A Set Covering Approach for the Double Traveling Salesman Problem with Multiple Stacks

Michele Barbato*, Roland Grappe, Mathieu Lacroix, and Roberto Wolfler Calvo

Université Paris 13, Sorbonne Paris Cité, LIPN, CNRS (UMR 7030),
F-93430, Villetaneuse, France
{barbato,lacroix,grappe,wolfler}@lipn.univ-paris13.fr

Abstract. In the double TSP with multiple stacks, a vehicle with several stacks performs a Hamiltonian circuit to pick up some items and stores them in its stacks. It then delivers every item by performing another Hamiltonian circuit while satisfying the last-in-first-out policy of its stacks. The consistency requirement ensuring that the pickup and delivery circuits can be performed by the vehicle is the major difficulty of the problem. This requirement corresponds, from a polyhedral standpoint, to a set covering polytope. When the vehicle has two stacks this polytope is obtained from the description of a vertex cover polytope. We use these results to develop a branch-and-cut algorithm with inequalities derived from the inequalities of the vertex cover polytope.

Keywords: double traveling salesman problem with multiple stacks, polytope, set cover, vertex cover, odd hole.

The *traveling salesman problem* (*TSP*) is the problem of finding a Hamiltonian circuit of minimum cost in a complete weighted digraph. The TSP is a well-known NP-hard problem. Nevertheless, one of the greatest advances in combinatorial optimization has been the design of algorithms that made possible to practically solve TSP instances of considerable size [1].

In this paper, we study a generalization of the TSP, namely the *double TSP with multiple stacks* (*DTSPMS*). In this problem, n items have to be picked up in one city, stored in a vehicle having s identical stacks of finite capacity, and delivered to n customers in another city. We assume that the pickup and the delivery cities are far from each other, thus the pickup phase has to be completed before the delivery phase starts. The pickup (resp. delivery) phase consists in a Hamiltonian circuit performed by the vehicle which starts from a depot and visits the n pickup (resp. delivery) locations exactly once before coming back to the depot. Each time a new item is picked up, it is stored on the top of an available stack of the vehicle and no rearrangement of the stacks is allowed. During the delivery circuit the stacks are unloaded by following a last-in-first-out policy:

* corresponding author

only the items currently on the top of their stack can be delivered. The goal is to find a pickup and a delivery circuit that are s-consistent and that minimize the total traveled distance — a pickup and a delivery Hamiltonian circuits are *s-consistent* if a vehicle with s stacks can perform both while satisfying the last-in-first-out policy and the capacity of the stacks.

The DTSPMS has been recently introduced in [22] and has received since considerable attention. Several heuristics [6], [7], [9], [22], combinatorial exact methods [15, 16] and branch-and-cut algorithms [17], [21] have been proposed for its resolution. When the vehicle has two stacks, the best algorithms [2], [5] solve to optimality instances with up to 16 items, but mostly fail from 18 items. The main conclusion that can be drawn is that the DTSPMS is extremely hard to solve in practice. We emphasize that, as noted in [2], the finiteness of the capacity is not the major computational difficulty.

An explanation of the fact that exact approaches fail to solve the DTSPMS efficiently is the following. The combinatorial structure behind the consistency of the two circuits has not been deeply addressed. In contrast, the routing part associated with TSP circuits is well understood.

In this paper, we enhance the approach of [2] to overcome this difficulty. More precisely, by focusing on the consistency requirements, we reveal a strong polyhedral connection between the formulation of [2] and set cover problems. This allows us to derive new valid inequalities for the DTSPMS which are embedded into a competitive branch-and-cut algorithm.

The approach of [2] mainly considers the variant of the problem where the stacks have an infinite capacity. The authors develop theoretical results which are implemented in a branch-and-cut framework. A second version of their algorithm is developed with additional features to handle stacks of finite capacity. As we focus on the consistency requirements, we restrict our attention to the problem with stacks of infinite capacity. Indeed, the features of [2] to handle the finite capacities can also be added to our framework. We refer to [2] for more details on these additional features. For a sake of clarity, DTSPMS will now refer to the variant where the stacks have an infinite capacity.

This paper is organized as follows. In Section 1 we recall the formulation for the DTSPMS introduced in [2] and the known results about the routing part associated with this formulation. In Section 2 we study the set covering polytope that arises from the consistency requirements. In Section 3 we consider the case of the DTSPMS with two stacks. We show that in this case the set covering polytope associated with the consistency requirements corresponds to a vertex cover polytope. By using this observation, we derive valid inequalities for the DTSPMS with two stacks. Finally, we test these inequalities in a branch-and-cut algorithm.

1 Formulation of the DTSPMS

In this section, we first describe the DTSPMS in terms of graphs and then present the integer linear formulation for the DTSPMS introduced in [2].

An instance of the DTSPMS with n items is given by a complete digraph, two cost vectors defined on its arcs and a positive integer. The complete digraph $D = (V, A)$ with $V = \{0, \ldots, n\}$ and $A = \{(i, j)\colon i \neq j \in V\}$ models both cities. The depot is vertex 0. Item i has to be picked up from vertex i of the first city, and delivered to vertex i of the second city. The vectors $c^1 \in \mathbb{R}^{|A|}$ and $c^2 \in \mathbb{R}^{|A|}$ represent the distances between the locations of the pickup and delivery cities, respectively. The positive integer s is the number of stacks of the vehicle. Hence, the DTSPMS consists in finding a pair of s-consistent Hamiltonian circuits C_1 and C_2 whose cost $c^1(C_1) + c^2(C_2)$ is minimum.

