/3	19,675	21,128	–	–	–
150-3/3	220,850	134,777	–	–	–
175-3/3	238,167	603,673	–	–	–
200-3/3	78,741	44,442	–	–	–
225-3/3	75,065	420,801	–	–	–
250-3/3	942,746	–	–	–	–

Table 4 shows, for each of the 28 instances solved, the number of branch-and-bound nodes needed to solve the integer programs to proven optimality. A dash (–) indicates that the given formulation could not be solved within the given time limit. We see that, for smaller values of *n*, the MCF formulation gives the true integer optimal solution without any branch-and-bound. For the SCF formulations, quite a few branch-and-bound nodes are needed in

Table 5
Branch-and-bound time, in seconds.

Instance	SCF1	SCF2	MCF	TS1	TS2
25-1/3	0.25	0.00	0.00	79.68	58.02
50-1/3	2.12	0.20	0.00	–	–
75-1/3	4.01	4.98	0.00	–	–
100-1/3	10.03	7.07	0.00	–	–
125-1/3	11.53	13.74	0.00	–	–
150-1/3	38.03	162.40	102.10	–	–
175-1/3	147.12	64.52	0.00	–	–
200-1/3	33.80	76.46	0.00	–	–
225-1/3	2221.34	1032.31	1420.6	–	–
250-1/3	438.72	–	290.8	–	–
25-2/3	0.00	0.00	0.00	104.96	71.84
50-2/3	1.48	1.86	0.00	–	–
75-2/3	3.46	4.81	12.15	–	–
100-2/3	6.74	6.65	0.00	–	–
125-2/3	26.76	79.94	3676.48	–	–
150-2/3	202.33	97.02	–	–	–
175-2/3	162.58	223.85	–	–	–
200-2/3	457.07	4235.12	–	–	–
25-3/3	0.22	0.08	0.00	623.69	271.42
50-3/3	1.47	0.23	0.00	–	–
75-3/3	0.42	2.96	0.00	–	–
100-3/3	9.19	9.14	608.90	–	–
125-3/3	45.41	39.67	–	–	–
150-3/3	211.23	149.15	–	–	–
175-3/3	393.43	1066.05	–	–	–
200-3/3	144.32	125.46	–	–	–
225-3/3	282.00	1064.5	–	–	–
250-3/3	4147.20	–	–	–	–

general. For the TS formulations, an excessive number of nodes is needed even for small values of n.

It should be noted that, in a few cases, the number of branch-and-bound nodes is zero even though the integrality gap is positive. The explanation for this anomalous behaviour is that, in some cases, CPLEX automatically appended some of its own internal cutting planes to the LP relaxation. If one were to switch off all internal cutting plane generation in CPLEX, then branching would become necessary whenever there is a positive integrality gap.

Finally, Table 5 shows the total time, in seconds, taken by the branch-and-bound procedure. We see that the SCF formulations perform remarkably well, enabling one to find a provably optimal solution even for quite large values of n. This is presumably due to the fact that, although the number of nodes is high, the time taken to process each node is small, due to the small number of variables.

For the MCF formulation, the time is small or even zero in many cases. Nevertheless, as the instances grow in size, MCF starts to struggle. This is probably due to the comparatively large number of variables. The TS formulations, on the other hand, perform consistently very poorly.

All things considered, we would recommend using the MCF formulation for n up to 100 or so, and SCF2 for n between 100 and 250.

6. Some related problems

Many variants and extensions of the TSP have appeared in the literature, such as the *Orienteering Problem* (e.g., [11,12,17]), the *Prize-Collecting* TSP (e.g., [3,4,11]), the *Capacitated Profitable Tour Problem* (e.g., [11,22]), the TSP with *Time Windows* (e.g., [2,9]) and the *Sequential Ordering Problem* [10]. For each of these problems, it is easy to define a 'Steiner' version. It suffices to define the problem on a general graph $G = (V,E)$, designate node 1 as the 'depot', define a set $V_R \subset V \backslash \{1\}$ of 'customer' nodes, permit edges to be traversed more than once if desired, and permit nodes to be visited more than once if desired.