Each Hamiltonian circuit of D induces a linear order on $V \setminus \{0\}$ corresponding to the order in which the vertices of $V \setminus \{0\}$ are visited starting from 0. Since the pickup and delivery circuits are Hamiltonian circuits of D, the following proposition characterizes the s-consistency thanks to the linear orders they induce:

Proposition 1 ([4], [6], [25]). *A pickup circuit and a delivery circuit are s-consistent if and only if no $s+1$ vertices of $V \setminus \{0\}$ appear in the same order in the linear orders induced by the two circuits.*

Our starting point is the formulation of the DTSPMS of [2], which we now explain. First, the Hamiltonian circuits of D are represented with *arc variables* $x \in \mathbb{R}^{|A|}$ which model the arcs of the Hamiltonian circuits, and *precedence variables* $y \in \mathbb{R}^{n(n-1)}$ which model the associated linear orders. They are described by the following constraints [24]:

$$\sum_{j \in V \setminus \{i\}} x_{ij} = 1 \quad \text{for all } i \in V, \tag{1}$$

$$\sum_{i \in V \setminus \{j\}} x_{ij} = 1 \quad \text{for all } j \in V, \tag{2}$$

$$y_{ij} + y_{ji} = 1 \quad \text{for all distinct } i, j \in V \setminus \{0\}, \tag{3}$$

$$y_{ij} + y_{jk} + y_{ki} \geq 1 \quad \text{for all distinct } i, j, k \in V \setminus \{0\}, \tag{4}$$

$$x_{ij} \leq y_{ij} \quad \text{for all distinct } i, j \in V \setminus \{0\}, \tag{5}$$

$$y_{ij} \in \{0, 1\} \quad \text{for all distinct } i, j \in V \setminus \{0\}, \tag{6}$$

$$x_{ij} \in \{0, 1\} \quad \text{for all distinct } i, j \in V. \tag{7}$$

By the integrality constraints (6) and (7), constraints (1) and (2) ensure that each vertex has exactly one leaving and one entering arc. Inequalities (3) and (4) are the antisymmetry and transitivity constraints respectively, and each binary vector y satisfying them represents a linear order on $V \setminus \{0\}$ [13]. Finally, constraints (5) imply that if the arc (i, j) is in the Hamiltonian circuit then i precedes j in the associated linear order.

Therefore, the DTSPMS can be formulated as follows. Let $(x^1, y^1, x^2, y^2) \in \mathbb{R}^{|A|} \times \mathbb{R}^{n(n-1)} \times \mathbb{R}^{|A|} \times \mathbb{R}^{n(n-1)}$. The variables (x^1, y^1) will correspond to the arc and precedence variables associated with the pickup circuit whereas (x^2, y^2)

will refer to the arc and precedence variables associated with the delivery circuit. The solutions to the DTSPMS are described by the following constraints:[1]

$$(x^t, y^t) \text{ satisfies (1)–(7)} \quad \text{for } t = 1, 2, \tag{8}$$

$$\sum_{i=1}^{s} (y^1_{v_i v_{i+1}} + y^2_{v_i v_{i+1}}) \geq 1 \quad \text{for all distinct } v_1, \ldots, v_{s+1} \in V \setminus \{0\}. \tag{9}$$

Inequalities (8) ensure that (x^t, y^t) corresponds to a Hamiltonian circuit for $t = 1, 2$. Inequalities (9) imply that the two Hamiltonian circuits are s-consistent. Indeed, if a constraint (9) is not satisfied, then the vertices $v_{s+1}, \ldots, v_1$ associated with this constraint appear in this order in both the pickup and delivery circuits — a contradiction to Proposition 1. Proposition 1 being an equivalence, the correctness of the above formulation follows.

In the rest of the paper, we will denote by $DTSPMS_{n,s}$ the convex hull of the solutions to (8)–(9). Moreover, $ATSP_n$ will denote the convex hull of the solutions to (1)–(7).

The above formulation makes apparent that the DTSPMS may be separated into two parts: a routing part associated with (8) and a consistency part associated with (9). Every valid inequality for $ATSP_n$ can be used to strengthen the linear relaxation of the DTSPMS. Actually, every facet of $ATSP_n$ gives two facets $DTSPMS_{n,s}$, as expressed in the following theorem.

Theorem 2 ([2]). *For $n \geq 5$ and $s \geq 2$, if $ax + by \geq c$ defines a facet of $ATSP_n$, then $ax^t + by^t \geq c$ defines a facet of $DTSPMS_{n,s}$, for $t = 1, 2$.*

Theorem 2 characterizes a super-polynomial number of facets of $DTSPMS_{n,s}$ since $ATSP_n$ has a super-polynomial number of facets [10]. Unfortunately, none of these facets relies on the consistency part of the problem. This part has actually not been well studied, and the next section will address this matter.