In this section, we explore possible ways to formulate these other problems of 'Steiner' type. For the sake of brevity, however, we restrict attention to three specific problems, which we call the *Steiner Orienteering Problem*, the *Steiner Capacitated Profitable Tour Problem*, and the *Steiner TSP with Time Windows*. These are considered in the following three subsections.

6.1. The Steiner Orienteering Problem

We define the *Steiner Orienteering Problem* (SOP) as follows. For each $e \in E$, we are given a non-negative cost c_e. For each $i \in V_R$, we are given a positive *revenue* (or 'prize') p_i. The nodes in V_R do not all have to be visited, but the revenue can only be collected from such a node if that node is visited at least once. We are also given an upper bound U on the total route cost. The task is to maximise the sum of the prizes collected, subject to the upper bound.

Observe that Lemma 1 applies to the SOP. To see this, let $V^* \subset V_R$ be the set of nodes whose prizes are collected in the optimal solution. The optimal solution is then also optimal for a STSP instance defined on the same graph, but with V_R set to V^*.

Knowing that Lemma 1 applies, it is easy to adapt the classical (non-compact) formulation of the STSP, presented in Section 2.3, to the SOP. For each $i \in V_R$, we define a new binary variable y_i, taking the value 1 if and only if the salesman collects a prize from node i. We then change the objective function from (19) to:

$$\max \sum_{i \in V_R} p_i y_i, \tag{51}$$

replace the connectivity constraints (21) with:

$$\sum_{e \in \delta(S)} x_e \geqslant 2y_i \qquad (i \in V_R, S \subseteq V \setminus \{1\} : i \in S), \tag{52}$$

and add the route-cost constraint

$$\sum_{e \in E} c_e x_e \leqslant U. \tag{53}$$

It is also easy to adapt the TS formulation of the STSP (Section 4.1) to the SOP. It suffices to add the y_i variables mentioned above, change the objective function from (45)–(51), add the 'expanded' route-cost constraint

$$\sum_{k=1}^{|A|} \sum_{a \in A} c_a r_a^k \leqslant U,$$

and replace the constraints (48) with the constraints

$$\sum_{k=1}^{|A|} \sum_{a \in \delta^+(i)} r_a^k \geqslant y_i \quad (\forall i \in V_R). \tag{54}$$

Moreover, Theorem 5, given in Section 4.1, applies to the SOP as well (for the same reason that Lemma 1 applies). So one can reduce the number of stages to $2(|V| - 1)$, without losing any optimal solutions.

It is also easy to adapt the MCF formulation of the STSP (Section 3.4) to the SOP. It suffices to add the same y_i variables, change the objective function from (23)–(51), change the right-hand sides of constraints (36) and (39) from 1 to y_i, change the right-hand sides of constraints (40) from -1 to $-y_i$, and add the 'directed' route-cost constraint:

$$\sum_{a \in A} c_a \tilde{x}_a \leqslant U. \tag{55}$$

As for the SCF formulation of the STSP (Section 3.2), there is an elegant way to adapt it to the SOP, which leads to an LP relaxation with desirable properties. The key is to redefine the variables f_a, so that:

- if arc a is traversed (i.e., $\tilde{x}_a = 1$), then f_a represents the total cost accumulated so far when the salesman begins to traverse the arc;
- if arc a is not traversed (i.e., $\tilde{x}_a = 0$), then $f_a = 0$.