2 A Set Cover Approach for the s-consistency

As stated in the previous section, there is a one-to-one correspondence between Hamiltonian circuits of D and linear orders on $V \setminus \{0\}$. Thus, the projection onto the precedence variables y^1, y^2 of the solutions to the DTSPMS corresponds to the couples of linear orders on $V \setminus \{0\}$ satisfying (9). When focusing on the consistency part of the problem, we will consider only the consistency constraints (9). In this case, we are interested in the following polytope:

$$SC_{n,s} = conv\{(y^1, y^2) \in \{0,1\}^{n(n-1)} \times \{0,1\}^{n(n-1)} : \text{(9) are satisfied}\}.$$

Clearly, we have $proj_{(y^1,y^2)}(DTSPMS_{n,s}) \subseteq SC_{n,s}$. Moreover, $SC_{n,s}$ is a *set covering polytope*, that is a polytope of the form $conv\{x \in \{0,1\}^d : Ax \geq \mathbf{1}\}$,

[1] In the rest of the paper, the DTSPMS will refer to either the problem and the integer linear formulation depending on the context.

with A being a 0,1-matrix. Set covering polytopes have been intensively studied — see for instance [3].

In constraints (9), the coefficients associated with y^1_{ij} and y^2_{ij} are the same for all $i \neq j \in V \setminus \{0\}$, and hence $SC_{n,s}$ has a specific form. Indeed, it turns out that all facets of $SC_{n,s}$ can be obtained by studying the following polytope, hereafter called *restricted set covering polytope*:

$$RSC_{n,s} = conv\{\ y \in \{0,1\}^{n(n-1)}:$$
$$\sum_{i=1}^{s} y_{v_i v_{i+1}} \geq 1 \text{ for all distinct } v_1, \ldots, v_{s+1} \in V \setminus \{0\}\}.$$

Moreover, as shown in the following lemma, the vertices of $RSC_{n,s}$ are connected to the ones of $SC_{n,s}$.

These results are not surprising, yet we did not find them in the literature, thus we provide our own proof. In our proofs we often implicitly use the fact that a binary point of a binary polytope is one of its vertices.

Lemma 3. *For $y = (y^1, y^2) \in \mathbb{R}^{n(n-1)} \times \mathbb{R}^{n(n-1)}$, define $f(y) \in \mathbb{R}^{n(n-1)}$ by $f(y)_{ij} = \max\{y^1_{ij}, y^2_{ij}\}$, for all distinct $i, j \in \{1, \ldots, n\}$. Then,*

$$RSC_{n,s} = conv\{f(y) \in \mathbb{R}^{n(n-1)} : y \text{ is a vertex of } SC_{n,s}\}.$$

Proof. Let $P = conv\{f(y) \in \mathbb{R}^{n(n-1)} : y$ is a vertex of $SC_{n,s}\}$.

To show $P \subseteq RSC_{n,s}$, let $\bar{v}$ be a vertex of P. By construction, $\bar{v} = f(\bar{y})$ for some vertex $\bar{y}$ of $SC_{n,s}$. Since $\bar{y}$ is binary, so is $\bar{v}$. In addition, $\bar{v}_{j_i j_{i+1}} = 0$ if and only if $\bar{y}^1_{j_i j_{i+1}} = \bar{y}^2_{j_i j_{i+1}} = 0$. The vector $\bar{v}$ being binary, $\sum_{i=1}^{s} \bar{v}_{j_i j_{i+1}} < 1$ if and only if $\bar{v}_{j_i j_{i+1}} = 0$ for all $i = 1, \ldots, s$. But this can happen only if $\sum_{i=1}^{s} (\bar{y}^1_{j_i j_{i+1}} + \bar{y}^2_{j_i j_{i+1}}) = 0$, which is impossible by $\bar{y} \in SC_{n,s}$ and (9). Hence, $\bar{v} \in RSC_{n,s}$. As this holds for every vertex $\bar{v}$ of P, convexity implies $P \subseteq RSC_{n,s}$.

We prove now that $RSC_{n,s} \subseteq P$. Given a vertex $\bar{v}$ of $RSC_{n,s}$, we define $\bar{y} = (\bar{y}^1, \bar{y}^2) \in \{0,1\}^{n(n-1)} \times \{0,1\}^{n(n-1)}$ as follows.

$$\bar{y}^1_{j_i j_{i+1}} = \bar{y}^2_{j_i j_{i+1}} = 1 \ \text{ if } v_{j_i j_{i+1}} = 1,$$
$$\bar{y}^1_{j_i j_{i+1}} = \bar{y}^2_{j_i j_{i+1}} = 0 \ \text{ otherwise.}$$

For distinct $j_1, \ldots, j_{s+1}$, we have $\sum_{i=1}^{s} (\bar{y}^1_{j_i j_{i+1}} + \bar{y}^2_{j_i j_{i+1}}) = 0$ if and only if $\bar{v}_{j_1 j_2} = \bar{v}_{j_2 j_3} = \cdots = \bar{v}_{j_s j_{s+1}} = 0$. The latter is impossible since $\bar{v} \in RSC_{n,s}$. Hence, since $\bar{y}$ is binary, it satisfies (9). Thus $\bar{y}$ is a vertex of $SC_{n,s}$. By construction, we have $\bar{v} = f(\bar{y})$, therefore $\bar{v}$ is a vertex of P. This holds for every vertex $\bar{v}$ of $RSC_{n,s}$, hence $RSC_{n,s} \subseteq P$ by convexity. □

The next proposition shows how the linear description of $SC_{n,s}$ can be deduced from the one of $RSC_{n,s}$. Inequalities that consist in 0,1 bounds on the variables are called *trivial*.