Once this is done, one can introduce the same additional y_i variables, and use the objective function (51), along with the following constraints:

$$\sum_{a \in \delta^+(1)} \tilde{x}_a \geqslant 1 \tag{56}$$

$$\sum_{a \in \delta^+(i)} \tilde{x}_a \geqslant y_i \quad (\forall i \in V_R) \tag{57}$$

$$\sum_{a \in \delta^+(i)} \tilde{x}_a = \sum_{a \in \delta^-(i)} \tilde{x}_a \quad (\forall i \in V) \tag{58}$$

$$\sum_{a \in \delta^-(1)} (f_a + c_a \tilde{x}_a) - \sum_{a \in \delta^+(1)} f_a \leqslant U \tag{59}$$

$$\sum_{a \in \delta^+(i)} f_a = \sum_{a \in \delta^-(i)} (f_a + c_a \tilde{x}_a) \quad (\forall i \in V \setminus \{1\}) \tag{60}$$

$$0 \leqslant f_a \leqslant (U - c_a)\tilde{x}_a \quad (\forall a \in A) \tag{61}$$

$$\tilde{x}_a \in \{0,1\} \quad (\forall a \in A) \tag{62}$$

$$y_i \in \{0,1\} \quad (\forall i \in V_R). \tag{63}$$

We then have the following analogue of Theorem 2:

Proposition 1. *Let P^1 be the polytope in $(\tilde{x}, y, f)$-space defined by the constraints (56)–(61) and the trivial bounds $\tilde{x} \in [0,1]^{|A|}$ and $y \in [0,1]^{|A|}$. Let P^2 be the projection of P^1 into $(\tilde{x}, y)$-space, and let P^3 be the projection of P^2 into (x, y)-space. Then P^3 is described by the inequality (53) and the following linear inequalities:*

$$\sum_{e\in\delta(1)} x_e \geqslant 2$$
$$\sum_{e\in\delta(i)} x_e \geqslant 2y_i \quad (i \in V_R)$$
$$\sum_{e\in\delta(S)} x_e \geqslant \frac{1}{U}\sum_{\{i,j\}\in E:\{i,j\}\cap S\neq\emptyset} c_e x_e \quad (\forall S \subset V\setminus\{1\})$$
$$(x,y)\in[0,2]^{|E|}\times[0,1]^{|E|}.$$

Proof. Similar to the proof of Theorem 2. One small addition is that one must sum together the constraint (59) and all of the constraints (60), and then replace pairs of $\tilde{x}$ variables with their corresponding x variables, to obtain the inequality (53). □

As in the case of the STSP (Section 3.3), it is possible to strengthen this SCF formulation of the SOP. Indeed, if a given arc (i,j) is traversed, then the smallest value that f_{ij} can take is equal to the cost of the shortest path from the depot to node i. Similarly, the largest value that f_{ij} can take is equal to $U - c_{ij}$ minus the cost of the shortest path from node j to the depot. One can adjust the constraints (60) accordingly, and then derive a stronger projection result, analogous to Theorem 3. We omit details, for the sake of brevity.

6.2. The Steiner Capacitated Profitable Tour Problem

The *Steiner Capacitated Profitable Tour Problem* (SCPTP) is similar to the SOP, but with the following differences:

- We are given a positive *demand* q_i for each $i \in V_R$, in addition to the revenue p_i.
- If we wish to gain the revenue for a given $i \in V_R$, then we have to deliver the demand q_i.
- Instead of an upper bound U on the route cost, we are given a vehicle capacity Q, which does not exceed the sum of the demands. The total demand of the serviced customers must not exceed Q.
- The task is to find a tour of maximum total profit, where the profit is defined as the sum of the revenues gained, minus the cost of the edges traversed.

Observe that Lemma 1 applies to the SCPTP, for the same reason that it applies to the SOP. Then, one can easily adapt the classical formulation of the STSP to the SCPTP. We use the same binary variables y_i as used in the previous subsection, change the objective function from (19) to

$$\max \sum_{i\in V_R} p_i y_i - \sum_{e\in E} c_e x_e,$$

replace the connectivity constraints (21) with the constraints (52), and add the capacity constraint

$$\sum_{i\in V_R} q_i y_i \leqslant Q. \tag{64}$$

One can adapt the TS formulation in a similar way. It suffices to add the same y_i variables, add the capacity constraint (64), change the objective function (45) to

$$\max \sum_{i\in V_R} p_i y_i - \sum_{k=1}^{|A|}\sum_{a\in A} c_a r_a^k,$$

and replace the constraints (48) with the constraints (54). Moreover, Theorem 5 is again applicable.