Proposition 4. *Every non-trivial facet-defining inequality of $SC_{n,s}$ is of the form $ay^1 + ay^2 \geq b$, where $ay \geq b$ is a non-trivial facet-defining inequality of $RSC_{n,s}$.*

Proof. Well-known results about set covering polytopes — see *e.g.,* [19] — immediately imply the following:

(i) $SC_{n,s}$ is full dimensional.
(ii) Inequalities $y^t_{ij} \leq 1$ define facets of $SC_{n,s}$ for all distinct $1 \leq i, j \leq n$ and $t = 1, 2$.
(iii) If $a^1y^1 + a^2y^2 \geq b$ is non-trivial and defines a facet of $SC_{n,s}$, then $b > 0$ and $a^t_{ij} \geq 0$ for all distinct $1 \leq i, j \leq n$ and $t = 1, 2$.

We first show that all facets of $RSC_{n,s}$ define facets of $SC_{n,s}$.

Claim. If $ay \geq b$ is a non-trivial facet-defining inequality of $RSC_{n,s}$, then $ay^1 + ay^2 \geq b$ is a facet-defining inequality of $SC_{n,s}$.

Proof. We first prove that $ay^1 + ay^2 \geq b$ is valid for $SC_{n,s}$. Let $\gamma = (\gamma^1, \gamma^2)$ be a vertex of $SC_{n,s}$ and suppose that $a\gamma^1 + a\gamma^2 < b$. By Lemma 3, $f(\gamma)$ is a vertex of $RSC_{n,s}$. From $\gamma \geq 0$, we get $f(\gamma)_{ij} \leq \gamma^1_{ij} + \gamma^2_{ij}$ for all distinct $1 \leq i, j \leq n$. Since, by (iii), $a_{ij} \geq 0$, we get $af(\gamma) \leq a\gamma^1 + a\gamma^2 < b$, contradicting the validity of $ay \geq b$ for $RSC_{n,s}$.

We now prove that $ay^1 + ay^2 \geq b$ defines a facet of $SC_{n,s}$. Let F' denote the facet of $RSC_{n,s}$ defined by $ay \geq b$ and $\{\xi^1, \ldots, \xi^{n(n-1)}\}$ be an affine base of F'. Since $b > 0$ these vectors are linearly independent. Thus the $2n(n-1)$ vectors $\{(\xi^\ell, \mathbf{0}), (\mathbf{0}, \xi^\ell)\}_{\ell=1,\ldots,n(n-1)}$ are linearly independent points of $SC_{n,s}$, satisfying $ay^1 + ay^2 \geq b$ with equality. ■

We now show that non-trivial facet-defining inequalities of $SC_{n,s}$ have a symmetric structure:

Claim. Let $a^1y^1 + a^2y^2 \geq b$ be a non-trivial facet-defining inequality of $SC_{n,s}$. Then $a^1 = a^2$.

Proof. Let us fix $i, j \in \{1, \ldots, n\}$ with $i \neq j$ and let us write for convenience the vectors $\gamma \in \mathbb{R}^{2n(n-1)}$ as $(\bar{\gamma}, \gamma^1_{ij}, \gamma^2_{ij})$. By contradiction, we suppose that $a^1_{ij} > a^2_{ij}$. By (iii), we get $a^1_{ij} > 0$. If $(\bar{\gamma}, 1, 1)$ is a vertex of $SC_{n,s}$, then so are $(\bar{\gamma}, 1, 0)$ and $(\bar{\gamma}, 0, 1)$, since, in each of constraints (9), the coefficients of y^1_{ij} and y^2_{ij} are the same.

Let $F = SC_{n,s} \cap \{a^1y^1 + a^2y^2 = b\}$ be the facet defined by the given inequality and B a base of F. It is not restrictive to assume B is composed of vertices of $SC_{n,s}$. Then, no element of B has the form $(\bar{\gamma}, 1, 1)$, as otherwise, by $\bar{a}\bar{\gamma} + a^1_{ij} + a^2_{ij} = b$ and $a^1_{ij} > 0$, we would get that $(\bar{\gamma}, 0, 1)$ violates the given inequality. Given that F arises from a non-trivial facet-defining inequality of $SC_{n,s}$, there exists $(\bar{\gamma}, 1, 0) \in B$ as otherwise, $F \subseteq SC_{n,s} \cap \{y^1_{ij} = 0\}$. This implies that $(\bar{\gamma}, 0, 1)$ violates the facet-defining inequality. We deduce that $a^1_{ij} \leq a^2_{ij}$. Symmetrically, $a^2_{ij} \leq a^1_{ij}$ and the desired equality follows. ■

We finally prove that all the facets of $RSC_{n,s}$ can be obtained from those of $SC_{n,s}$.

Claim. If $ay^1 + ay^2 \geq b$ is a non-trivial facet-defining inequality of $SC_{n,s}$, then $ay \geq b$ is a non-trivial facet-defining inequality of $RSC_{n,s}$.

Proof. The point $(\gamma, \mathbf{0})$ is a vertex of $SC_{n,s}$ whenever γ is a vertex of $RSC_{n,s}$. Thus the validity of $ay \geq b$ for $RSC_{n,s}$ follows from the validity of $ay^1 + ay^2 \geq b$ for $SC_{n,s}$.

Now, let us suppose, by contradiction, that $ay \geq b$ does not define a facet of $RSC_{n,s}$. Then there exists an integer $f \geq 2$ such that $a = \sum_{i=1}^{f} \lambda_i a^i$ and $b = \sum_{i=1}^{f} \lambda_i b^i$, where $\lambda_i > 0$ and $a^i y \geq b^i$ is a facet of $RSC_{n,s}$ for every $0 \leq i \leq f$. Thus, the inequalities $a^i y^1 + a^i y^2 \geq b^i$ are valid for $SC_{n,s}$. However, $(a, a) = \sum_{i=1}^{f} \lambda_i (a^i, a^i)$, contradicting the fact that $ay^1 + ay^2 \geq b$ defines a facet of $SC_{n,s}$. ■

□

Proposition 4 asserts that the linear description of $RSC_{n,s}$ immediately gives the description of $SC_{n,s}$. Since $proj_{(y^1,y^2)}(DTSPMS_{n,s}) \subseteq SC_{n,s}$, the s-consistency of two Hamiltonian circuits can be modeled by using inequalities which are valid for $RSC_{n,s}$. Our goal is to use such inequalities to better capture the s-consistency in a branch-and-cut algorithm to solve the DTSPMS.

3 Focus on Two Stacks

In this section we first observe that, in the special case of the DTSPMS with two stacks, the restricted set covering polytope is a vertex cover polytope. This result allows us to derive valid inequalities for the DTSPMS. These inequalities are then embedded in a branch-and-cut algorithm, described at the end of the section together with the corresponding experimental results.

3.1 A Vertex Cover Approach

As explained in the previous section, the linear relaxation of our formulation can be strengthened by studying facet-defining inequalities of $RSC_{n,s}$. When considering only two stacks, the polytope $RSC_{n,2}$ is:

$$conv\{y \in \{0,1\}^{n(n-1)} : y_{ij} + y_{jk} \geq 1 \text{ for all distinct } i,j,k \in V \setminus \{0\}\}.$$

As it turns out, $RSC_{n,2}$ can be expressed as a vertex cover polytope. Let $G_n = (U, E)$ be the graph whose vertices are u_{ij} for all distinct $i, j \in V \setminus \{0\}$ and the edges are $\{u_{ij}, u_{jk}\}$ for all distinct $i, j, k \in V \setminus \{0\}$. A *vertex cover* of a graph is a set S of vertices such that each edge contains a vertex of S. The *vertex cover polytope* of a graph is the convex hull of the incidence vectors of its vertex covers.

Please note that $RSC_{n,2}$ and the vertex cover polytope of G_n have the same variables. Moreover, each non-trivial inequality of $RSC_{n,2}$ contains two variables which correspond to the extremities of an edge of G_n. Therefore $RSC_{n,2}$ is nothing but the vertex cover polytope of G_n.

The vertex cover polytope has been intensively studied. Many families of valid inequalities are known. We will more specifically use the so-called odd hole inequalities to derive new valid inequalities for the DTSPMS with two stacks.

Odd Hole Inequalities. An *odd hole* of a graph $G = (W, F)$ is a vertex subset $H = \{v_1, \ldots, v_{2k+1}\}$ such that $\{v_i, v_j\} \in F$ if and only if $|i-j| = 1$ or $|i-j| = 2k$ for all distinct $i, j \in \{1, \ldots, n\}$. The following inequalities are valid for the vertex cover polytope of G [20]:

$$y(H) \geq \frac{|H|+1}{2} \text{ for all odd holes } H \text{ of } G. \tag{10}$$

Corollary 5. *Inequalities*

$$y^1(H) + y^2(H) \geq \frac{|H|+1}{2} \textit{ for all odd holes } H \textit{ of } G_n, \tag{11}$$

are valid for $DTSPMS_{n,2}$.

There is a one-to-one correspondence between the vertices of G_n and the arcs of D. However, if every odd circuit of D provides an odd hole of G_n, the converse is not true. Thus, inequalities (11) generalize the odd circuit inequalities introduced in [2].

3.2 A Branch-and-Cut Algorithm

This section presents a branch-and-cut algorithm for the DTSPMS with two stacks. The reader interested in an exhaustive description of branch-and-cut methods can refer to *e.g.,* [18].

Initialization. The linear program we start with for computing the lower bounds is the one given by inequalities (1)–(3) and (5) and the trivial inequalities. Since the available instances are symmetrical, we add the constraint $y^1_{12} = 1$ to our starting formulation. This trick halves the number of solutions to our problem without affecting the correctness of the algorithm. In addition, we provide our algorithm with the upper bound given by the heuristic algorithm of [7].

Separation. To strengthen the routing part we consider the so-called GDDL inequalities [12] and the 2-simple cut inequalities [11]. The separation phase is as follows. The families of inequalities are separated in this order:

- 2-consistency constraints (9),
- GDDL inequalities,
- 2-simple cut inequalities,
- transitivity constraints (4),[2]
- odd hole inequalities (11).