As for the SCF formulation, we propose again to redefine the variables f_a. Now, f_a represents the total load (if any) that is carried along the arc a. Then, again using the additional y_i variables, it suffices to:

$$\max \sum_{i\in V_R} p_i y_i - \sum_{a\in A} c_a \tilde{x}_a \tag{65}$$

subject to the following constraints:

$$\sum_{a\in\delta^+(1)} \tilde{x}_a \geqslant 1 \tag{66}$$
$$\sum_{a\in\delta^+(i)} \tilde{x}_a \geqslant y_i \quad (\forall i \in V_R) \tag{67}$$
$$\sum_{a\in\delta^+(i)} \tilde{x}_a = \sum_{a\in\delta^-(i)} \tilde{x}_a \quad (\forall i \in V) \tag{68}$$
$$\sum_{a\in\delta^+(1)} f_a - \sum_{a\in\delta^-(1)} f_a \leqslant Q \tag{69}$$
$$\sum_{a\in\delta^-(i)} f_a - \sum_{a\in\delta^+(i)} f_a = q_i y_i \quad (\forall i \in V_R) \tag{70}$$
$$\sum_{a\in\delta^-(i)} f_a - \sum_{a\in\delta^+(i)} f_a = 0 \quad (\forall i \in V\setminus(V_R\cup\{1\})) \tag{71}$$
$$0 \leqslant f_a \leqslant Q\tilde{x}_a \quad (\forall a \in A) \tag{72}$$
$$\tilde{x}_a \in \{0,1\} \quad (\forall a \in A) \tag{73}$$
$$y_i \in \{0,1\} \quad (\forall i \in V_R). \tag{74}$$

The analogue of Theorem 2 is now as follows.

Proposition 2. *Let P^1 be the polytope in $(\tilde{x},y,f)$-space defined by the constraints (66)–(72), and the trivial bounds $\tilde{x}\in[0,1]^{|A|}$ and $y\in[0,1]^{|A|}$. Let P^2 be the projection of P^1 into $(\tilde{x},y)$-space, and let P^3 be the projection of P^2 into (x,y)-space. Then P^3 is described by the inequality (64), together with the following inequalities:*

$$\sum_{e\in\delta(1)} x_e \geqslant 2$$
$$\sum_{e\in\delta(i)} x_e \geqslant 2y_i \quad (i \in V_R)$$
$$\sum_{e\in\delta(S)} x_e \geqslant \frac{2}{Q}\sum_{i\in S\cap V_R} q_i y_i \quad (\forall S \subseteq V\setminus\{1\} : S\cap V_R \neq \emptyset)$$
$$(x,y)\in[0,2]^{|E|}\times[0,1]^{|E|}.$$

Proof. Similar to that of Theorem 2. One small addition is that one must sum together the constraints (69)–(71), to obtain the inequality (64). □

As for the MCF formulation described in Section 3.4, one can adapt it to the SCPTP by redefining the binary variables g_a^k, so that they take the value 1 if and only if q_k units of commodity k pass through arc a. We omit the details for brevity.

6.3. The Steiner TSP with Time Windows

Finally, we define the *Steiner Traveling Salesman Problem with Time Windows* (STSPTW) as follows. For each $e \in E$, we are given, not only a non-negative cost c_e, but also a non-negative *traversal time* t_e. Moreover, for each $i \in V_R$, we are given a non-negative *servicing time* s_i, along with a *time window* $[a_i,b_i]$. Finally, we are given a positive time T by which the vehicle must return to the depot. All nodes in V_R must be visited at least once. On one such visit, the customer must receive service. The time at which service begins must lie between a_i and b_i. The task is to minimise the cost of the tour. We assume without loss of generality that the vehicle departs from the depot at time zero. We also assume that the vehicle is permitted to wait at any time.

Perhaps surprisingly, the situation here is completely different from those of the previous two subsections. To be specific:

- Lemma 1 does not apply. To see this, set $V = \{1,\ldots,4\}$, $V_R = \{2,3,4\}$ and $E = \{\{1,2\},\{1,3\},\{2,4\}\}$, set $c_e = t_e = 1$ for all $e \in E$, set $s_i = 1$ for $i \in \{2,3,4\}$, and set $a_2 = b_2 = 1$, $a_3 = b_3 = 3$, and $a_4 = b_4 = 6$. The unique optimal solution is for the salesman to service nodes 2–4 in that order, and then return to the depot. In this solution, the edge $\{1,2\}$ is traversed four times.
- Theorem 5 does not apply either. In the same example, the total number of edge traversals is 8, whereas $2(|V| - 1)$ is only 6.
- In fact it is not even true that the total number of edge traversals is bounded by $2|E|$, as the same example shows.
- The only thing that one can say in general seems to be that the total number of edge traversals is bounded by $(n_R + 1)(|V| - 1)$. (This is so since the maximum number of edge traversals between two successive occasions of service, or between a service and the vehicle leaving or returning to the depot, will never exceed $|V| - 1$ in an optimal solution.)

For these reasons, it does not seem possible to adapt the classical, SCF or MCF formulations to the STSPTW. It may be possible to adapt the TS formulation, but this would not be desirable, since (a) one would appear to need $(n_R + 1)(|V| - 1)$ time stages and (b) we already saw in the last section that time-staged formulations perform very poorly.

With some work, it is possible to formulate the STSPTW using only $\mathcal{O}(n_R|E|)$ variables and constraints. The key idea is to keep track of the number of required nodes that have been serviced so far each time one visits a node or traverses an edge. We omit details for the sake of brevity.

7. Concluding remarks

Since many real-life vehicle routing problems are defined on sparse networks, such as road networks, continued research on the Steiner TSP and its variants is of practical importance. We have seen that, if the Steiner TSP is formulated intelligently, then one can solve instances with over 200 nodes to proven optimality, without invoking sophisticated machinery such as branch-and-cut. (As mentioned in Section 5, however, CPLEX did sometimes add cutting planes of its own at the root node.) We have also shown how to adapt our best formulations to some related problems.

Possible topics for future research would be the derivation of smaller and/or stronger compact formulations for the problems mentioned, the derivation of useful compact formulations for the Steiner version of other routing problems, or testing whether the addition of cutting planes could enable even larger instances to be solved using our formulations.

Acknowledgements

The authors thank Juan-José Salazar-González for bringing Theorem 1 to our attention. Thanks are also due to an anonymous referee for helpful comments that led to significant improvements in the paper.

References

[1] D. Applegate, R. Bixby, V. Chvátal, W. Cook, The Traveling Salesman Problem: A Computational Study, Princeton University Press, Princeton NJ, 2006.

[2] N. Ascheuer, M. Fischetti, M. Grötschel, Solving the asymmetric travelling salesman problem with time windows by branch-and-cut, Math. Program. 90 (2001) 475–506.

[3] E. Balas, The prize collecting traveling salesman problem, Networks 19 (1989) 621–636.

[4] E. Balas, The prize collecting traveling salesman problem and its applications, in: G. Gutin, A. Punnen (Eds.), The Traveling Salesman Problem and its Variations, Kluwer Academic Publishers, Boston, 2002, pp. 663–695.

[5] A. Claus, A new formulation for the travelling salesman problem, SIAM J. Algebra. Discr. Methods 5 (1984) 21–25.

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

[7] G.B. Dantzig, D.R. Fulkerson, S.M. Johnson, Solution of a large-scale traveling-salesman problem, Oper. Res. 2 (1954) 363–410.