[2] We use the lifted version $y_{ij} + y_{jk} + y_{ki} - x_{ji} \geq 1$, for all distinct $i, j, k \in V \setminus \{0\}$.

Constraints (4) and (9) are separated by enumeration. For the 2-simple cut inequalities we use the exact separation algorithms given in [11]. We also use for separating the GDDL inequalities the algorithm of [11] which we restrict to the most promising cases to speed it up. Finally we apply the heuristic separation algorithm given in [23] for the odd hole inequalities to the point $\bar{y} = \bar{y}^1 + \bar{y}^2$, where $(\bar{y}^1, \bar{y}^2)$ are the precedence variables of the current solution. The separation of each family is performed when separating the previous ones yielded no violated constraint. Moreover we mention that, since inequalities (4) and (9) are problem-defining, we always separate them on integer current points.

3.3 Experimental Results

The branch-and-cut algorithm described above is a first and preliminary implementation of the vertex cover approach for the DTSPMS with two stacks. The algorithm has been coded in C++ using CPLEX 12.5 [8]. The graph-based routines have been coded with the COIN-OR library LEMON [14]. The algorithm is tested over the benchmark instances introduced in [22], with a CPU time limit of 3 hours. Tests are run in a Linux environment, using a 3.4 GHz Intel Core i7 processor, in sequential mode (1 thread).

Since we test our algorithm only for two stacks of unlimited capacity, we present a comparison with the approach given in [2]. However, please recall the conclusion of [2] stating that the capacity of the stacks has little impact on the performance of the algorithm, in terms of CPU time and enumerated nodes of the search tree.[3] Both versions of the algorithm of [2] with finite and infinite capacity for the stacks outperform all other exact approaches.

Table 1 presents the results obtained by the branch-and-cut algorithm described in this paper and those obtained in [2]. Each row of the table corresponds to a tested instance. The first two columns contain the information relative to each instance: its name given in [22] and the number of items it involves. For both algorithms the remainder of the table consists of five columns. Columns UB and LB respectively contain the value of the best integer feasible solution obtained for that instance, and the best lower bound obtained by the algorithm within the 3 hours. Columns CPU and Nodes respectively report the time spent (in seconds) and the number of nodes of the branch-and-cut tree. Finally, column Gap reports the gap for each instance, calculated as 100·(UB-LB)/UB.

The algorithm proposed in this paper solves all the instances up to 16 items to optimality. Moreover, it solves nine out of the 20 instances with 18 items. For the instances not solved to optimality, the average gap is 1.94% for 18 items.

Compared with [2], our current algorithm exhibits a better performance. More precisely, it needs respectively 5.9%, 33.7% and 7.3% less time to solve the instances with 14, 16 and 18 items. Moreover, it solves within the time limit one instance more with 18 items with respect to the algorithm of [2]. Finally, we mention that the algorithm presented in this paper solves at optimality two instances with 20 items, within the time limit.

[3] Note that the optimal values can differ when passing from the finite capacity case to the infinite capacity case.

Table 1.

		Our B&C					B&C of [2]				
Instance	Items	UB	LB	CPU	Nodes	Gap	UB	LB	CPU	Nodes	Gap
R00	14	766	766.00	147.99	1717	0.00	766	766.00	118,39	1544	0.00
R01	14	761	761.00	22.80	239	0.00	761	761.00	27.97	346	0.00
R02	14	690	690.00	68.65	833	0.00	690	690.00	129.33	1648	0.00
R03	14	791	791.00	28.77	336	0.00	791	791.00	52.13	593	0.00
R04	14	756	756.00	606.73	8305	0.00	756	756.00	509.33	6918	0.00
R05	14	773	773.00	87.35	958	0.00	773	773.00	127.46	1589	0.00
R06	14	811	811.00	16.10	167	0.00	811	811.00	28.71	304	0.00
R07	14	693	693.00	24.13	239	0.00	693	693.00	28.21	319	0.00
R08	14	824	824.00	288.28	3749	0.00	824	824.00	259.09	3573	0.00
R09	14	733	733.00	9.15	67	0.00	733	733.00	5.93	58	0.00
R10	14	733	733.00	95.29	1267	0.00	733	733.00	99.86	1330	0.00
R11	14	719	719.00	362.73	4359	0.00	719	719.00	238.89	2975	0.00
R12	14	803	803.00	86.78	1088	0.00	803	803.00	59.10	722	0.00
R13	14	743	743.00	28.04	319	0.00	743	743.00	36.56	508	0.00
R14	14	747	747.00	193.64	2207	0.00	747	747.00	353.82	4847	0.00
R15	14	765	765.00	29.90	308	0.00	765	765.00	32.47	484	0.00
R16	14	685	685.00	37.69	411	0.00	685	685.00	31.57	376	0.00
R17	14	818	818.00	142.82	1591	0.00	818	818.00	246.35	2992	0.00
R18	14	774	774.00	68.06	920	0.00	774	774.00	94.40	1325	0.00
R19	14	833	833.00	211.86	2472	0.00	833	833.00	237.57	3002	0.00
Average				*127.84*	*1577.60*	*0.00*			*135.86*	*1772.65*	*0.00*
R00	16	795	795.00	1346.11	10356	0.00	795	795.00	1498.13	12002	0.00
R01	16	794	794.00	104.99	686	0.00	794	794.00	169.58	1467	0.00
R02	16	752	752.00	5239.07	40516	0.00	752	752.00	6688.66	51700	0.00
R03	16	855	855.00	2431.51	18037	0.00	855	855.00	1879.71	13641	0.00
R04	16	792	792.00	3350.76	26204	0.00	792	792.00	6616.13	52883	0.00
R05	16	820	820.00	1203.36	9616	0.00	820	820.00	4248.95	32078	0.00
R06	16	900	900.00	813.29	5930	0.00	900	900.00	988.01	8057	0.00
R07	16	756	756.00	87.90	624	0.00	756	756.00	130.26	958	0.00
R08	16	907	907.00	1057.53	9036	0.00	907	907.00	1526.68	12634	0.00
R09	16	796	796.00	67.47	535	0.00	796	796.00	99.46	789	0.00
R10	16	755	755.00	357.42	2791	0.00	755	755.00	664.12	5300	0.00
R11	16	759	759.00	1095.26	8151	0.00	759	759.00	909.18	7377	0.00
R12	16	825	825.00	348.77	2661	0.00	825	825.00	653.00	5264	0.00
R13	16	824	824.00	427.94	3051	0.00	824	824.00	719.47	5878	0.00
R14	16	823	823.00	2764.04	20967	0.00	823	823.00	5892.60	41223	0.00
R15	16	807	807.00	934.73	6731	0.00	807	807.00	568.39	4549	0.00
R16	16	781	781.00	462.52	3850	0.00	781	781.00	2347.62	18234	0.00
R17	16	852	852.00	1584.47	12029	0.00	852	852.00	2136.11	16101	0.00
R18	16	846	846.00	1674.27	13835	0.00	846	846.00	1289.01	10532	0.00
R19	16	882	882.00	1566.98	11750	0.00	882	882.00	1589.97	12501	0.00
Average				*1345.92*	*10367.80*	*0.00*			*2030.75*	*15658.40*	*0.00*
R00	18	839	839.00	3485.49	17926	0.00	839	839.00	5128.95	28232	0.00
R01	18	825	825.00	1101.54	5129	0.00	825	825.00	1574.57	7119	0.00
R02	18	793	759.81	10800.00	47666	4.19	793	750.06	10800.00	46046	5.42
R03	18	896	864.13	10800.00	44448	3.56	896	848.67	10800.00	43700	5.28
R04	18	832	781.29	10800.00	41852	6.09	832	781.50	10800.00	44790	6.07
R05	18	873	858.42	10800.00	55248	1.67	873	847.60	10800.00	50545	2.91
R06	18	930	930.00	6454.46	33943	0.00	930	930.00	9257.50	44850	0.00
R07	18	805	805.00	1686.81	9072	0.00	805	805.00	1488.97	7918	0.00
R08	18	962	922.29	10800.00	47664	4.13	962	907.68	10800.00	43758	5.65
R09	18	815	815.00	254.36	1354	0.00	815	815.00	448.44	2510	0.00
R10	18	856	820.18	10800.00	47890	4.18	856	825.04	10800.00	44155	3.62
R11	18	813	795.99	10800.00	55568	2.09	823	788.97	10800.00	51234	4.13
R12	18	871	871.00	2650.59	12942	0.00	871	871.00	4291.89	21560	0.00
R13	18	845	845.00	3415.79	17689	0.00	845	845.00	3455.85	19047	0.00
R14	18	862	830.62	10800.00	47245	3.64	873	813.67	10800.00	40037	6.80
R15	18	869	840.90	10800.00	48243	3.23	869	834.64	10800.00	47370	3.95
R16	18	811	811.00	3195.68	16843	0.00	811	811.00	5499.46	28197	0.00
R17	18	900	862.50	10800.00	43859	4.17	900	840.50	10800.00	38099	6.61
R18	18	883	867.22	10800.00	50824	1.79	883	867.33	10800.00	47342	1.77
R19	18	909	909.00	7982.98	37904	0.00	909	893.13	10800.00	51974	1.75
Average				*7451.39*	*34165.45*	*1.94*			*8037.28*	*35424.15*	*2.70*

4 Concluding Remarks

In this paper we have considered the DTSPMS. We have focused on the s-consistency requirements ensuring that both the pickup and delivery circuits can be performed by a vehicle with s stacks satisfying the last-in-first-out policy conditions. We have considered the polytope defined by the consistency constraints and the trivial inequalities. It is a relaxation of the convex hull of the solutions to the DTSPMS but it catches most of the difficulty of the problem and every valid inequality for this polytope can be used to reinforce the DTSPMS. This polytope is a set covering polytope and we have shown that when we have only two stacks, this latter can be reduced to a vertex cover polytope.

We used these results to develop a branch-and-cut algorithm to solve the DTSPMS with two stacks of infinite capacity. This algorithm uses the inequalities derived from the odd hole inequalities which are valid for the vertex cover polytope. This branch-and-cut algorithm is competitive with respect to the existing algorithms for the DTSPMS. We believe that strengthening the formulation using inequalities derived from the vertex cover approach will provide an efficient algorithm to solve instances of a larger size.