[8] E.W. Dijkstra, A note on two problems in connection with graphs, Numer. Math. 1 (1959) 269–271.

[9] Y. Dumas, J. Desrosiers, E. Gélinas, M.M. Solomon, An optimal algorithm for the travelling salesman problem with time windows, Oper. Res. 43 (1995) 367–371.

[10] L.F. Escudero, An inexact algorithm for the sequential ordering problem, Eur. J. Oper. Res. 37 (1988) 232–253.

[11] D. Feillet, P. Dejax, M. Gendreau, Traveling salesman problems with profits, Transport. Sci. 39 (2005) 188–205.

[12] M. Fischetti, J.-J. Salazar-González, P. Toth, The generalized traveling salesman problem and orienteering problems, in: G. Gutin, A. Punnen (Eds.), The Traveling Salesman Problem and its Variations, Kluwer Academic Publishers, Boston, 2002, pp. 609–662.

[13] B. Fleischmann, A cutting plane procedure for the traveling salesman problem on a road network, Eur. J. Oper. Res. 21 (1985) 307–317.

[14] L.R. Ford, D.R. Fulkerson, Maximal flow through a network, Can. J. Math. 8 (1956) 399–404.

[15] D. Gale, A theorem of flows in networks, Pacific J. Math. 7 (1957) 1073–1082.

[16] B. Gavish, S.C. Graves, The Travelling Salesman Problem and Related Problems, Working Paper, Operations Research Centre, Massachusetts Institute of Technology, 1978.

[17] B.L. Golden, L. Levy, R. Vohra, The orienteering problem, Nav. Res. Log. 34 (1987) 307–318.

[18] L. Gouveia, S. Voss, A classification of formulations of the (time-dependent) traveling salesman problem, Eur. J. Oper. Res. 83 (1995) 69–82.

[19] W.W. Hardgrave, G.L. Nemhauser, On the relation between the traveling salesman and longest path problems, Oper. Res. 10 (1962) 647–657.

[20] A. Hoffman, Some recent applications of the theory of linear inequalities to external combinatorial analysis, Proc. Symp. Appl. Math., vol. 10, AMS Publications, Providence, RI, 1960, pp. 113–128.

[21] D.J. Houck, J.C. Picard, R.R. Vemuganti, The traveling salesman problem as a shortest path problem: theory and computational experience, Opsearch 17 (1980) 93–109.

[22] M.K. Jepsen, B. Petersen, S. Spoorendonk, D. Pisinger, A Branch-and-Cut Algorithm for the Capacitated Profitable Tour Problem. Working Paper, Technical University of Denmark, 2012.

[23] A.N. Letchford, A. Oukil, Exploiting sparsity in pricing routines for the capacitated arc routing problem, Comput. Oper. Res. 36 (2009) 2320–2327.

[24] P. Miliotis, G. Laporte, Y. Nobert, Computational comparison of two methods for finding the shortest complete cycle or circuit in a graph, RAIRO 15 (1981) 233–239.

[25] C.E. Miller, A.W. Tucker, R.A. Zemlin, Integer programming formulations and travelling salesman problems, J. Assoc. Comput. Mach. 7 (1960) 326–329.

[26] D. Naddef, Polyhedral Theory and Branch-and-Cut Algorithms for the Symmetric TSP, in: G. Gutin, A. Punnen (Eds.), The Traveling Salesman Problem and its Variations, Kluwer Academic Publishers, Boston, 2002, pp. 29–116.

[27] T. Öncan, I.K. Altinel, G. Laporte, A comparative analysis of several asymmetric traveling salesman problem formulations, Compt. Oper. Res. 36 (2009) 637–654.

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

[29] M. Padberg, G. Rinaldi, A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems, SIAM Rev. 33 (1991) 60–100.

[30] M. Padberg, T. Sung, An analytical comparison of different formulations of the traveling salesman problem, Math. Program. 52 (1991) 315–357.

[31] S. Vajda, Mathematical Programming, Addison-Wesley, 1961.