Apart from these algorithmic questions, one can wonder whether the relaxation we have considered is far from the convex hull of the solutions to the DTSPMS. A way to answer this question is to determine which facets of the set covering polytope define facets of the convex hull. This is another direction of our future work.

References

1. Applegate, D.L., Bixby, R.E., Chvátal,V., Cook, W.J.: The Traveling Salesman Problem: A Computational Study. Princeton University Press, Princeton, NJ, USA (2007)
2. Barbato, M., Grappe, R., Lacroix, M., Wolfler Calvo, R.: Polyhedral Results and a Branch-and-Cut Algorithm for the Double Traveling Salesman Problem with Multiple Stacks. Accepted for publication in Discrete Optimization.
3. Borndörfer, R.: Aspects of Set Packing, Partitioning, and Covering. Doctoral dissertation, Ph. D. thesis, Technischen Universität Berlin, Berlin, Germany (1998)
4. Borne, S., Grappe, R., Lacroix, M.: ISCO 2012: The Uncapacitated Asymmetric Traveling Salesman Problem with Multiple Stacks. In: Mahjoub, A.R., Markakis, V., Milis, I., Paschos, V.T. (eds.) Combinatorial Optimization 2012. LNCS, vol. 7422, pp. 105–116. Springer, Heidelberg (2012)
5. Carrabs, F., Cerulli, R., Speranza, M.G.: A branch-and-bound algorithm for the double travelling salesman with two stacks. Networks 61, 58–75 (2013)
6. Casazza, M., Ceselli, A., Nunkesser, M.: Efficient algorithms for the double travelling problem with multiple stacks. Computers & Operations Research 39, 1044–1053 (2012)
7. Côté, J.-F., Gendreau, M., Potvin, J.-Y.: Large neighborhood search for the pickup and delivery traveling salesman problem with multiple stacks. Networks 60, 19–30 (2012)

8. IBM ILOG® CPLEX Optimization Studio. `http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud` (2012)
9. Felipe, A., Ortuno, M.T., Tirado, G.: The double traveling salesman problem with multiple stacks: A variable neighborhood search approach. Computers & Operations Research 36, 2983–2993 (2009)
10. Fiorini, S., Massar, S., Pokutta, S., Tiwary, H.R., De Wolf, R.: Exponential Lower Bounds for Polytopes in Combinatorial Optimization. Journal of the ACM 62, 17:1–17:23 (2015)
11. Gouveia, L., Pesneau, P.: On extended formulations for the precedence constrained asymmetric traveling salesman problem. Networks 48, 77–89 (2006)
12. Gouveia, L., Pires, J.M.: The asymmetric travelling salesman problem and a reformulation of the Miller-Tucker-Zemlin constraints. European Journal of Operational Research 112, 134–146 (1999)
13. Grötschel, M., Jünger, M., Reinelt, G.: Facets of the linear ordering polytope. Mathematical Programming 33, 43–60 (1985)
14. LEMON–Library for Efficient Modeling and Optimization in Networks. `http://lemon.cs.elte.hu/` (2010)
15. Lusby, R.M., Larsen, J., Ehrgott, M., Ryan, D.: An exact method for the double TSP with multiple stacks. International Transactions in Operational Research 17, 637–652 (2010)
16. Lusby, R.M., Larsen, J.: Improved exact method for the double TSP with multiple stacks. Networks 58, 290–300 (2011)
17. Martínez, A.A.M., Cordeau, J.-F., Dell'Amico, M., Iori, M.: A Branch-and-Cut Algorithm for the Double Traveling Salesman Problem with Multiple Stacks. INFORMS Journal on Computing 25, 41–55 (2011)
18. Mitchell, J.E.: Branch-and-cut algorithms for combinatorial optimization problems. In: Pardalos, P.M., Resende, M.G.C. (eds.) Handbook of Applied Optimization, Oxford University Press (2002)
19. Nobili, P., Sassano A.: Facets and lifting procedures for the set covering polytope. Mathematical Programming 45, 111–137 (1989)
20. Padberg, M.W.: On the facial structure of set packing polyhedra. Mathematical Programming 5, 199–215 (1973)
21. Petersen, H.L., Archetti, C., Speranza, M.G.: Exact solutions to the double travelling salesman problem with multiple stacks. Networks 56, 229–243 (2010)
22. Petersen, H.L., Madsen, O.B.G.: The double travelling salesman problem with multiple stacks – Formulation and heuristic solution approaches. European Journal of Operational Research 198, 139–147 (2009)
23. Rebennack, S., Oswald, M., Theis, D.O., Seitz, H., Reinelt, G., Pardalos, P.M.: A Branch and Cut solver for the maximum stable set problem. Journal of Combinatorial Optimization 21, 434–457 (2011)
24. Sarin, S.C., Sherali, H.D., Bhootra, A.: New tighter polynomial length formulations for the asymmetric traveling salesman problem with and without precedence constraints. Operations Research Letters 33, 62–70 (2005)
25. Toulouse, S., Wolfler Calvo, R.: On the Complexity of the Multiple Stack TSP, kTSP. In: Chen, J., Cooper, S.B. (eds.) Theory and Applications of Models of Computation. LNCS, vol. 5532 pp. 360–369. Springer, Heidelberg (2009)